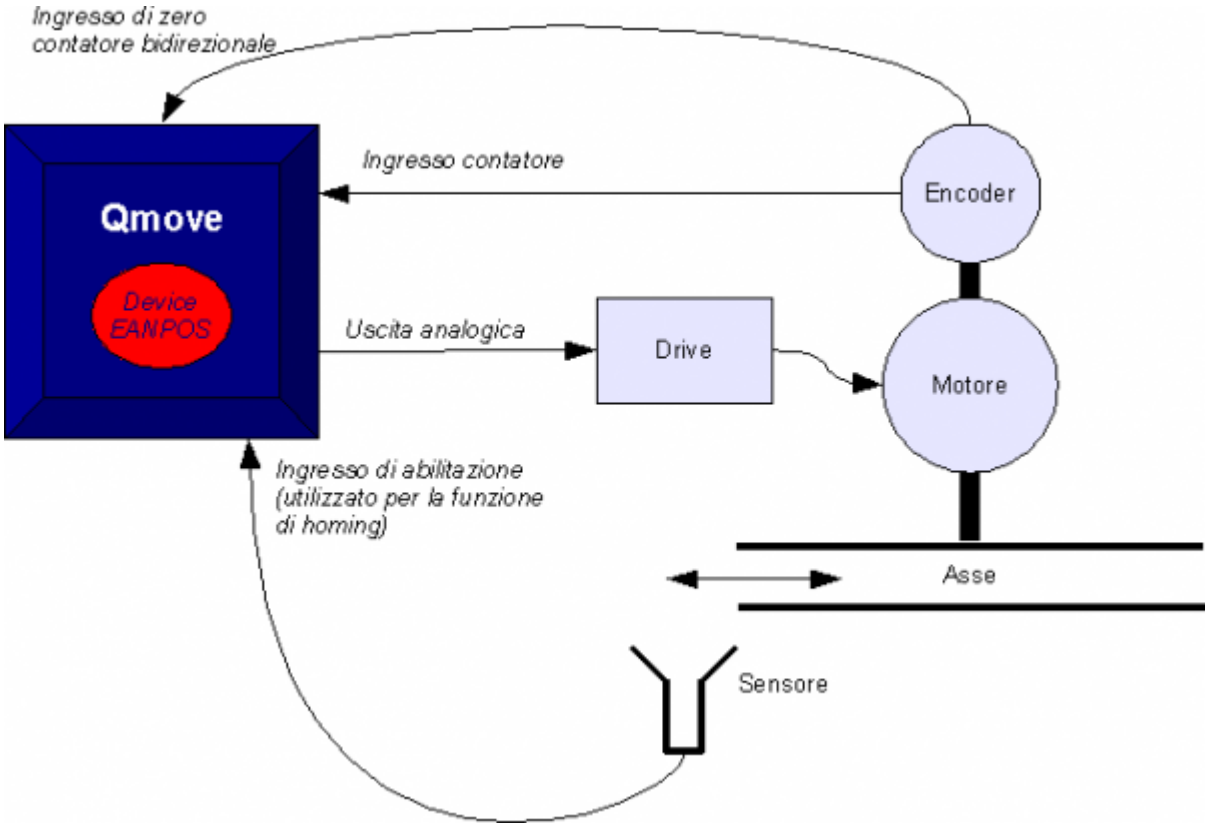
Sommario

AN004 - Example of using and calibrating the EANPOS device

In this section we want to describe the first steps that will make the user in his first contact with the EANPOS device. Provides a simple example that uses the EANPOS device to implement an analog positioner.

We can divide in the following sections, the proceed of the operation:

- device declaration in the configuration unit
- introduction of the parameters to correctly calibrate inputs and outputs
- development of the application according to the needs



Device declaration in the configuration unit

As was already explained in the description of the EANPOS device, you must program properly the unit application configuration. It is very important to the piece of code that declares the device, Here you should indicate the hardware resources to be used to ensure proper operation. It will be the responsibility of the programmer to pick and choose the most appropriate inputs and outputs. For example with the following line of code:

```
;--------------------------------
; Internal device declaration
;--------------------------------
INTDEVICE
AxisX  EANPOS  2  2.CNT02 3 3.INP01  3.AN01
```

You define an EANPOS device with “AxisX” name where the sampling time is 2 ms. Have been declared the following hardware resources: the input to the bidirectional meter has address 2.CNT02 (where 2 indicates the slot the card is installed while CNT02 is the mnemonic name of the input), the number of digital input for interruption devoted to zero-pulse bidirectional counter (number 3), a digital input for capturing zero pulse (3.INP01) and the address of the analog output 3.AN01 that regulating the output voltage to manage a drive.

An application that has just inside the device Declaration in configuration unit and a qcl unit that it does not execute anything (unless forced to WAIT) already allows to execute the first operations using the capabilities of the device. After you have transferred the application in the system and having put in RUN, It can change the parameters, observe the States or give commands to devices using the appropriate monitor from QView.
This is very convenient in the early stages of planning when you just want to make some runs or debugging of an application.

Correct device parameterization

After declaring hardware resources properly to use you need to set some parameters as components that are connected to the

Qmove product.

Introduction of measure and pulse

Let us consider the case where the bi-directional meter is a digital encoder. Suppose that this is directly keyed on an engine that is to move an axis. You will need to set the parameters correctly measure and pulse of the EANPOS device so that the latter can interpret the pulses of the meter and calculate the position of the axis.
The introduction of measure and pulse lets you match between a space in a unit of your choice and a certain number of pulses. In the event that the user already knows the space covered in a round encoder then you'll proceed directly to projecting values. Let's clarify this concept with an example: if the encoder emits 1000 pulses/Rev and you know that the axis moves about 5 cm when the encoder performs exactly one revolution then you can enter the following values:

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

The *measure* value introduced involves choosing a unit of measure of mm for measuring positions, in the *pulse* parameter introduces a value equal to the number of encoder impulses multiplied by 4. Remember that the relationship *measure/pulse* must be a value between 0.00935 to 1 (for precision limits of the device). It is important to emphasize that the values described above are taken as reference: It is not necessary to introduce the parameters with reference to an encoder revolution as we will describe below.

When the user does not know in advance the measurement parameters, will still be able to make the correct calibration by following these steps:

- give the *INIT* command to device, verify that the *st_init* status switch to 1
- through the “device monitor” of QView see on the pc the *posit* parameter value
- sets *measure* and *pulse* both to value 1
- move the axis manually by having him make a move a position easily measurable
- read the *posit* value
- now insert 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.

Another important step to take is to set the *maxpos* and *minpos* parameters that define respectively the maximum and the minimum position accessible from axis.

Choice of speed unit measurement

The unit of measurement of the instantaneous speed of the axis it was chosen by using *unitvel* and *decpt* parameters. You can select the unit of time of speed with the *unitvel* parameter: if this is to 0 then the speed is measured in Um/min, if it is to 1 then the speed is measured in Um/s. The *decpt* parameter instead determines whether speed-measuring in multiples of the fundamental units Um. For example, if fundamental measurement units Um=mm, and unitvel=1 you get the speed indicator in the vel variable in:
mm/s (con decpt = 0),
cm/s (con decpt = 1),
dm/s (con decpt = 2),
m/s (con decpt = 3).
Afterwards, if necessary, you need to set the proper display on the terminal operator to adjust the correct decimal point position.

Analogue output calibration

Attention:
before actual placement you must make sure that wiring and mechanical parts are not cause malfunctions.

We review the case where the EANPOS device you're using an analog output that is implemented with a DAC device: this will assume 16 bit resolution input discrete values (so between -32768 to 32767) to give analog voltage output range ±10V. This calibration function with analog output can be driven with a constant value in order to test links and functionality.

Preliminary motion

This section describes the operations to be carried out to verify the correctness of the connections and the functionality of the

system which was built.

- give the *INIT* command to the device, verify that the *st_init* status switch to 1
- give the *RESUME* command to remove a possible emergency condition (*st_emrg* = 1)
- enable axis calibration status by running the *CALON* command, the *st_cal* state switch to 1
- in these conditions you can set the analog voltage using *vout* parameter: the value is expressed in tenths of a Volt (then the range of values entered is ±100). It is recommended to introduce low values (5, 10, 15 ...)
- because now the device is used as “voltage generator” the axis should start moving. If this is not the case you should verify the correctness of the connections. When the axis is moving the *frq* parameter indicates the frequency of one of the input signals to the bi-directional meter, vel indicates the speed of the axis while *posit* the position according to the selected unit of measure. If giving positive voltage the position decreases you must reverse the phases of transducer (or by reversing the cables, or by using the *CNTREV* command) or reverse the direction of the drive
- if with output voltage 0 V you note however that the axle is moving due to offset voltages, These can be offset using the *offset* parameter. For an optimal outcome of the calibration, the operation must be performed with the system temperature
- now you can disable calibration status with the *CALOFF* command (the *st_cal* state switch to 0)

Analog output parameterization (setting maxvel parameter)

The EANPOS device generates the voltage value of the analogue output on the basis of a ratio between the maximum speed of the axis and the maximum output voltage. To do this you must set the *maxvel* parameter, that is the speed at which it moves the axle when you are given maximum voltage to the drive. Obviously the axle must behave symmetrical analog voltage to zero, Therefore the speed must be the same (in module) to the maximum positive or negative voltage.
In order to know the maximum speed there are two ways: the “theoretical method” assumes to know the maximum speed set in drive from which one can easily derive the linear speed.
If you are not aware of the official maximum speed of the motor you must do this:

- enter the in the calibration mode (as described above)
- if the system permits provide the maximum voltage to the drive and read the value of the *vel* parameter
- you can also provide a lower voltage and calculate the maximum speed with the *vout* : 10 V = *vel* : *maxvel* proportion

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

First moving

Attention:
before moving the axis, verify proper operation of emergency and protection devices.

The procedures described here have allowed us to complete the first phase of parameterizing device. Now you can run smooth movement of the axis. For example, follow these steps:

- give the *INIT* command to the device, verify that the *st_init* status switch to 1
- move the axis in a position whereby you can make a particular space unobstructed position switches
- set the parameter that defines the time of the axis to achieve maximum speed from 0 (and vice versa) *taccdec*=50
- set the positioning speed with the *setvel* parameter
- set the positioning quota with the *setpos* parameter
- reset any state of emergency with the *RESUME* command
- start positioning with the *START* command, to stop the movement give the *STOP* command (or *EMRG*).

This first movement was done without turning the ring by reaction of space, so any error introduced by the values of offset voltage or from external influences is not correct.

First setting the parameters of the PID controller

Until now it was explained how to do the first placements with the ring of open space, let us now see how to make a first

calibration of the PID parameters.
Follow this steps:

- give the *LOOPON* commnadd to the device, verify that the *st_loopon* state switch to 1
- right now the ring of space will be closed
- set the *feedfw* parameter to 1000 (or 100%)
- give an arbitrary value to the *pgain* parameter, for example 20
- increase the value of the *pgain* parameter until you notice that the axis connected to the motor being monitored vibrate
- at this point begin to decrease *pgain* until this phenomenon of vibration finish
- the value of the *integt* and *derivt* parameters can be left to 0. If you need to follow the procedures below to give them a value.

Using of RECDATA device for calibrate the PID controller

When the programmer intends to regulate precisely the parameters of PID controller, he can use a device inside (residing on the QMove) that allows you to record data of the type: primary pulse position, virtual axis positions, analog outputs, following errors, inputs and outputs states. This tool is called RECDATA and uses system CPU RAM of QMove. When used, the memory usage of the data must not exceed 50% of total disability.

Since the effects of PID control produce events very difficult to appreciate with the naked eye, the RECDATA device becomes very useful when you want to adjust *pgain*, *integt* or *derivt* parameters. In fact phenomena such as overshoot or slow due to an excessive value of the integral time must be taken into account during parameterization.

The us of the RECDATA device is very simple, It is sufficient that it is correctly declared in the unit of configuration with a type line:

```
;---------------------------------
; Internal device declaration
;---------------------------------
INTDEVICE
Recorder  RECDATA  TCamp  QCTL1  QCTL2  IOutA1  IOutA2  IntL1  IntL2  Ing1  Ing2  Out1  Out2
```

The device is particularly suitable to monitor the settings of a EANPOS device, can then view: position values in impulses (taking data directly from bi-directional counter), theoretical position values that the calculated speed profile generator, the values of an analog output. The user can then easily see whether the parameters you have entered are correct for your needs. The RECDATA device can record information about two axes, for our purpose we can only declare relative addresses to the first axis leaving to X.X the other.

A more detailed description of the RECDATA device it can be found in section, here we simply list the steps to use it:

- device declaration in the configuration unit entering as input and output parameters used by the EANPOS device
- Open the menu panel Monitor...and choice which parameters register
- give the *STARTR* command when you want to start recording, and *STOPR* when you want to stop
- after completing the operations from the “Monitor” menu of the QView select “Device Panels”, and from there the relative value to the RECDATA device
- in the window that opens, you have several options, the most important step to take is to give the “Start data acquiring” command through the appropriate icon
- on thee cartesian axes, at the end of loading, will show the parameter values that were monitored.

Note:
the RECDATA device registers a significant amount of data within the product QMove RAM, at the time of acquisition by the pc the transfer takes place via serial. This operation can be very long because of the low speed of serial comunication. The user then can adopt some measures as:
- The user then can adopt some precautions pay attention to record only the variables that need to be monitored
- increase the sampling time of the RECDATA device

Development of an application that implements a positioner

In the previous section, you learned what are the first steps to be followed for handling procedure of an axis with an analog positioner. This example uses only a small range of adjustable parameters of the device, in this section we include sample code, commented in detail, from which the user can get ideas to develop an application. The way the device must be declared is explained above, and so this section is omitted the configuration unit.See here

This application uses 4 inputs: 1 manual forward, 1 manual backward, 1 for start command and 1 for stop command. Manual and automatic operation mode will be available: in manual can be use the jog commands while in automatic it can to make a positioning to the position and the desired speed:

```
;*******************************************************************************
; Configuration unit (reported only the declaration of variables and
; and are omitted declarations of the bus and the device)
;*******************************************************************************
;-------------------------------------------------------------------------------
; SYSTEM Variables Definition
;-------------------------------------------------------------------------------
SYSTEM
slQuotaPos L                ;Variable for placement quota
slVelAsse L                 ;Variable for axis speed
;-------------------------------------------------------------------------------
; GLOBAL Variables Definition
;-------------------------------------------------------------------------------
GLOBAL
gfMovMan F                  ;Flag signaling ongoing manual movements
gfMovAuto F                 ;Flag signaling ongoing automatic movements
;-------------------------------------------------------------------------------
; INPUT Variables Definition
;-------------------------------------------------------------------------------
INPUT
ifAvMan F 2.INP01           ;Manual forward input
ifInMan F 2.INP02           ;Manual backward input
ifStart F 2.INP03           ;START axis input
ifStop  F 2.INP04           ;STOP axis input
;-------------------------------------------------------------------------------
; OUTPUT Variables Definition
;-------------------------------------------------------------------------------
OUTPUT
ofToll F 2.OUT01            ;Axis in tolerance output
ofAxeFermo F 2.OUT02        ;Stopped axis output

;*******************************************************************************
; Unit qcl
;*******************************************************************************
;-------------------------------------------------------------------------------
; Device parameter adjustment operations
;-------------------------------------------------------------------------------
Axis:measure = 10000
Axis:pulse = 40000          ;how to calculate the measure and pulse is explained in special section.*
Axis:maxvel = 100000        ;how claculate maxvel is explained in special section.*
Axis:maxpos = 999999        ;Maximum quota
Axis:minpos = -999999       ;Minimum quota
Axis:maxfollerr = 10000     ;Maximum following error
Axis:unitvel = 0            ;Time unit of speed (speed in Um/min)
Axis:decpt = 0              ;Decimal digits in speed calculation
Axis:rampmode = 0           ;Type of used ramps (in this case the same time of acceleration or deceleration
                            ;from zero speed to maximum speed)
Axis:taccdec = 100          ;Acceleration and deceleration time
Axis:pgain = 10             ;Proportional gain
Axis:feedfw = 1000          ;Feedforward

INIT Axis                   ;Device initialization
WAIT Axis:st_init           ;Wait until the device is initialized
CNTUNLOCK Axis              ;Unlock position capture
WAIT NOT Axis:st_cntlock    ;Wait for the acquisition of the position is unlocked
CNTDIR Axis                 ;Sets the direction of the acquisition of position
WAIT NOT Axis:st_cntrev     ;Wait until it has the sense of acquisition of position
REGON Axis                  ;Enable the calibration
WAIT NOT Axis:st_regoff     ;Wait for enabling calibration
RESUME Axis                 ;Remove the axle from the emergency state
WAIT NOT Axis:st_emrg       ;Wait until the axle is not in emergency
LOOPON Axis                 ;Snap reaction axis loop
WAIT Axis:st_loopon         ;Wait until it is hooked on the reaction axis loop
IF (slVelAsse EQ 0)         ;In case the set of axis speed is zero
  slVelAsse = 50            ;Sets a positioning speed
ENDIF
IF (slQuotaPos EQ 0)        ;In case the axis placement quota is zero
  slQuotaPos = 2000         ;Sets a quota of positioning
ENDIF

;-------------------------------------------------------------------------------
; Positioning tasks
;-------------------------------------------------------------------------------
; ------------------------------ used variables -----------------------------
; slVelAsse:  Adjustable variable that represents the axis speed (expressed in % of the max speed)
; slQuotaPos: Adjustable variable that represents the share of axis positioning
; ---------------------------------- used flags ---------------------------------
; gfMovMan:   Manual operation in progress
; gfMovAuto:  Automatic movement in progress
;-------------------------------------------------------------------------------

MAIN:
;-------------------------------------------------------------------------------
; Output management
;-------------------------------------------------------------------------------
ofToll = Asse:st_toll       ;Sets the tolerance output as tolerance state
ofAxeFermo = Asse:st_still  ;Sets the stationary axis output as the stationary axis state

;-------------------------------------------------------------------------------
; Managing automatic movements
;-------------------------------------------------------------------------------
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:setvel=(slVelAsse*Asse:maxvel)/100 ;Sets the speed of the axis
      Axis:setpos = slQuotaPos      ;Sets the placement quota
      START Axis            ;Run the start of axis
      gfMovAuto = 1         ;Reports 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 stationary
    STOP Axis               ;Execute the stop of the axis
  ENDIF
ENDIF
IF gfMovAuto                ;Controls automatic movement reporting in progress
  IF Axis:st_still          ;Check that the axis is stopped
    gfMovAuto = 0           ;Reset status of automatic movement
  ENDIF
ENDIF

;-------------------------------------------------------------------------------
; Manual movement management (JOG)
;-------------------------------------------------------------------------------
IF ifAvMan                  ;Awaiting the input of manual operation
  IF NOT (gfMovAuto OR gfMovMan)    ;Check that there are no automatic or manual movements in progress
    IF Axis:st_still        ;Check that the axis is stopped
      Axis:setvel=(slVelAsse*Axis:maxvel)/100 ;Sets the speed of the axis (percentage of the maximum speed)
      MANFW Axis            ;Forward axis in manual
      gfMovMan = 1          ;Manual movement in progress reports
    ENDIF
  ENDIF
ENDIF
IF ifInMan                  ;Awaiting the input of manual operation
  IF NOT (gfMovAuto OR gfMovMan)    ;Check that there are no automatic or manual movements in progress
```

```
    IF Axis:st_still                ;Check that the axis is stopped
      Axis:setvel=(slVelAsse*Axis:maxvel)/100 ;Sets the speed of the axis
      MANBW Axis                    ;Backward axis in manual
      gfMovMan = 1                  ;Manual movement in progress reports
    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                    ;Take your reporting moving axis manual
  ENDIF
ENDIF

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

*How to measure and calculate pulse is explained in special section.