ideiOriginal
sistemas de domótica e segurança, lda.

io.xec

User's manual

v0.8

Índice

1. Introduction

IO.XEC is a device that captures electrical signals, and transmits this information through a TCP/IP connection, in an Ethernet network. It implements the following TCP applications:

- Web server (port 80)
- Event server (port 8200)
- Command server (port 8201)
- Serial(RS232)-Ethernet bridge (port 9761)

2. Hardware description

Ports:

Ethernet
Serial (RS232)

Inputs:

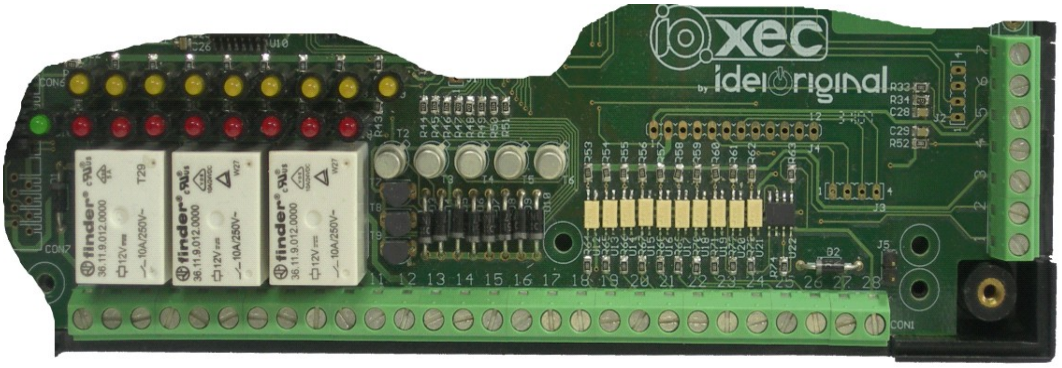
8 general purpose digital inputs (several configurations are possible)
1 fast digital input, designed for incremental encoders
1 analogue temperature input (the recommended sensor is LM35)
1 analog input (0-10V)

Outputs:

3 relay digital outputs
5 open collector digital outputs
1 analog output (0-10V)

2.1. Connection blocks

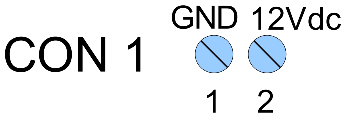
The connection blocks are accessible after opening the device's enclosure.



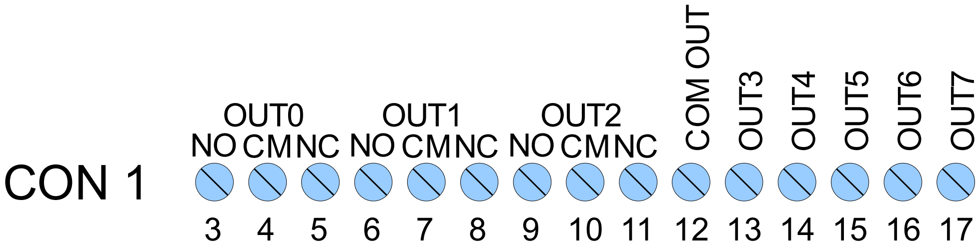
Bloco de terminais CON2

Bloco de terminais CON 1

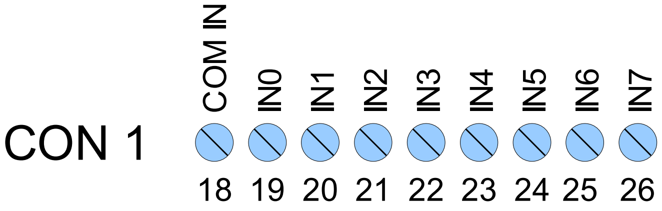
Power supply



Digital outputs

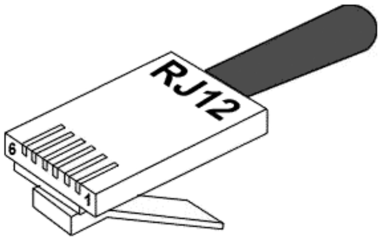


Digital inputs



2.2. Serial port (RS-232)

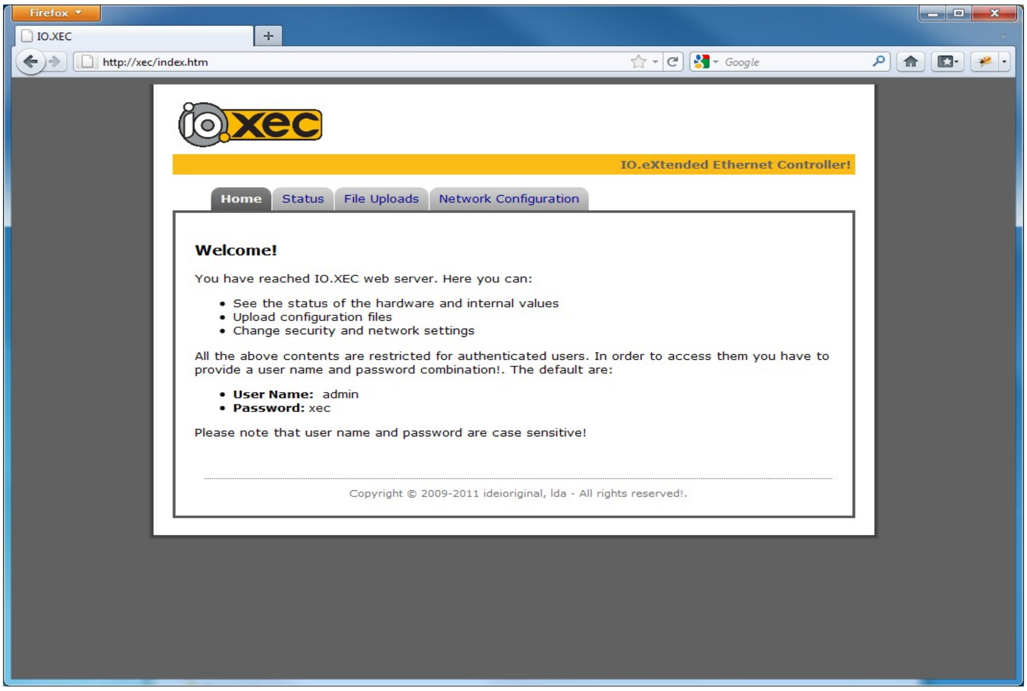
RJ11		RS232(DB9)
2	↔	5
4	↔	3
5	↔	2



4. Web server

The web server is available with a standard browser. The device's status and configuration can be viewed and changed accessing the various pages. The default username and password required to login are “admin” e “xec”, respectively.

Home page



Welcome!

You have reached IO.XEC web server. Here you can:

- See the status of the hardware and internal values
- Upload configuration files
- Change security and network settings

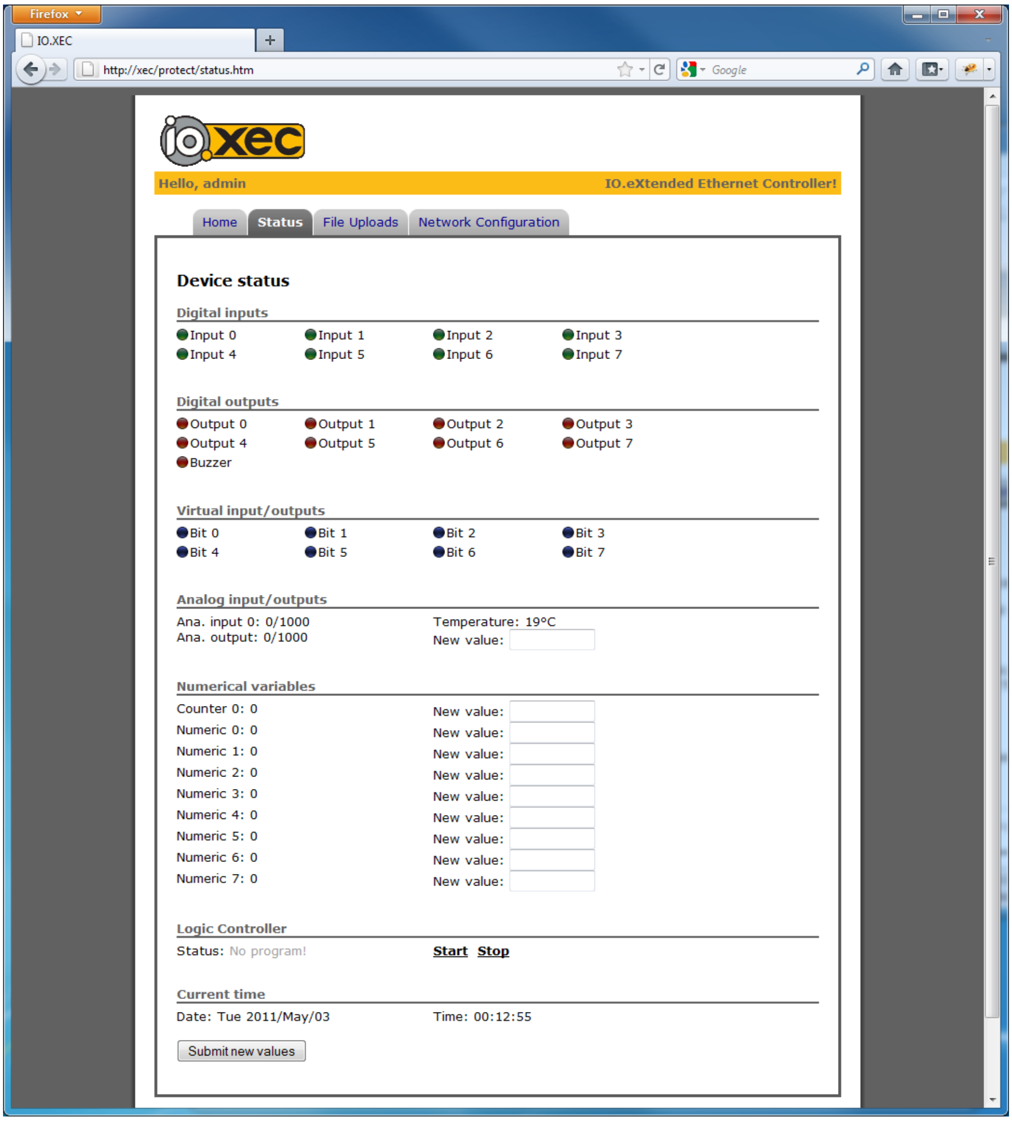
All the above contents are restricted for authenticated users. In order to access them you have to provide a user name and password combination!. The default are:

- **User Name:** admin
- **Password:** xec

Please note that user name and password are case sensitive!

Copyright © 2009-2011 ideioriginal, lda - All rights reserved!.

Variable status



Hello, admin IO.eXtended Ethernet Controller!

Device status

Digital inputs

● Input 0	● Input 1	● Input 2	● Input 3
● Input 4	● Input 5	● Input 6	● Input 7

Digital outputs

● Output 0	● Output 1	● Output 2	● Output 3
● Output 4	● Output 5	● Output 6	● Output 7
● Buzzer			

Virtual input/outputs

● Bit 0	● Bit 1	● Bit 2	● Bit 3
● Bit 4	● Bit 5	● Bit 6	● Bit 7

Analog input/outputs

Ana. input 0: 0/1000	Temperature: 19°C
Ana. output: 0/1000	New value:

Numerical variables

Counter 0: 0	New value:
Numeric 0: 0	New value:
Numeric 1: 0	New value:
Numeric 2: 0	New value:
Numeric 3: 0	New value:
Numeric 4: 0	New value:
Numeric 5: 0	New value:
Numeric 6: 0	New value:
Numeric 7: 0	New value:

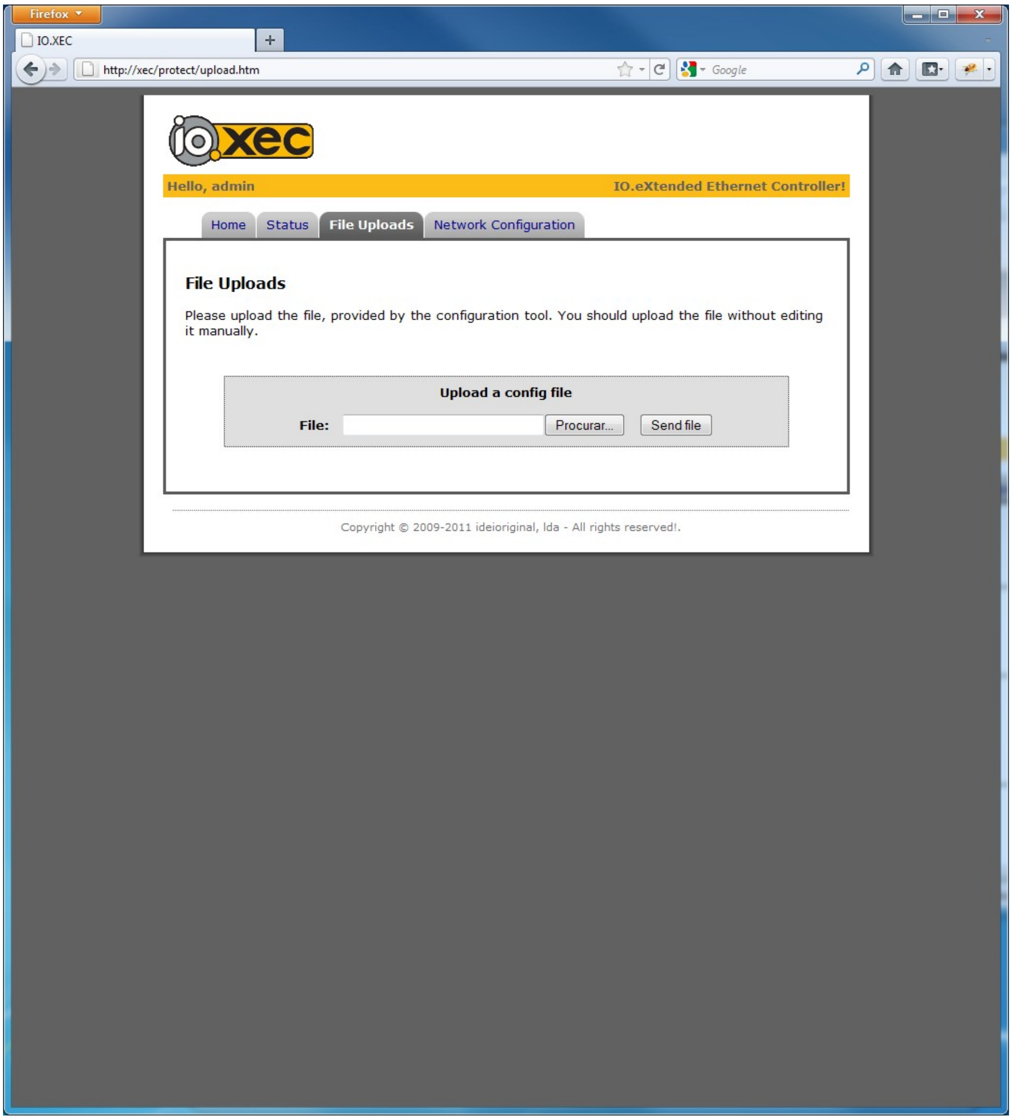
Logic Controller

Status: No program! **Start** **Stop**

Current time

Date: Tue 2011/May/03 Time: 00:12:55

Upload of configuration file



Hello, admin IO.eXtended Ethernet Controller!

File Uploads

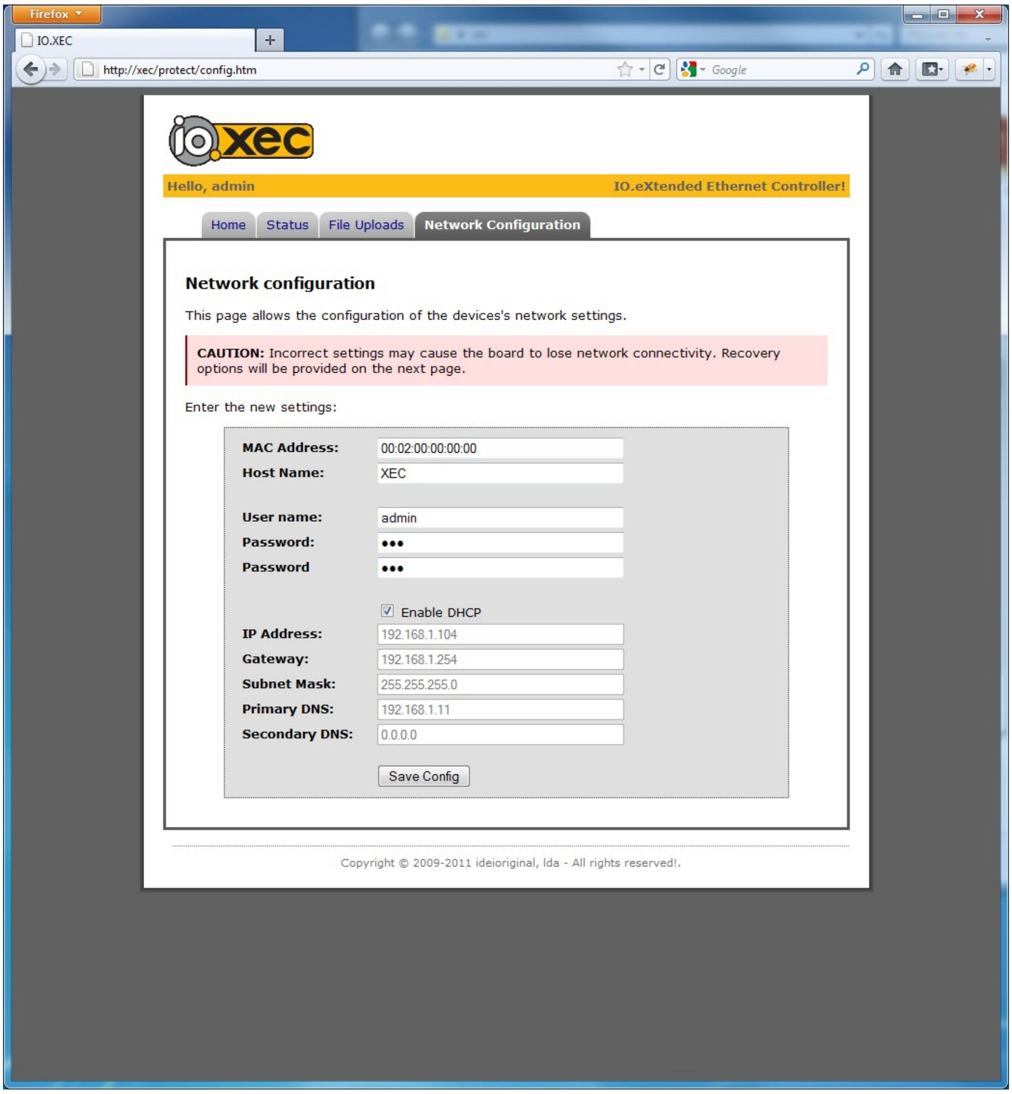
Please upload the file, provided by the configuration tool. You should upload the file without editing it manually.

Upload a config file

File:

Copyright © 2009-2011 ideioriginal, lda - All rights reserved!.

Network configuration



Network configuration

This page allows the configuration of the devices's network settings.

CAUTION: Incorrect settings may cause the board to lose network connectivity. Recovery options will be provided on the next page.

Enter the new settings:

MAC Address:	00:02:00:00:00:00
Host Name:	XEC
User name:	admin
Password:	•••
Password	•••
	☑ Enable DHCP
IP Address:	192.168.1.104
Gateway:	192.168.1.254
Subnet Mask:	255.255.255.0
Primary DNS:	192.168.1.11
Secondary DNS:	0.0.0.0

Copyright © 2009-2011 ideioriginal, lda - All rights reserved!.

5. Command server

The command server makes it possible to control and get status of the device. It allows to activate the outputs and get the status of all the variables. In order to access this service, a TCP socket must be created to the device, on port 8200. The default username and password required to login, are "admin" and "xec", respectively. The TCP port number is 8200. All the commands must be terminated with <!>.

The variables whose status can be retrieved, and activated, are:

Qx: digital outputs	x: 0..8
Bx: internal bits	x: 0..7
C0: high speed counter	
Nx: internal numerical variables	x: 0..7
AO: analog output	
Ix: digital input	x: 0..7
AI0: analog input	
AIT: analog temperature input	

5.1 Authentication

Syntax: LOGIN, user,pass<!>

user:	user name
pass:	password

5.2 Variable set

All the settable variables above, can be set with this command.

Syntax: SET,var,val[,timer]<!>

var:	Qx/Bx/C0/Nx/AO
val:	0/1/T: Qx/Bx -2147483648..2147483647: Nx/Cx 0..1000: AO
timer:	0..9999: Qx/Bx (in 1/10s)

5.2 Status

If the STAT command is issued, the status of all the variables is retrieved.

Syntax: STAT<!>

The response of the command has several fields, according to the variable type:

[IQB]=iiiiiiiiqqqqqqqqbbbbbbbb Digital variables

i...:	8 inputs
q...:	8 outputs
b...:	8 internal bits
i,q,b:	f: variable is false (inactive)
i,q,b:	t: variable is true (active)
i:	r: push button is released
i,q,b:	F: variable changed to false
i,q,b:	T: variable changed to true
i:	S: push button was pressed for a short time
i:	L: push button was pressed for a long time
i:	R: push button was pressed for a long time, and released

[NMx]=y Value of numerical variables

x:	0..7
y:	-2147483648..2147483647

[CN0]=y Value of the high speed counter

y:	-2147483648..2147483647

[AIx]=y Value of analog inputs

x:	0, T
y:	0..1000

[AO]=y Value of analog output

y:	0..1000

5.3 Keep connection alive

Like all embedded devices, the resources are limited. In order to save memory, IO.XEC only keeps connections active when needed. If no commands are sent during 120 seconds, the connection is closed. In order to keep the socket alive, a "keep alive" command must be issued. Note that this command is only needed if no other commands are sent for a 120 seconds period.

Syntax: KA<!>

5.3 Reboot

Reboots the device.

Syntax: REBOOT<!>

6. Event server

According to the configuration that is done (see section 10), IO.XEC sends information about the events detected on internal variables and hardware.

This server only implements 2 commands: "login" (see section 5.1) and "ka" (see section 5.3). The syntax is the same as described before.

When an event happens, IO.XEC sends a string like the "stat" (see section 5.2) command, but limited to the field containing the event. For example, if an event occurs in an input, only the [IQB] event is sent.

The TCP port number of this service is 8201.

7. IO.XEC responses

Welcome!	Connection successfully opened
[OK]	Command successfully executed
[ERR]:100	Invalid command
[ERR]:200	Argument error: invalid variable
[ERR]:300	Argument error: invalid value
[ERR]:400	Argument error: invalid number of arguments
[ERR]:500	Login error
[ERR]:800	Command not allowed
[ERR]:900	Login not performed

8. Serial server

The serial server is a transparent bridge between a TCP socket and a serial port (RS232). After a connection is established, all the bytes received on socket are dispatched through the serial port, and all the bytes that arrive on the RS232 port are forward to the TCP connection. The TCP port number is 9761. The baud rate of the serial port is selected using the configuration tool (see section 10).

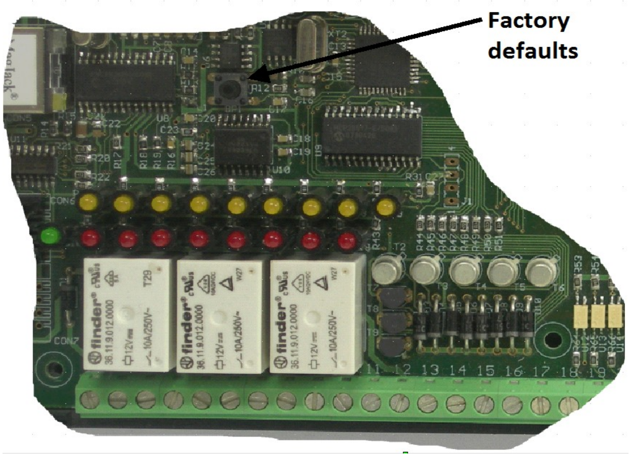
9. Factory defaults

When IO.XEC is connected for the first time, it has the following network configuration:

DHCP:	active
IP address:	169.254.1.1 (in case there is no DHCP server)
Subnet mask:	255.255.0.0 (in case there is no DHCP server)
NetBIOS name:	XEC
Username:	admin
Password:	xec

To restore factory default parameters, locate the push-button on the board and do the following:

- turn off IO.XEC
- press and hold the push-button
- turn on the device
- hold the push-button pressed for at least 5 seconds, during boot.

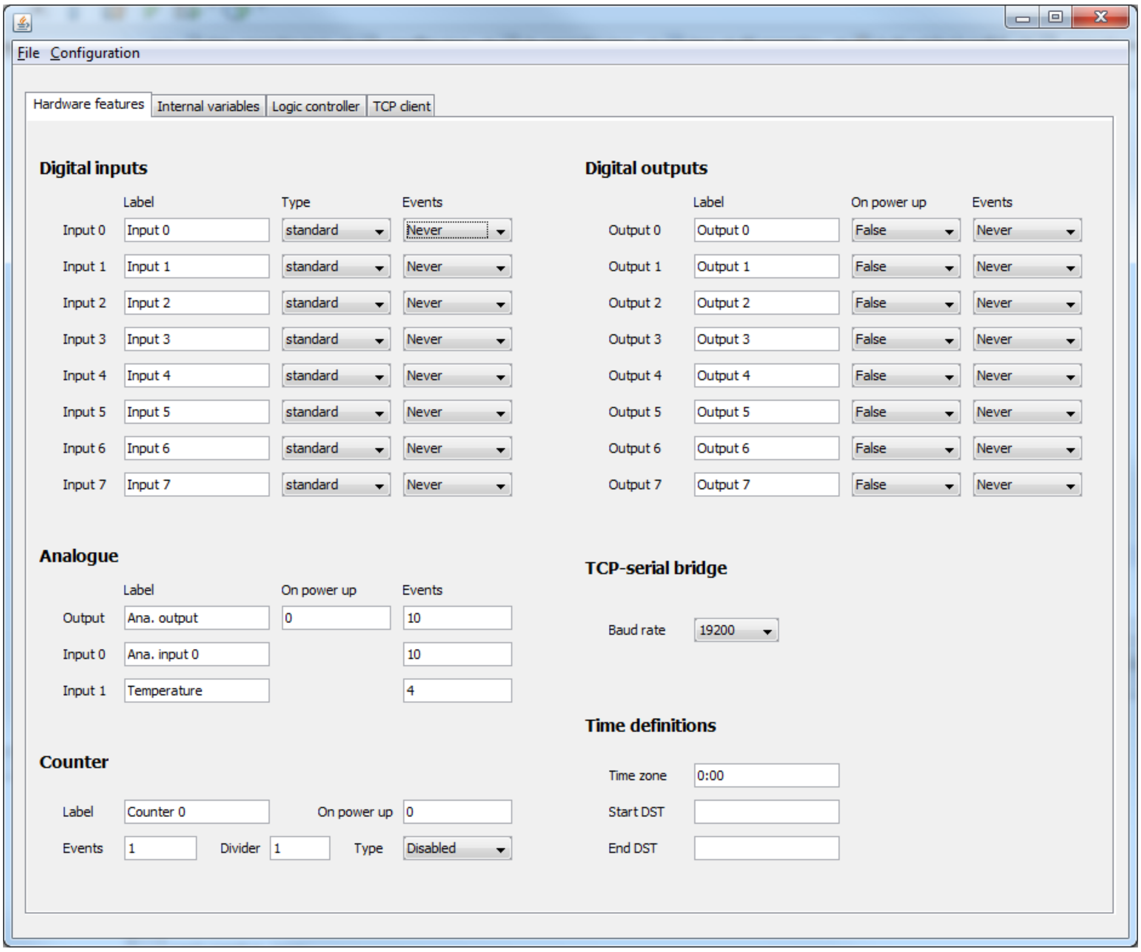


10. Configuration tool

The configuration tool, allows to:

- name the various variables, with suitable labels. These labels will be use to identify the variables in the web server
- define the type of hardware that is connected to the inputs
- configure the events that are sent through the event server
- define the baud rate of the serial port
- define the time zone properties
- implement some automation features
- send configuration to IO.XEC
- make test connections to the command and event server

10.1 Hardware features



Digital inputs

	Label	Type	Events
Input 0	Input 0	standard	Never
Input 1	Input 1	standard	Never
Input 2	Input 2	standard	Never
Input 3	Input 3	standard	Never
Input 4	Input 4	standard	Never
Input 5	Input 5	standard	Never
Input 6	Input 6	standard	Never
Input 7	Input 7	standard	Never

Digital outputs

	Label	On power up	Events
Output 0	Output 0	False	Never
Output 1	Output 1	False	Never
Output 2	Output 2	False	Never
Output 3	Output 3	False	Never
Output 4	Output 4	False	Never
Output 5	Output 5	False	Never
Output 6	Output 6	False	Never
Output 7	Output 7	False	Never

Analogue

	Label	On power up	Events
Output	Ana. output	0	10
Input 0	Ana. input 0		10
Input 1	Temperature		4

Counter

Label: Counter 0   On power up: 0

Events: 1   Divider: 1   Type: Disabled

TCP-serial bridge

Baud rate: 19200

Time definitions

Time zone: 0:00

Start DST:

End DST:

In this tab, the hardware is configured according to the application needs.

Inputs

In the inputs section, you can define, for each input:

- the label
- the type
  - standard (a regular contact)
  - push-button
- the events that are captured and sent through the event server

Analog

In the analog section, you can define:

- the label
- the value of the analogue output on power up
- the value variation that has to occur, in order to create an event

Counter

In this section, you can define:

- the label
- the value of the counter on power up
- the type of counter
- the value variation that has to occur, in order to create an event
- the divider (number of pulses detected to increment the counter)

Digital outputs

In this section, you can define:

- the label
- the value of each output on power up
- the events that are captured

TCP-serial bridge

Here you can define the baud rate of the RS232 port.

Time definitions

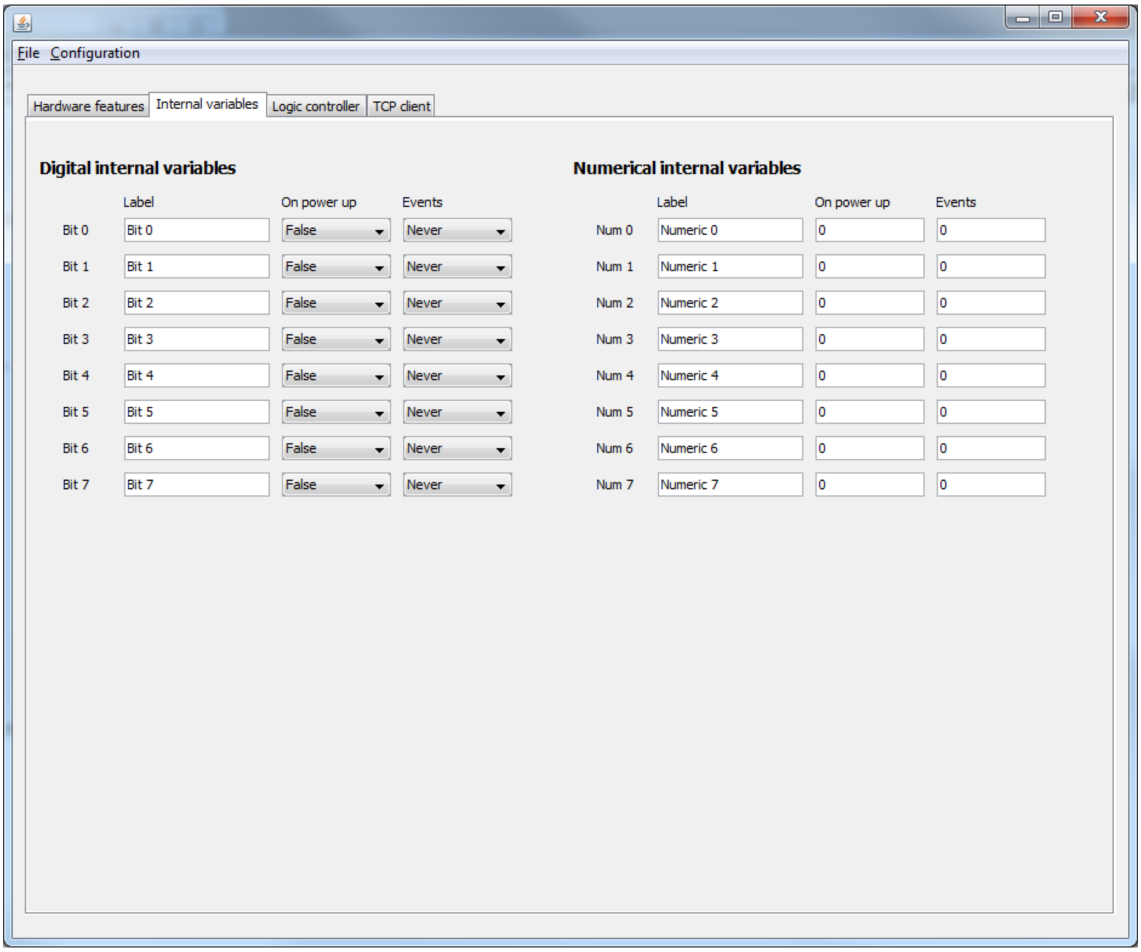
IO.XEC has an internal clock that synchronizes to ntp.org time server. In order to adjust local time settings it can be defined:

- the time zone, in terms of hours. For instance, if the device is located in:
  - Lisbon, 0:00 should be used
  - Paris, 1:00 should be used
  - Azores, -1:00 should be used
  - Helsinki, 2:00 should be used

- Daylight Saving Time (DST) start and end patterns. These patterns have four sections, separated by a dot (.). You can setup the day, month, day of week, and hour where the DST start or end takes place.

- The syntax is: <month>.<day>.<day of week>.<search>. For example:
  - 04.01.6+2.2:00: start DST on the second Sunday of April, at 2:00
  - 03.31.6-1.2:00: start DST on the last Sunday of March, at 2:00
  - 03.31...2:00: start DST on the 31th March, at 2:00
  - 10.31.6-1.2:00: end DST on the last Sunday of October, at 2:00

10.2 Internal variables



Digital internal variables

	Label	On power up	Events
Bit 0	Bit 0	False	Never
Bit 1	Bit 1	False	Never
Bit 2	Bit 2	False	Never
Bit 3	Bit 3	False	Never
Bit 4	Bit 4	False	Never
Bit 5	Bit 5	False	Never
Bit 6	Bit 6	False	Never
Bit 7	Bit 7	False	Never

Numerical internal variables

	Label	On power up	Events
Num 0	Numeric 0	0	0
Num 1	Numeric 1	0	0
Num 2	Numeric 2	0	0
Num 3	Numeric 3	0	0
Num 4	Numeric 4	0	0
Num 5	Numeric 5	0	0
Num 6	Numeric 6	0	0
Num 7	Numeric 7	0	0

Digital internal variables

They act as virtual IOs. It can be configured:

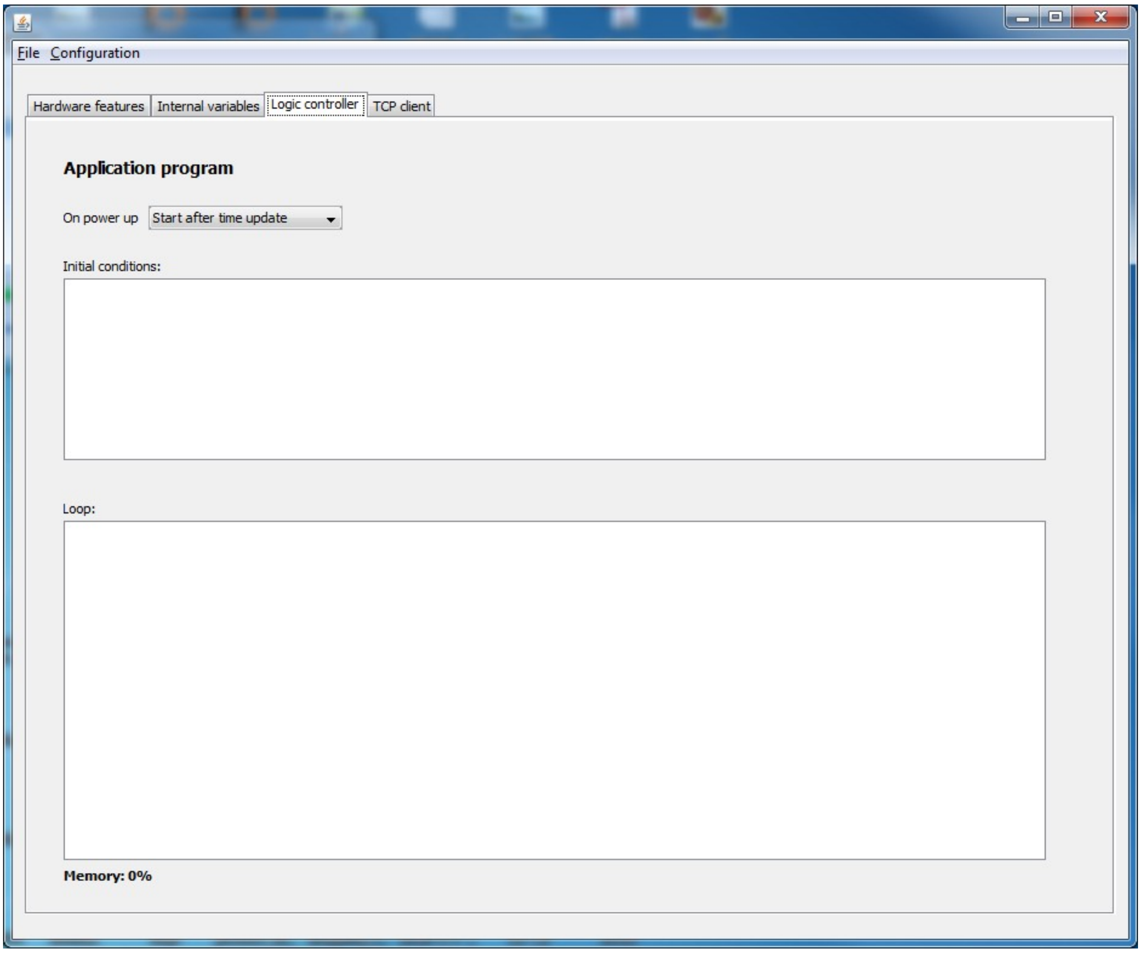
- the label
- the value on power up
- the events that are captured

Numerical variables

These variables have length of 32bit (signed). It can be defined:

- the label
- the value on power up
- the value variation that has to occur, in order to create an event

10.3 Logical Controller



Application program

On power up: Start after time update

Initial conditions:

Loop:

Memory: 0%

The logical controller allows to implement automation functions. The device can perform calculations logical combinations of the various variables (internal and external).

Variables

1. Boolean

Ix	Digital input Ix, x: 0..7
Qx	Digital output Qx, x: 0..7
Bx	Internal bit Bx, x: 0..7

-

2. Numerical

Nx	Numerical internal variable Nx, x: 0..7
C0	High speed counter
AO	Analog output (0..1000)
AI0	Analog input (0..1000)
AIT	Temperature analog input (0..90)
TS	Internal clock: current second
TM	Internal clock: current minute
TH	Internal clock: current hour
DD	Internal clock: current day
DM	Internal clock: current month
DY	Internal clock: current year
DW	Internal clock: current week day

Function EVENT

This function detects an event on a digital IO or internal bit. Returns true when the event occurs

Syntax: EVENT(var,event)

var: Ix/Qx/Bx

event: T: variable turns to **true**
F: variable turn to **false**
S: when a **short** push occurs in a push button
L: when a **long** push occurs in a push button
R: when a push button is **released**, after a long push

Function IS

Returns true if the logical value of a digital IO or internal bit equals the value parameter. Returns false, otherwise.

Syntax: IS(var,value)

var: Ix/Qx/Bx

event: **T: variable is true**
F: variable is **false**
S: during a **short** push in a push button
L: during a **long** push in a push button
R: after a push button is **released**

Function COMPARE

Returns the result (true or false) of the comparison between two numerical variables, using a comparison operator.

Syntax: COMPARE(var1,operator,var2)

var1: Nx/C0/AI0/AIT/AO/TS/TM/TH/DY/DM/DD/DW

var2: Nx/C0/AI0/AIT/AO/TS/TM/TH/DY/DM/DD/DW/x(32bit value)

event: >: returns true, if var1 is **greater** than var2
<: returns true, if var1 is **less** than var2
>=: returns true, if var1 is **greater than or equal** to var2
<=: returns true, if var1 is **less than or equal** to var2
!=: returns true, if var1 is **different** var2

Function SET

If a true condition is met, assigns a value to a variable. This functions has 3 variations, depending on the variables and parameters used.

Syntax 1: SET(var)

Sets the value of the condition to var

var: Ix/Qx/Bx

Syntax 2: SET(var,value[,tmr])
Sets var to value, if the condition is true. Does nothing otherwise.

var: Ix/Qx/Bx

value: 0 (false) / 1 (true) / T (toggle)

tmr: timer in 1/10s. optional.

Syntax 3: SET(var1,var2)
Sets var1 with the value of var2, if the condition is true. var2 can contain simple calculations, such as N4+10; C0*N1. Does nothing otherwise.

var1: Nx/C0/AI0/AIT/AO/TS/TM/TH/DY/DM/DD/DW

var2: Nx/C0/AI0/AIT/AO/TS/TM/TH/DY/DM/DD/DW/x(32bit value)
simple calculations allowed.

Syntax

The syntax is very simple. The automation concept is organized in lines, each one divided in two sections (left and right).

The lines are executed one after the other. When the processor reaches the lats lien, it jumps to the first one. While the logic controller is active, this process is repeated endlessly.

As stated, each line has two sections: left and right. On the left, the combination of events and status is performed using the functions IS, EVENT and COMPARE and the following operators:

- *: logical AND
- +: logical OR
- ^: logical XOR
- (): parenthesis to implement priorities in the evaluation

On the right a SET operation must be defined. According to the syntax used, an action is performed. Some examples ar provided:

- Toggle Q0, when I0 is active:
  event(i0,t)=set(q0,t)
- If I0 is true and I1 or I2 are true, activate Q1. Deactivate Q1 otherwise
  is(i0,t)*(is(i1,t)+is(i2,t))=set(q1)
- Blink an output every second:
  compare(ts,!=,ts)=set(q1,1,5)
- Count the number of seconds that an input is active
  is(i3,t)*compare(ts,!=,ts)=set(n0,n0+1)

Multiple lines can be added to achieve the required automation. The number of lines is limited by the available memory. The configuration tool indicates how much memory is occupied by the written lines.

Two blocks can be added. The first block is executed one time, in order to initialize the process. The second is iterated.

The behavior of the logic controller on power up, can be one of the following:

- Don't start: It can be started through the web page.
- Start immediately: After the reboot, as soon as possible
- Start after a successful time update. Waits until the NTP server is available and an update to the current time is performed. It needs an Internet connection

The logic controller status is shown in the web server.

11. Technical specifications

Power	12V/1A
Ethernet	10T/100T base @ 10Mbit/s
Serial port (RS-232)	4800bps / 9600bps / 19200bps / 57600bps / 115200bps no parity, no flux control
Relay outputs	6A/230Vac
Open collector outputs	12V/150mA
Inputs	5-24Vdc, no polarity
High speed counter input	Incremental encoder
High speed counter power	10-30Vdc
Temperature sensor	LM35
Analog input	0-10Vdc
Analog output	0-10Vdc