

Índice

DEVICE FREQ	3
1. Introduction	3
2. Device declaration	4
3. Operation	5
3.1 Antiglitch filter	5
3.2 Edge detect	5
3.3 Edge counter	5
3.4 Upper limiter	5
3.5 Scaling	5
3.6 Low pass filter	5
3.6.1 Notes on operation of the device	5
4. Parameters table	7
5. States table	10
6. Commands table	11
7. Errors and alarms	12
7.1 Errors	12
7.2 Alarms	12

DEVICE FREQ

1. Introduction

FREQ is an internal device that allows you to measure the frequency of a digital signal. Also integrates a pulse counter one way a first-order low-pass digital filter on the measure, a function of scaling and a duty cycle.

The main features of the device are:

- selectable frequency resolution between 100th and 1/1000th of a hertz.
- forehead measurement signal acquisition selectable between ascent and descent.
- low-pass digital filter of first-order on the adjustable time constant measurement.
- scaling of the measurement by means of three programmable coefficients.
- acquisition of the duty cycle of the signal to be measured (for signals with frequencies below to 150 Hz).
- availability of frequency, measurement and measurement value filtered.

2. Device declaration

In order to use the device you must declare it in the appropriate INTDEVICE section of the configuration unit.

```
-----  
; Internal device declaration  
-----  
INTDEVICE  
<device_name>  FREQ  TCamp  IntL
```

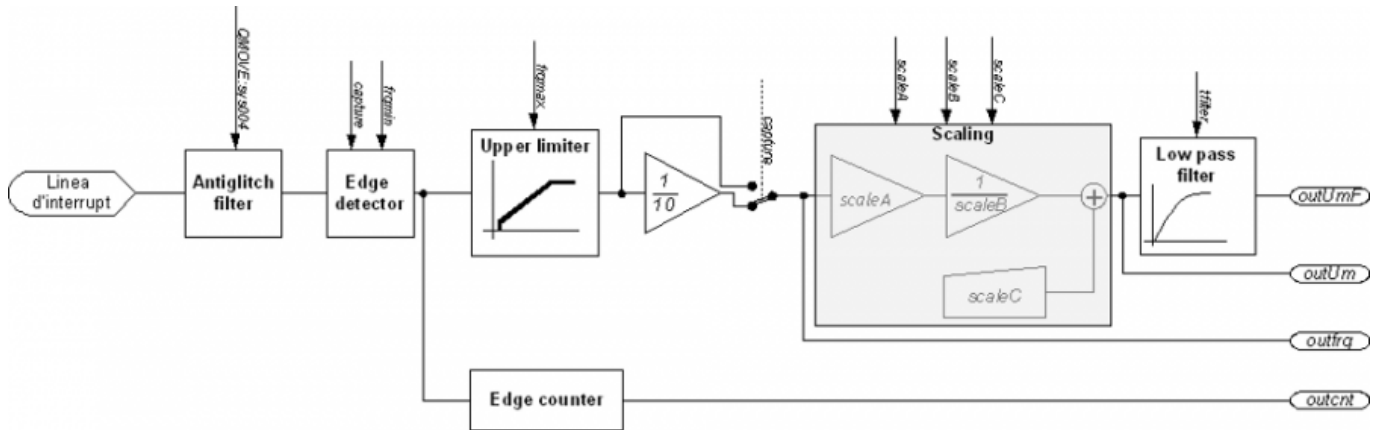
Where:

<device_name>	the name given to the device
FREQ	keyword that identifies the FREQ device
TCamp	sample time device in milliseconds (1÷250)
IntL	interrupt line associated to digital input to which is applied the signal to monitor



**Attention: All fields of the declaration are necessary and must be present on the same line.
Set "X.X" or "X" if a resource is not available or cannot be used.
Disable a resource means disable all functionality of the device using it.**

3. Operation



3.1 Antiglitch filter

The “antiglitch filter” block execute a digital filtering on the input signal for troubleshooting. It samples the input signal at regular intervals and determines the exit status by checking the status of the majority of the samples. The size of the filter depends on a parameter of the QMOVE system (“QMOVE:sys004” system variable).



The existence of the “antiglitch filter” block and it's features depend on the product that you are using. Consult the documentation relating to the product for more information.

3.2 Edge detect

The “edge detect” block capture the output signal from the antiglitch filter and measuring the time between two consecutive fronts to get the frequency measurement. The rising or falling is selected using capture parameter. The *frqmin* parameter determines the maximum time after which the block value becomes zero.

3.3 Edge counter

The “Edge counter” block creates a counter of faces captured by “edge detect” block.

3.4 Upper limiter

Il blocco “Upper limiter”, come dice la parola stessa, limita i valori di uscita del blocco “edge detect” fino a *frqmax*.

3.5 Scaling

The “scaling” block execute mathematical operations of multiplication, division and sum, of the block output “upper limiter” value with the *scaleA*, *scaleB* and *scaleC* parameters to get the value measured (*outUm*) in the desired entity.

The formula to get the measured value is:

$$outUm = (outbit * scaleA / scaleB) + scaleC$$

3.6 Low pass filter

The “Low pass filter” block execute a digital filter of the first order (type of the classic RC filter) on the measured value. The time constant of the filter can be set using *tfilter* parameter.


3.6.1 Notes on operation of the device

- Antiglitch filter has a facility to remove any disturbances on the input signal but it has the disadvantage of introducing a “noise” on the output signal. This noise is more evident when the input frequency is higher than 10KHz. In these cases, you can reduce the noise considerably, until you cancel it, simply by increasing the sampling time of the device. If the application permits, we recommend using sampling

times equal to or greater than 20ms.

- The device cannot function simultaneously as a frequency meter and duty cycle detector therefore, when the *capture* parameter is set to 5, the output value of the “edge detect” block represents the duty cycle expressed in parts per 10000 (5000 means duty of the 50.00%). The output value of the “edge detect” block goes directly to the *outfreq* parameter bypassing the “upper limiter” block; the “scaling” block produce outputs which could lose meaning. To use the filter of the first order with the duty cycle properly requires that the *scaleA* and *scaleB* parameters are equal to 1 and the *scaleC* parameter is equal to zero. Using the transmitter duty cycle works fine for low frequency signals (less than about 150 Hz).

4. Parameters table

Name	Dimension	Default value	Access type	Unit of measure	Valid Range	Write conditions	Description
frqmax	Long	Retentive	RW	Hz/100 or Hz/1000	0÷2147483647	-	Maximum frequency Is the maximum value that can take <i>outfrq</i> . For higher frequency signals to <i>frqmax</i> , <i>outfrq</i> is set equal to <i>frqmax</i> .
frqmin	Long	Retentive	RW	Hz/100 or Hz/1000	0÷2147483647	-	Minimum frequency Is the minimum detectable frequency from the device. For lower frequencies, the output of the "edge detect" block worth zero.  frqmin It also affects the time limit beyond which no input signal is detected; for example, by setting <i>frqmin</i> equal to 1Hz, the device detects no signal and then poses to zero <i>outfrq</i> after 1 sec. By setting, <i>frqmin</i> equal to 0.01 Hz the lack of the signal will be detected after 100 sec. By setting <i>frqmin</i> equal to 0 is how to set 0.01Hz if <i>outfrq</i> it is expressed in hundredths of hertz or 0.001Hz if <i>outfrq</i> is expressed in thousandths of hertz
capture	Byte	Retentive	RW	-	0÷5	-	Capture mode and resolution signal frequency measurement Defines the signal capture mode (front and frequency/duty) and resolution of frequency measurement (hundredths or thousandths of an hertz). 0 the "edge detect" and "edge counter" blocks are disabled so the device provides the value of the <i>outfrq</i> always equals zero while <i>outcnt</i> do not increase more. 1 the "edge detect" block capture the falling edge of the input signal and the value of <i>outfrq</i> is expressed in hundredths of Hertz. The "edge counter" block, also increases its output value <i>outcnt</i> on the falling edge of the input signal. 2 the "edge detect" block capture the rising edge of the input signal and the value of <i>outfrq</i> is expressed in hundredths of Hertz. The "edge counter" block, also increases its output value <i>outcnt</i> on the rising edge of the input signal. 3 the "edge detect" block catch the falling edge of the input signal and the value of <i>outfrq</i> is expressed in thousandths of Hertz. The "edge counter" block, also increases its output value <i>outcnt</i> on the falling edge of the input signal. 4 the "edge detect" block catch the rising edge of the input signal and the value of <i>outfrq</i> is expressed in thousandths of Hertz. The "edge counter" block, also increases its output value <i>outcnt</i> on the rising edge of the input signal.
scaleA	Long	Retentive	RW	-	-2147483648÷2147483647	-	Coefficient A in the Scaling function Is the parameter for which multiplies the frequency value detected by the "edge detect" block in the scaling function. To setting the parameter to 0 the measured value will always have value 0.

Name	Dimension	Default value	Access type	Unit of measure	Valid Range	Write conditions	Description
scaleB	Long	Retentive	RW	-	-2147483648÷2147483647	-	Coefficient B in the Scaling function Is the parameter for which divides the frequency value detected by the “edge detect” block in the scaling function. To setting the parameter to 0 the measured value will always have value 0.
scaleC	Long	Retentive	RW	-	-2147483648÷2147483647	-	Coefficient C in the Scaling function Is the parameter that is added to the result obtained by multiplying the value of the detected frequency from “edge detect” block multiplied by the ratio between parameters <i>scaleA</i> and <i>scaleB</i> of the scaling function.
tfilter	Word	Retentive	RW	ms	0÷32767	-	The filter time constant Time constant.  in the event that <i>tfilter</i> is less than or equal to the sampling time of the device is disabled, the filter and the <i>outUmF</i> parameter updates with the same value as <i>outUm</i>.
par01	Long	Retentive	RW	-	-2147483648÷2147483647	-	Generic parameter Parameter available for future implementations.
par02	Long	Retentive	RW	-	-2147483648÷2147483647	-	Generic parameter Parameter available for future implementations.
outfrq	Long	-	R	Hz/100 or Hz/1000	0÷frqmax	-	Frequency detected Is the output value of the “edge detect” block.
outUm	Long	-	R	-	-	-	Measured value Is the value of the detected frequency passed by the scaling function.
outUmF	Long	-	R	-	-	-	Measured filtered value Is the value of the low-pass digital filter applied.
outcnt	Long	-	RW	-	-	-	Pulse counter Is the output value of the “edge counter” block then the number of rising or falling (depending on the <i>capture</i> parameter) detected by the “edge detect” block.
outvar	Long	-	RW	-	-	-	Generic variable Parameter that is currently not implemented.
errcode	Byte	0	R	-	-128÷127	-	Error identification code Indicates the type of error occurred in the device. The code is valid only if <i>st_error</i> = 1. For more information see the dedicated chapter
errvalue	Byte	0	R	-	-128÷127	-	Identifying code of the cause of error Indicates the cause of the failure in the device. The code is valid only if <i>st_error</i> = 1. For more information see the dedicated chapter
wrncode	Byte	0	R	-	-128÷127	-	Identification code warning Indicates the type of warning in device. The code is valid only if <i>st_warning</i> = 1. For more information see the dedicated chapter

Name	Dimension	Default value	Access type	Unit of measure	Valid Range	Write conditions	Description
wrnvalue	Byte	0	R	-	-128÷127	-	Identification code of the cause of warning Indicates the cause of the warning in device. The code is valid only if <i>st_warning</i> = 1. For more information see the dedicated chapter

5. States table

Name	Default value	Description
st_01	0	Input state Source input state (interrupt line or frequency counter number).
st_02	0	Generic state Parameter available for future implementations.
st_error	0	Error presence Indicates the error status of the device, to recognize the type of error you must refer to <i>errcode</i> and <i>errvalue</i> variables: 0 : error not present, 1 : error present
st_warning	0	Presence of a warning Indicates the warning state of the device, to recognize the type of warning you must refer to <i>wrncode</i> and <i>wrnvalue</i> variables: 0 : warning not present, 1 : warning present

6. Commands table

Name	Conditions	Description
CMD01	-	Generic command Command available for future implementations.
CMD02	-	Generic command Command available for future implementations.
RSERR	-	Reset of the error state Reset the <i>st_error</i> state.
RSWRN	-	Reset of the warning state Reset the <i>st_warning</i> state.

7. Errors and alarms

7.1 Errors

When the device fails to perform operations are caused by incorrect programming by the user reports this condition through activation of the *st_error* state. The device, also, makes available, through the values information *errcode* and *errvalue* parameter, some information for a better understanding of the type of error and what condition the generated. This information and the error state *st_error*, continue until you run the appropriate *RSERR* command that will remove them. The following table specifies the values assumed by *errcode*:

errcode	Description
0	no errors
1	interrupt line associated to digital input not supported by hardware

Through the *errvalue* parameter you can get more detailed information about the error. The following table specifies the values assumed by *errvalue*:

errvalue	Description
0	no details about the error

7.2 Alarms

When the parameters of the device are programmed with out-of-range values they are not accepted and maintain their previous value. The device reports this condition through activation of the *st_warning* state. The device also provides, through the values parameter information *wrcode* and *wrnvalue*, some information to better understand the type of warning and what condition the generated. This information and the status of warning *st_warning*, continue until you run the appropriate *RSWRN* command that will remove them.

The following table specifies the values assumed by *wrcode*:

wrcode	Description
0	no warning

Through the *wrnvalue* parameter you can get more detailed information about the warning. The following table specifies the values assumed by *wrnvalue*:

wrnvalue	Description
0	no details about the warning

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.