

## Table of Contents

|   |   |
|---|---|
| <b><i>AN006 - Example of using and calibrating the HEAD2 device</i></b> ..... | 3 |
| Device declaration in the configuration unit .....                            | 3 |
| Correct device parameterization .....   | 3 |
| Calibration for the measurement of the speed tape .....                       | 4 |
| Calibrating presence piece sensor .....                                       | 4 |
| Setup working heads .....   | 5 |



## AN006 - Example of using and calibrating the HEAD2 device

In this section we want to describe the first steps that will make the first approach to the HEAD2 device. 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 device, you must program properly the configuration unit of application. 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
Heads HEAD2 0002 2.CNT01 3.INP01 X.X X.X 3.OUT01 3.OUT02 3.OUT03 3.OUT04 X.X X.X X.X X.X

```

You define an HEAD2 device with "Heads" name with sampling time to 2 ms. Have been declared the following hardware resources: the input to the bidirectional counter has 2.CNT01 address (where 2 indicates the slot seat clamp identification while CNT01 is the mnemonic name of the input), the digital input on the piece presence sensor 3.INP01 (the remaining two entries have been omitted which may be used by sensors for power factor correction of the position of the workpiece) and four digital outputs that will command the machine sanding heads.

An application that has just inside the device declaration in the configuration unit and a qcl unit that it does not run anything (except WAIT forced) already allows to perform the first operations using the device capabilities. In fact, after you download the application on the instrument and having done work, 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 being debugged.

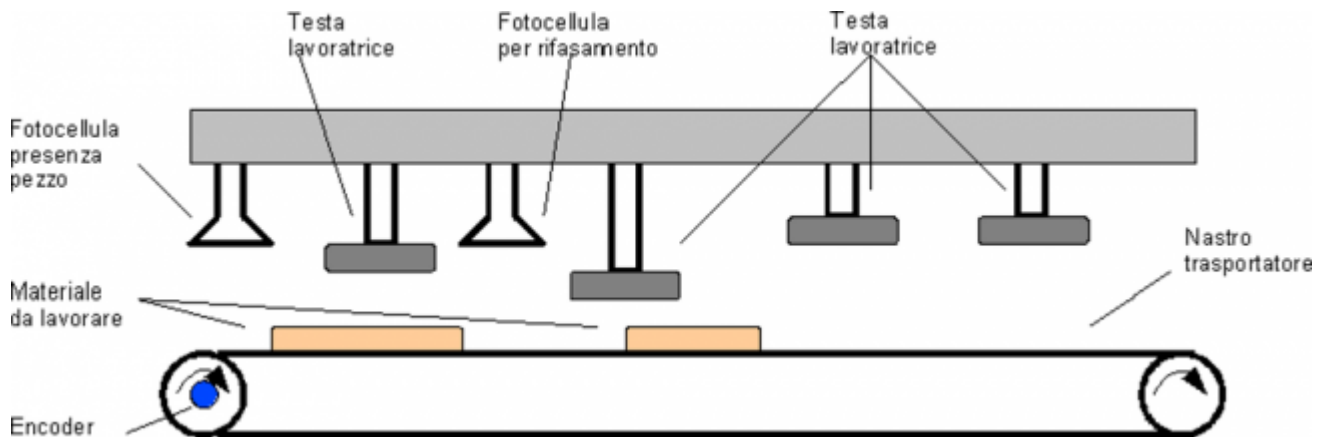
Ensure that the device operates properly however, some parameters that are configured correctly carrying out a proper calibration to adapt the capabilities of the instrument to the physical issues of the machine.

### Correct device parameterization

At this point it is worth recalling how could a machine controlled by the HEAD device: a conveyor belt allows you to move material below some machining heads, This is connected to an encoder that allows the tool to know his speed. It should be noted that this speed is only measured and that there is no control of axis movement.

At the beginning of the tape is usually placed a sensor (optical or mechanical) that determines the presence and length of a piece of material to be processed, it is possible also to use more two sensors at different distances in order to verify the presence of the piece.

The heads are controlled using the digital outputs.



As you can deduce from the declaration line device they identify three basic elements that are controlled by device:

- the signals from the encoder to detect the speed of the belt
- the signal sent by the sensor piece presence it will also determine the length of the piece of material to be processed
- the heads of processing must be controlled so that fall when below is material and go up when this ends

## Calibration for the measurement of the speed tape

### Measure and pulse introduction

Let us consider the case (widespread) where the bi-directional transducer is a digital encoder directly keyed on the roller axis that moves the tape. You will need to set the correctly *measure* and *pulse* parameters of the device so that it can interpret the pulses arriving at QMove, the instrument then you can calculate the speed of the axis.

The *measure* and *pulse* introduction establishes a correspondence 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 lap then you can insert the following values:

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

The *measure* value introduced It also involves choosing a unit of measure of mm for measuring the positions, in the *pulse* parameter Yes it's introduced a value equal to the number of pulses is multiplied for 4. Remember that the *measure/pulse* relationship must be a value between 0.00935 to 1. 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:

- through the “device monitor” of QView see on the pc the *posit* parameter value
- set *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.

### Calculation of belt speed

The device, to calculate the speed at which it moves the tape, count the number of pulses received from the encoder for a short period of time (sampling time), dividing that value for the duration of the interval you get the average speed within the period.

The *tbtf* parameter allows you to choose the length of sampling time, was given this possibility to the user because, in the event that the tape moves at very low speed you will have to set a long time. Otherwise it could happen that in sampling time you do not receive sufficient samples to determine the speed.

It is recalled that, a long sampling time involves a rather slow speed update.

Was introduced also a filter software that can be used in the calculation of the band speed. This filter is lowpass and its time constant can be set with the *tfilter* parameter, You can also choose whether to enable this filter when the belt speed undergoes a change in a sample time exceeds a certain parameter. This parameter is set through *dvelf*. This filter returns to avoid a measured value of speed that oscillate too, It can be said that its function is to “mediate” the calculated speed.

The unit of measurement of the instantaneous speed of the tape is chosen through *unitvel* and *decpt* parameters. You can select the unit of time of speed with *unitvel* parameter: if this is equal to 0 then the speed is measured in Um/min, if is equal to 1 then the speed is measured in Um/s. The *decpt* parameter instead determines whether multiple speed-measuring of fundamental units Um. For example, if the fundamental unit of measure is Um=mm, and unitvel=1 you get the speed indicator in the vel in:

mm/s (with decpt = 0),  
cm/s (with decpt = 1),  
dm/s (with decpt = 2),  
m/s (with decpt = 3).

Later, if needed, we must properly configure the display to adjust the correct decimal point position.

### Calibrating presence piece sensor

The presence piece sensors have fixed positions in machine configuration: the first, at the start of the conveyor, has the task to identify the beginning and end of the piece, possibly can be used two other sensors placed at a certain distance from the beginning of the tape to make a “correction” of the position of the workpiece.

The *disti02* and *disti03* parameters allow you to set the distance from the sensors to 2 and 3 from first.

The first sensor will also identify the beginning of the conveyor, whenever this will activate the *posit* parameter and encoder is reset.

In case the sensor piece is a mechanical limit switch It may be necessary to introduce an advance or delay the end of piece than the falling edge of the signal. This can be set with the *offseti01* parameter.

The *beltlength* parameter allow sets the length of the tape, the device when will identify the presence of material will keep blower output.

## Setup working heads

The machining heads configuration is different from the simple introduction of a value in a parameter. In fact, you must use the appropriate commands (*WRITESET*, *WRITEPRG* and *WRITEVAR*), This is due to the internal implementation of the device.

The first parameters to be introduced will be the ones using the *WRITESET* command:

*dist* determines the distance between the first piece present sensor and the head center of work involved,

*mode* determines the function that will have the head,

the *riseadv* and *downlag* parameters configure the advances/delays rise/fall of heads with the way described in the manual of the device.

How to introduce the parameters is as follows:

```
[...]
Heads:headin = 2           ;head 2 configuration
Heads:dist = 100           ;set distance to 100 Um from the piece presence sensor
Heads:mode = 1             ;head set as the Sander
Heads:riseadv1 = 5         ;setting the advance head ascent related to correvell1 speed
Heads:riseadv2 = 7         ;setting the advance head ascent related to correvell2 speed
Heads:riseadv3 = 10        ;setting the advance head ascent related to correvell3 speed
Heads:downlag1 = 6         ;delaying descent head related to correvell1 speed
Heads:downlag2 = 8         ;delaying descent head related to correvell2 speed
Heads:downlag3 = 12        ;delaying descent head related to correvell3 speed
Heads:headout = 0          ;set headout to 0
WRITESET Heads             ;enter write command
WAIT Heads:headout EQ Heads:headin ;wait command executed
[...]
```

This procedure (choice of head to be set with the headin parameter, changing parameters, sending the command) must be done even when using read and write commands *WRITEPRG*, *READPRG*, *WRITEVAR*, *READVAR*.

Through *WRITEPRG* you can place the configurations for how head machining.

*WRITEVAR* instead reads of ongoing processing parameters.

The same mechanism of reading parameters works with the *READPIECE* command that lets you know of the information about a specific workpiece.

When you want to read continuously parameter values for all heads, you can simply use a for cycle to read and store it in an array the parameters in which you are interested:

```
[...]
FOR (gbCount=1, gbCount LE 8, 1) ;initializing for cycle
  Heads:headin = gbCount          ;configure all the heads from 1 to 8
  Heads:dist = gbCount * 10       ;equally spaced heads of 10 Um
  Heads:mode = 1                  ;all honing heads
  WRITESET Heads                 ;send write command
  WAIT Heads:headout EQ Heads:headin ;wait command executed
NEXT
[...]
```

```
[...]
FOR (gbCount=1, gbCount LE 8, 1) ;initializing for cycle
  Heads:headin = gbCount          ;configure all the heads from 1 to 8
  READVARSET Heads               ;send read command
  WAIT Heads:headout EQ Heads:headin ;wait command executed
  Heads:actriseadv = garray[1]    ;memorize the fix in advance the first position of an array
  Heads:actdownlag = garray[2]    ;store delay correction in the second position of an array
  Heads:actpiece = garray[3]      ;memorize the number of workpiece in the third position of an array
NEXT
[...]
```

Documento generato automaticamente da Qem Wiki - <https://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.