Table of Contents

DEVICE RECDATA

1. Introduction

The internal RECDATA device is the tool that resides in CPU that allows you to manage data logging type:

- Encoder counters
- Axis virtual counters (is the theoretical position that must take the axis)
- Analog outputs
- Following error axes
- Inputs state
- Outputs state

To record this data is exploited the RAM of the CPU, for this reason, to use this device, the percentage utilization of data memory (USER Data memory) total shall not exceed 50%. The device occupies the 50% of the RAM.

1.1 Installation

1.1.1 Device declaration in the configuration file (.CNF)

In the configuration file (.CNF), the BUS section must be declared so that you have the hardware resources required for the implementation of the RECDATA device.
In the INTDEVICE section of the .CNF file must be add the following definition:

It is necessary that each definition are present on the same line. In case you do not want to assign a resource, for example IntL, You must enter in the appropriate field the X string.

```
;---------------------------------
; Internal device declaration
;---------------------------------
INTDEVICE
;..
<DEVICE_NAME> RECDATA TCAMP QCTL1 QCTL2 IOUTA1 IOUTA2 INTL1 INTL2 ING1 ING2 OUT1 OUT2
```

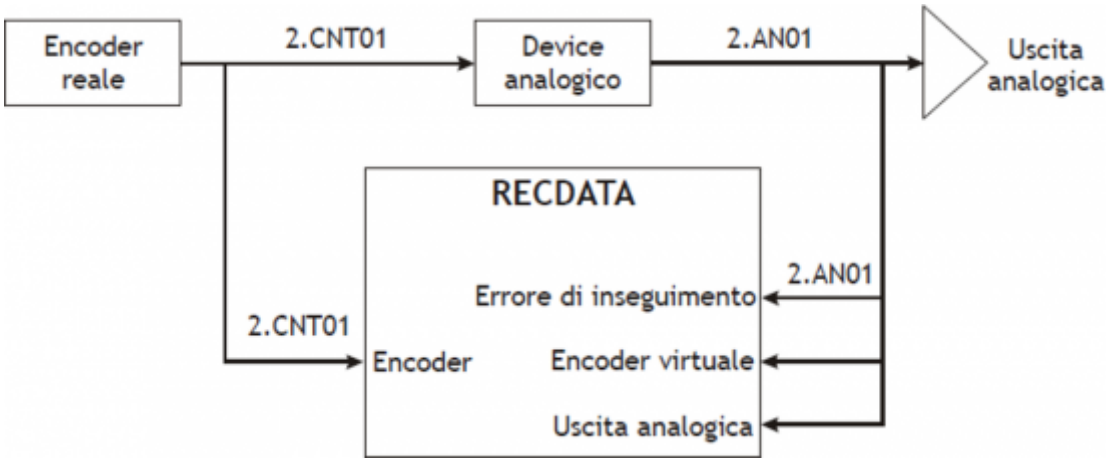
where:

<device name>	The name assigned to the device.
RECDATA	Keyword that identifies the devices.
TCamp	Tempo di campionamento device (1÷255 ms).
QCTL1	Address bi-directional meter 1 (to prevent the device uses this resource to put the X.X character).
QCTL2	Address bi-directional meter 2 (to prevent the device uses this resource to put the X.X character).
IOutA1	Hardware address of the DAC component of analog output 1. (to prevent the device uses this resource to put the X.X character).
IOutA2	Hardware address of the DAC component of analog output 2. (to prevent the device uses this resource to put the X.X character).
IntL1	Number of the interrupt line 1 (to prevent the device uses this resource to put the X character). This interrupt line can give the start to registration.
IntL2	Number of the interrupt line 2 (to prevent the device uses this resource to put the X character).
Ing1	Generic input 1
Ing2	Generic input 2
Out1	Generic output 1
Out2	Generic output 2

```
;---------------------------------
; Internal device declaration
;---------------------------------
INTDEVICE
Rec  RECDATA  2  2.CNT01  2.CNT02  2.AN01  2.AN02  1  2  2.INP01  2.INP02  2.OUT01  2.OUT02
```

1.2 Operation

Below is a diagram of the operation of the RECDATA device



The RECDATA device works only with the maximum resolution of the transducer (bit encoder * 4). In the case that the analog output declared in the configuration file is used by a device to analog positioning (CAMMING or EANPOS type), the monitor offers the display of the theoretical position of the axis in addition to the practical. The difference between the two counts is the following error of axis. The reference device for this information is the one that currently using the analog output hardware resource.
You can start recording via the interrupt input (`IntL1`)

1.2.1 Available steps for storing

The device uses 50% of CPU RAM memory, which is split into banks of 16 Bytes. The number of steps available for registration is indicated by *stepnum* parameter, which is read only, and is calculated with the formula:
stepnum = RAM available : 16
The 16 Bytes of the Bank (named step) are divided into 4 Long, named *data1*, *data2*, *data3* and *data4*. Formatting data in memory depends on how you set the mode parameter according to the following order:
State I/O
Encoder 1
Encoder 2
Analog output 1
Analog output 2
Following error 1
Following error 2
Virtual encoder 1
Virtual encoder 2
For the user it will be difficult to interpret the data obtained without going through the QVIEW (the development software has a dedicated tool). For this reason is not delved into the topic if not upon the customer's request.

1.3 Device error management

An error in the device is signaled by the *st_error* state.
When *st_error* is equal to 1, are present on the *errcode* variable the type of error occurred (see table) and in the *errvalue* variable an indication on the cause of the error

Code	Priority	Description
1	1	Modified mode parameter during registration

If the device goes wrong, in order to start working you have to clear the *st_error* status through the *RSERR* command.

1.4 Warning device management

The presence of a warning in the system camming is signaled by the *st_warning* state.
Being caused by a minor event and being guaranteed in this situation the management of device, the tool continues his work.
When *st_warning* is equal to 1, are present on the *wrncode* variable the type of warning intervened (see table) and in the *wrnvalue* variable an indication as to the cause of the warning.

Code	Priority	Description
1	0	Command not executed

To clear the *st_warning* state must be send the *RSWRN* command.

1.5 Commands and parameters table

1.5.1 Symbols used

The parameter name, condition or command is taken back to the left side of the table.

R
Indicates if the parameter or state is retentive (upon initialization of the device maintains the previously defined), or the state assumes upon initialization of the device.
If the device does not need to initialize the “R” field indicates the value that the parameter or state take at the power up of the card.
R = Retentive
0 = Upon initialization of the device the value is forced to zero.
1 = Upon initialization of the device the value is forced to one.
- = Upon initialization of the device is presented significant value.

D
Indicates the size of the parameter.
F = Flag
B = Byte
W = Word
L = Long
S = Single Float

1.5.1.1 Conditions

Describes all the conditions necessary so that the parameter is considered correct or because the command is accepted.
In some cases, limit values are specified for the acceptance of the parameter: if introduced any values outside the limits set, the data is however accepted; therefore appropriate controls of the application must be provided to ensure the proper functioning.
To execute a command, all the conditions must be met; otherwise, the command is not sent.

A
Indicates the **access mode**.
R = Read.
W = Write.
RW = Read / Write.

1.5.2 Commands

The commands were ranked by decreasing priority. For example, in the case of contemporary of *INIT* and *EMRG* commands, is acquired first the *INIT* command.

Name	Conditions	Description
STARTR	st_rec = 0 st_error = 0	**Start recording** Command the beginning of data acquisition. Activates the *st_reck* state.
STOPR	st_rec = 1	**Stop recording** Stops the registration process data. Reset the *st_rec* state.
READSTEP	stepnum>stepin>1	**Read step** You can read the step specified in stepin. The data is available when stepin=stepout. It is recommended to read when a recording is in progress (st_rec = 0).
INTENBL	capture>0	**Interrupt enable** Enable interrupt capture indicated in IntL1 in the configuration file. Upon arrival of the interrupt the devices starts recording that will be stored in delta. Activates the *st_intenbl* state.
INTDSBL	No	**Interrupt disable** Disable the interrupt capture. Reset the *st_intenbl* state.
RSCAPTURE	No	**Reset capture** Disable the *st_capture* state.
RSERR	No	**Reset error** Reset the *st_error* state.
RSWRN	No	**Reset warning** Reset the *st_warning* state.

1.5.3 Parameters

Name	D	R	A	Conditions	Description
stepnum	L	0	R	No	**Step number** Indicates the maximum number of steps possible to register. The value is automatically fixed when you turn on the system and cannot be changed. Valid range: 1 ÷ 999999
stepin	L	0	R-W	No	**Step input** Indicates the number of steps that the user wants to read with the READSTEP command. Valid range: 1 ÷ *stepnum*
stepout	L	0	R	No	**Step output** Indicates the number of steps available to read in data1, data2, data3 and data4 parameters. Valid range: -999999 ÷ 999999
data1	L	0	R	No	**Data number 1** Variable use. Inside are the values captured during recording. See dedicated chapter.
data2	L	0	R	No	**Data number 2** Variable use. Inside are the values captured during recording. See dedicated chapter.
data3	L	0	R	No	**Data number 3** Variable use. Inside are the values captured during recording. See dedicated chapter.
data4	L	0	R	No	**Data number 4** Variable use. Inside are the values captured during recording. See dedicated chapter.
capture	B	R	R-W	No	**Capture mode** Defines how to catch the interrupt on zero-pulse. **0**=Disable. **1**=One-shot on the rising edge. He was captured on the first rising edge of zero-pulse after enabling of st_intenbl. **2**=One-shot on falling edge. He was captured on the first falling edge of zero-pulse after enabling of st_intenbl. Valid range: 0 ÷ 2.
mode	B	R	R-W	st_rec = 0	**Recording mode** The recording mode is defined by a Byte whose individual bit, if set to 1, enable the acquisition. 2^0 = Encoder 1 2^1 = Encoder 2 2^2 = Analog output 1 2^3 = Analog output 2 2^4 = Following error 1 2^5 = Following error 2 2^6 = Virtual encoder 1 2^7 = Virtual encoder 2 For example, if you want to record the values of Encoder 1 and Encoder 2, you will set mode = 3
tbase	W	R	R	No	**Time base** View sample time programmed in the configuration file.
errcode	B	0	R	No	**Error code** Indicates the type of error intervened in the system. The code is valid only if st_error = 1 (See dedicated chapter) Valid range: 0 ÷ 100.
errvalue	B	0	R	No	**Error value** Indicates the cause of the error in the system. The code is valid only if st_error = 1 (See dedicated chapter) Valid range: 0 ÷ 100.
wrncode	B	0	R	No	**Warning code** Indicates the warning type intervened in the system. The code is valid only if st_warning = 1 (See dedicated chapter) Valid range: 0 ÷ 100.
wrnvalue	B	0	R	No	**Warning value** Indicates the cause of the warning intervened in the system. The code is valid only if st_warning = 1 (See dedicated chapter) Valid range: 0 ÷ 100.

1.5.4 States

Name	D	R	A	Conditions	Description
st_intenbl	F	0	R	No	**Interrupt enabled** Reports enabling the capture of start recording interrupt line. Is activated by the INTENBL command and disable by the INTDSBL command or on the rising edge of st_capture. **0** = Interrupt disable **1** = Interrupt enable. To power up by default is set to zero.

Name	D	R	A	Conditions	Description
st_capture	F	0	R	No	**Interrupt captured** Interrupt signal captured; is reset with the RSCAPTURE command. **0** = Not captured interrupt **1** = Captured interrupt To power up by default is set to zero.
st_int1	F	0	R	No	**Status of interrupt line 1** Indicates the status of the IntL1 interrupt line. **0** = Disable interrupt input. **1** = Active interrupt input. To power up by default is set to zero.
st_int2	F	0	R	No	**Status of interrupt line 2** Indicates the status of the IntL2 interrupt line. **0** = Disable interrupt input. **1** = Active interrupt input. To power up by default is set to zero.
st_error	F	0	R	No	**Status of device error** Indicates the error state in the device. To decode the error you must reference the errcode and errvalue variables. **0** = Error not present. **1** = Error present. To power up by default is set to zero.
st_warning	F	0	R	No	**Status of device warning** Indicates the warning state in the device. To decode the warning you must reference the wrncode and wrnvalue variables. **0** = Warning not present. **1** = Warning present To power up by default is set to zero.

1.6 Limitations

No limitation