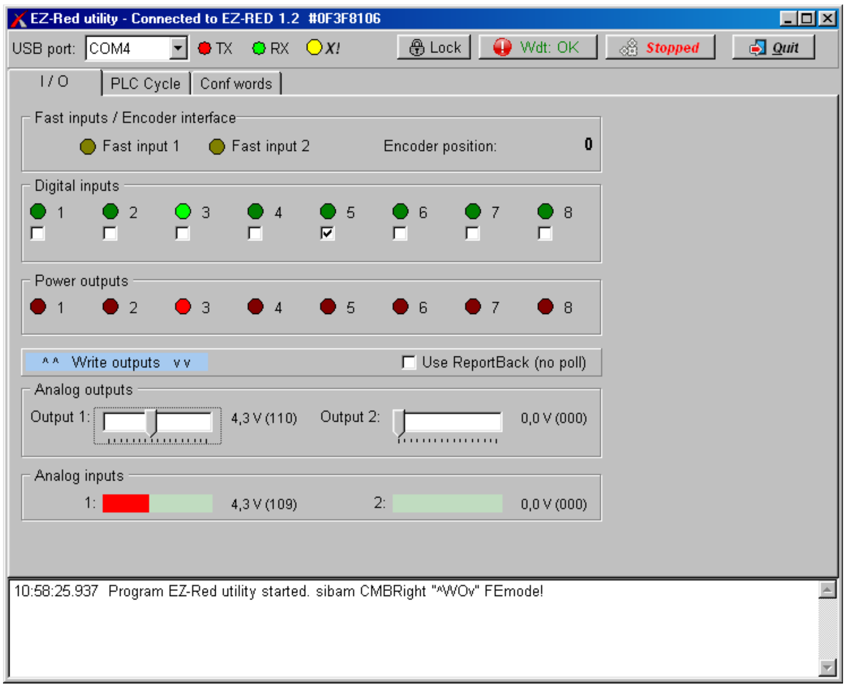
EZ-Red – Power I/O module

TSMon utility user manual
See http://www.xonelectronics.it for other manuals

Index

Introduction

TSmon is a Windows program used to test and manage the EZ-Red I/O module, and to compile and transfer PLC cycles (programs) to the module. It presents itself with a single window containing three tabs; the first:



EZ-Red utility - Connected to EZ-RED 1.2 #0F3F8106

USB port: COM4

Encoder position: 0

Digital inputs: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☑ 5 ☐ 6 ☐ 7 ☐ 8

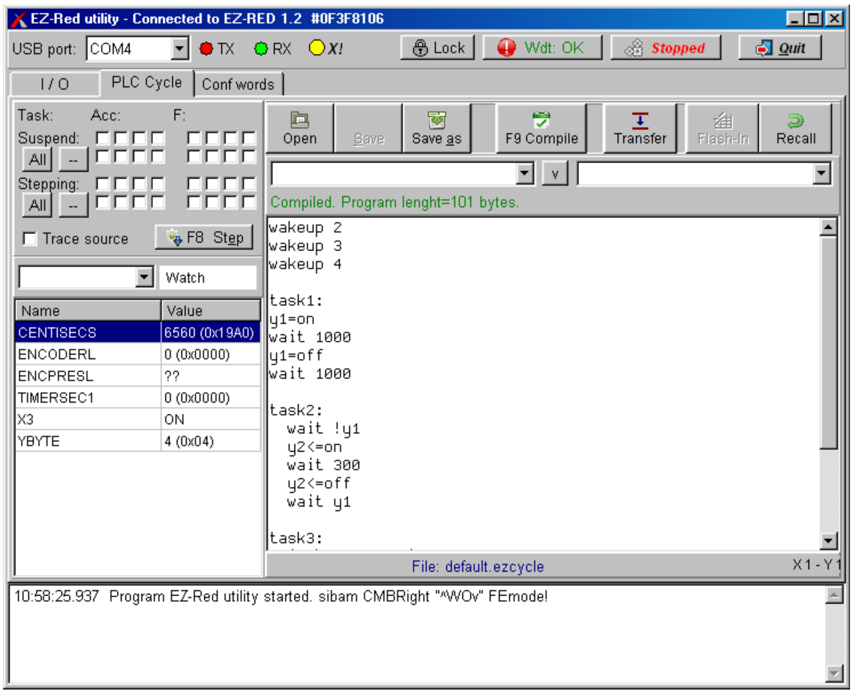
☐ Use ReportBack (no poll)

Analog outputs: Output 1: 4,3 V (110) Output 2: 0,0 V (000)

Analog inputs: 1: 4,3 V (109) 2: 0,0 V (000)

```
10:58:25.937  Program EZ-Red utility started. sibam CMBRight "^WOv" FEmode!
```

is used to watch the inputs and modify the outputs of the module; the second tab:



EZ-Red utility - Connected to EZ-RED 1.2 #0F3F8106

USB port: COM4

Compiled. Program lenght=101 bytes.

Name	Value
CENTISECS	6560 (0x19A0)
ENCODERL	0 (0x0000)
ENCPRESL	??
TIMERSEC1	0 (0x0000)
X3	ON
YBYTE	4 (0x04)

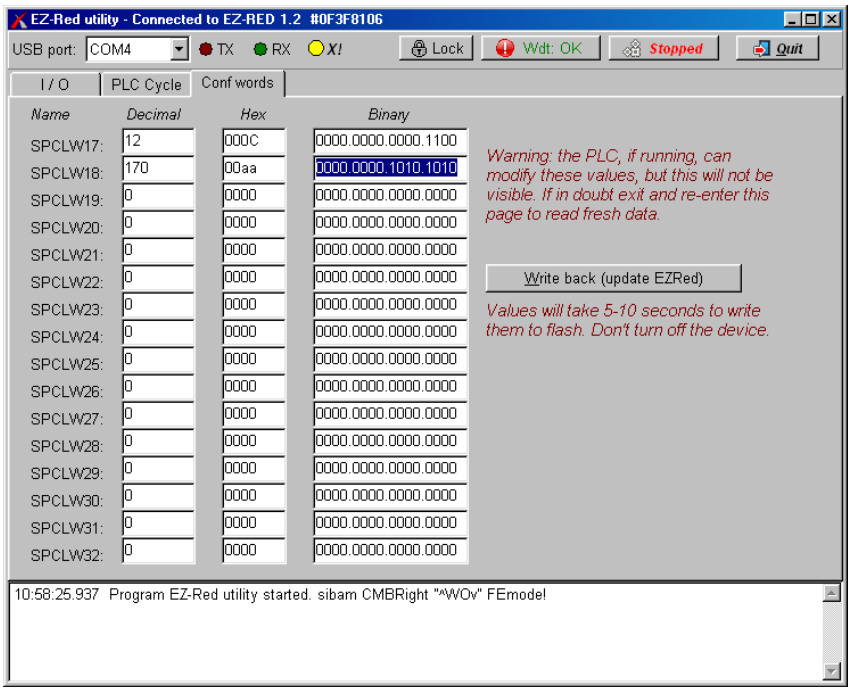
```
wakeup 2
wakeup 3
wakeup 4

task1:
y1=on
wait 1000
y1=off
wait 1000

task2:
  wait !y1
  y2<=on
  wait 300
  y2<=off
  wait y1

task3:
```

File: default.ezcycle X1 - Y1

```
10:58:25.937  Program EZ-Red utility started. sibam CMBRight "^WOv" FEmode!
```

is used to write, compile, debug, send to the module and store in it PLC cycles.

The third tab is used to review and sets configuration variables:



Name	Decimal	Hex	Binary
SPCLW17:	12	000C	0000.0000.0000.1100
SPCLW18:	170	00aa	0000.0000.1010.1010
SPCLW19:	0	0000	0000.0000.0000.0000
SPCLW20:	0	0000	0000.0000.0000.0000
SPCLW21:	0	0000	0000.0000.0000.0000
SPCLW22:	0	0000	0000.0000.0000.0000
SPCLW23:	0	0000	0000.0000.0000.0000
SPCLW24:	0	0000	0000.0000.0000.0000
SPCLW25:	0	0000	0000.0000.0000.0000
SPCLW26:	0	0000	0000.0000.0000.0000
SPCLW27:	0	0000	0000.0000.0000.0000
SPCLW28:	0	0000	0000.0000.0000.0000
SPCLW29:	0	0000	0000.0000.0000.0000
SPCLW30:	0	0000	0000.0000.0000.0000
SPCLW31:	0	0000	0000.0000.0000.0000
SPCLW32:	0	0000	0000.0000.0000.0000

Warning: the PLC, if running, can modify these values, but this will not be visible. If in doubt exit and re-enter this page to read fresh data.

Write back (update EZRed)

Values will take 5-10 seconds to write them to flash. Don't turn off the device.

```
10:58:25.937  Program EZ-Red utility started. sibam CMBRight "^WOv" FEmode!
```

These configuration variables are “resources”, in EZ-Red terminology, which remember their content across sessions (power-off / power-on). These variables can be managed by the PC as well as by the PLC cycle, in order to configure it or remember data for subsequent sessions.

Installation

FTDI Driver

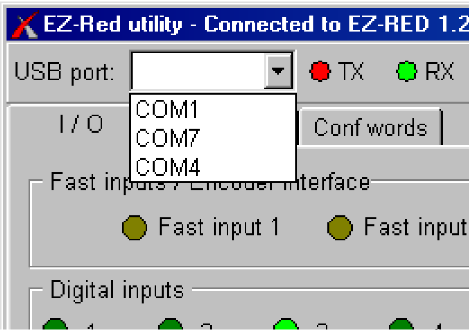
The correct Windows drivers are required in order to connect the device to the computer. These drivers can be downloaded from the chip producer web site (www.ftdichip.com) or XON Electronics web site in the EZ-Red page. Please refer to the User manual for details.

Needed files

TSmon needs to load the shared library EZRedDll.dll, which comes with the program, and a file named “tsmon.inc”. Both files, together with the executable itself “Tsmon.exe”, can be put in the same directory. After the first execution a new file “tsmon.ini” will be created.

USB interface

To instantiate the communication the EZ-Red must be powered and connected to an USB port of the PC; then run **Tsmon.exe** and choose the (virtual) communication port among the available ones:



EZ-Red utility - Connected to EZ-RED 1.2

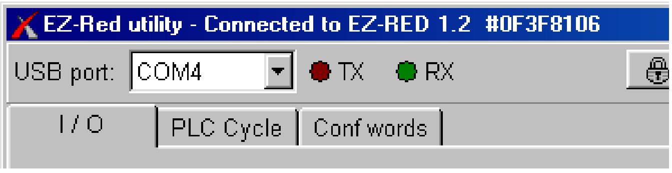
USB port:

The “USB port” field lists the available ports. This list is automatically refreshed when the field is expanded, so it is possible to run Tsmon first, and then connect a device.

If an EZ-Red is powered and connected to the computer, and the USB is correctly configured, the two leds TX and RX will start to blink after choosing the right port. If this does not happen, check in order:

1. Is EZ-Red powered on and connected to a working PC USB port?
2. Are the FTDI drivers correctly installed?
3. Does the operating system detect the new hardware?
4. Are the optional settings of the driver correct? (The default settings are ok).

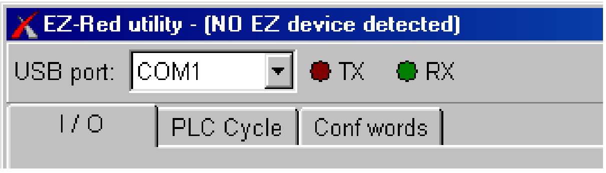
If in doubt on which port to choose in the “USB port” field, it is possible to try every single listed port. After having chosen a port, the program tries to initiate a communication; if the module is detected correctly, the window shows a message with device name and serial number:



EZ-Red utility - Connected to EZ-RED 1.2 #0F3F8106

USB port: COM4

If no EZ-Red can be detected, this different message is shown:

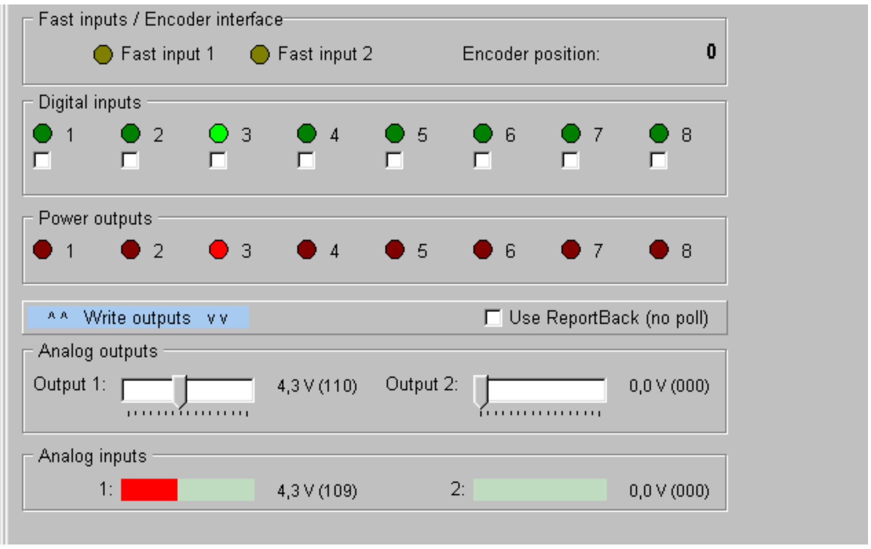


EZ-Red utility - (NO EZ device detected)

USB port: COM1

I/O tab

The I/O tab shows the module's inputs and outputs:



A few frames are visible:

- Fast inputs / Encoder interface.
  Shows the status of the two “Fast inputs”, and the position of the encoder if connected to these inputs. If voltage is present to an input, its color turns from dark yellow to bright yellow.

  To zero the encoder position, simply double click on the number (to the extreme right of the frame).

- Digital inputs.
  Shows the status of the 8 digital inputs. Their color is dark green if the input is off (no voltage), and bright green when it is ON (voltage applied). Click on the leds to invert and “force” its status. When an inputs is “forced”, EZ-Red uses the forced, simulated state instead of the real one. The check-box under the led is checked when the input is forced, and can be un-checked to return the input to its normal state.

- Power outputs.
  Shows, with dark or bright red, the status of the power outputs; click on these leds to invert the output status. Differently from inputs, outputs can not be forced – a running PLC cycle can always set or reset them.

- Analog outputs.
  Contains two cursors which show, and can be used to set, the voltage (0-10 volts) on the analog outputs. If a PLC cycle is running, the cursors can be moved by the cycle.

- Analog inputs.
  Shows the voltage at the analog inputs with a colored bar and two numbers: the real voltage from 0 to 10 volts and the corresponding value from 0 to 255.

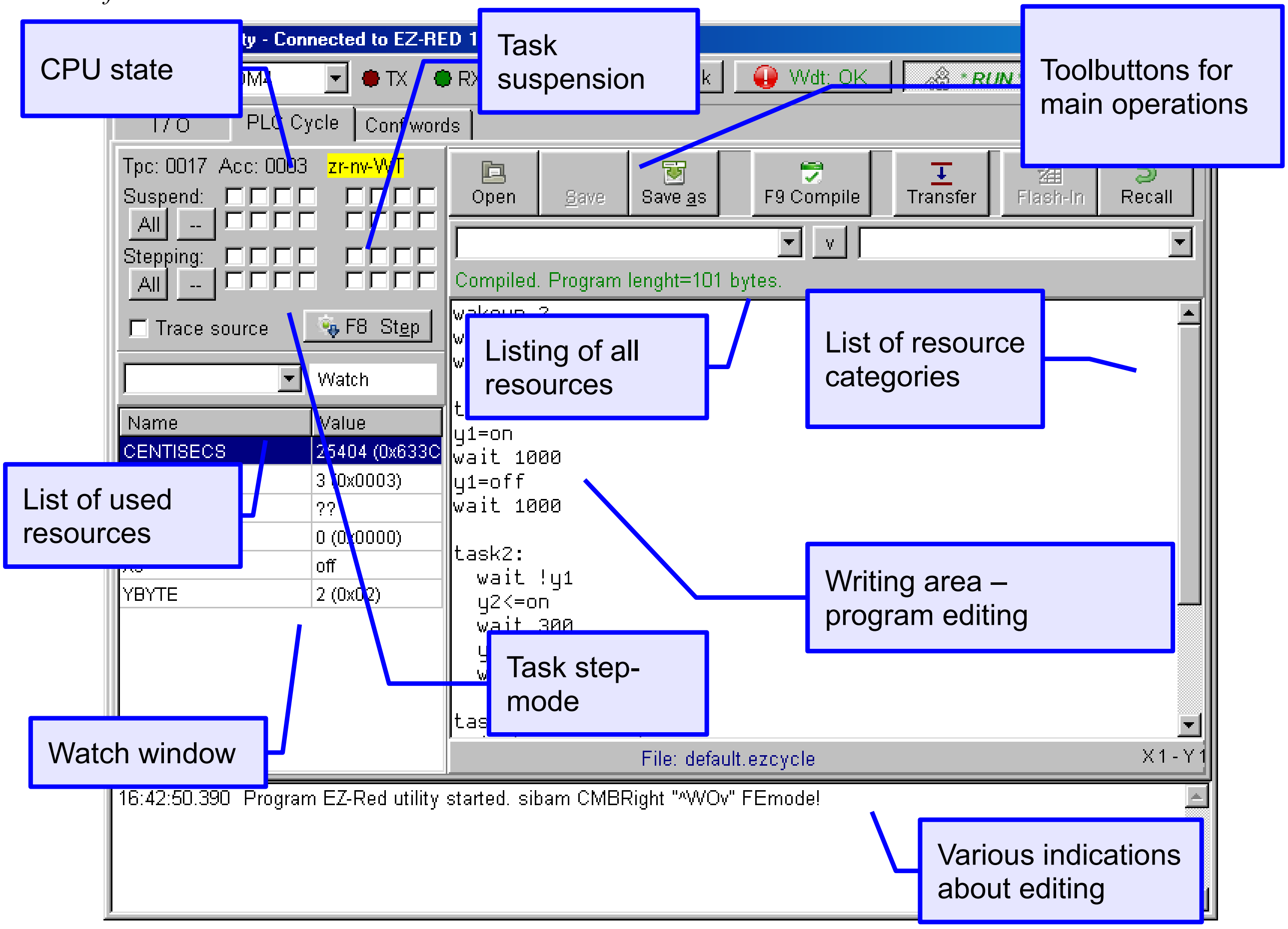
The checkbox "Use ReportBack (no poll)" indicates the strategy used to update the input status on screen. When unchecked, Tsmon cyclically polls the device every about 50 ms; input variations faster than that sometimes are not detected. This kind of polling is not optimal, but simple in concept. When the box is checked, instead, EZ-Red sends automatically the new input states to the computer, which can show them immediately (see REPORTBACK in the documentation).

PLC cycle tab

Introduction

In this tab there is all is needed to write a PLC cycle, compile, transfer to module, execute to debug, and save it permanently into the device flash memory. Actually, Tsmon is a complete IDE (integrated development environment).

This manual does not document the EZ-Red programming language: please refer to EZ-Red – Programming manual" for this.



Tpc: 0017 Acc: 0003 zr-nv-WT

Name	Value
CENTISECS	25404 (0x633C
	3 (0x0003)
	??
	0 (0x0000)
	off
YBYTE	2 (0x02)

Compiled. Program lenght=101 bytes.

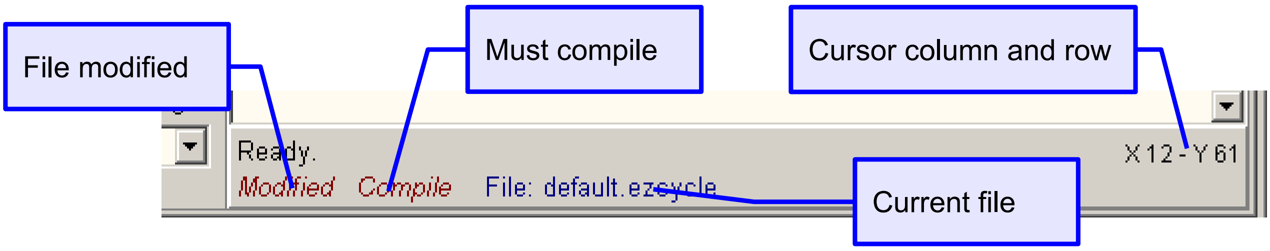
```
wakeup 3
y1=on
wait 1000
y1=off
wait 1000

task2:
  wait !y1
  y2<=on
  wait 300
```

File: default.ezcycle X 1 - Y 1

16:42:50.390 Program EZ-Red utility started. sibam CMBRight "^WOv" FEmode!

Disk archives

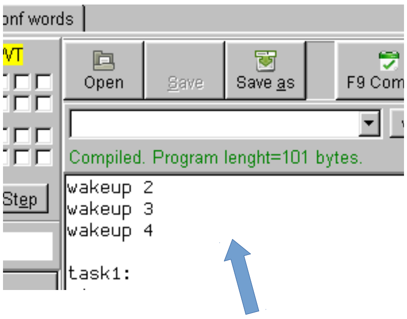
PLC cycles (programs) can be loaded from disk and saved to it. These files are normal text files, with extension **.ezCycle**; on Tsmon start-up, if present, the file "default.ezCycle" is automatically loaded. All the editing operations are made in the writing area, and the status of this area is shown below it:



Ready.

Modified Compile File: default.ezcycle X 12 - Y 61

The button bar contains three buttons to load, save and save with different name:



Compiled. Program lenght=101 bytes.

```
wakeup 2
wakeup 3
wakeup 4

task1:
```

"Open" button loads a file from disk. If the current file has been modified but not saved, a confirmation is asked to discard the editing.

The "Save" button writes the current file to disk, using the current name; if the current file is not modified, the button is disabled.

"Save as" writes the current file to disk, allowing to choose a different name than the current one.

Writing area

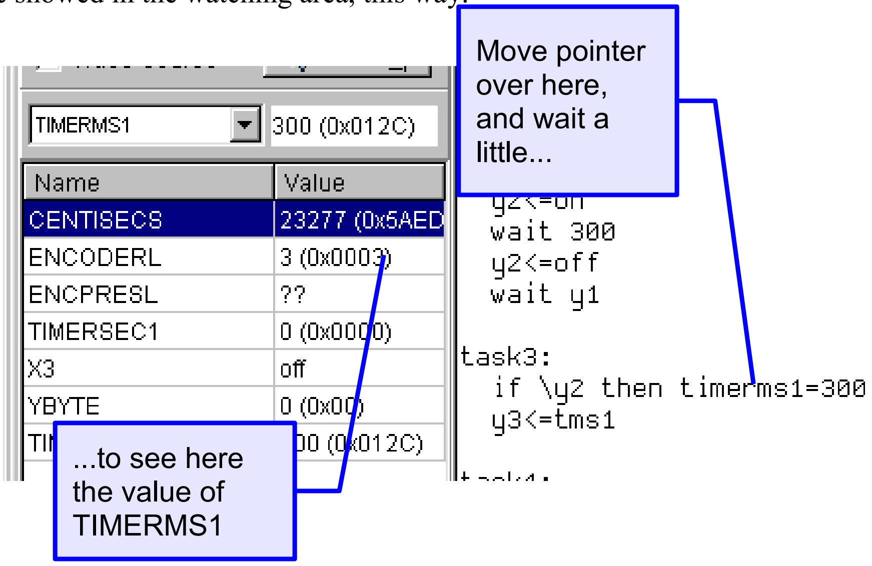
The writing area is where the program text is entered and corrected. Its behavior is the same of a normal text editor, with a few special functions added explained below.

Using the mouse

As usual, the mouse is mainly used to move the text cursor in different places of the text, by clicking. It is also possible to click and drag to select some text. Clicking with the right button makes a pop-up menu appear:

- Change font...
  Selects a font to use for text editing. The chosen font is remembered in future sessions.
- Show ASM window
  Shows a window containing machine code of the compiled program. Can be used to better verify the code generated by the compiler, and can be useful during debugging.
- Show log window
  Shows the lower window, used for messages, if this window has been hidden.
- Show hint popups
  Lets to turn off the hints (pop-up explanations), useful only in the first sessions. This setting is remembered in future sessions.

If the mouse pointer is moved over a resource name written in the text, after a little time the resource and its current value are showed in the watching area, this way:



TIMERMS1 300 (0x012C)

Name	Value
CENTISECS	23277 (0x5AED
ENCODERL	3 (0x0003)
ENCPRESL	??
TIMERSEC1	0 (0x0000)
X3	off
YBYTE	0 (0x00)
TI	00 (0x012C)

```
  y2<=on
  wait 300
  y2<=off
  wait y1

task3:
  if \y2 then timerms1=300
  y3<=tms1
```

Move pointer over here, and wait a little...

...to see here the value of TIMERMS1

Keyboard shortcuts (special function keys)

During text editing a few key combinations recall special functions listed in the table below:

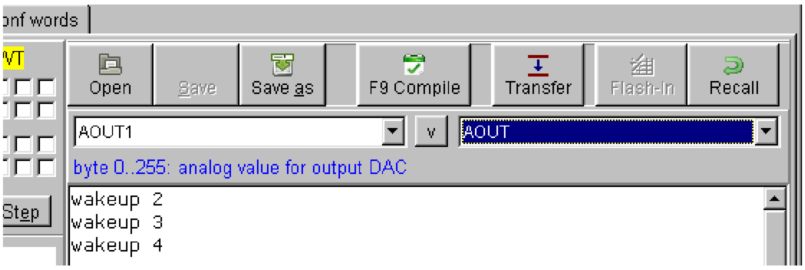
Key combination	Operation
F9	Compile the program
Ctrl+F9	Compile, send to EZ-Red, and run
F8	Makes a single step in the current task
Ctrl+F	Search a word in the program text
Ctrl+L, or F3	Search again the last word (next occurrence)
Ctrl+W, or F7	Watch (see the value of) the resource under the cursor
F1	Shows a short description of the resource under the text cursor, in the higher message area.
Ctrl+Alt+UP (up arrow)	Jumps backwards to the previous occurrence of the word under the cursor
Ctrl+Alt+DN (down arrow)	Jumps forward to the next occurrence of the word under the cursor

Program Compilation

When a file is freshly opened, or after the current file has been modified, it must be compiled before it can be transferred to the module. Of course, it is possible to compile without sending the file, just to check for errors.

To compile the program, click the button “F9 Compile” or hit the F9 key. If the compiler founds an error, a message is shown in the upper message area and the cursor moves to the location of the error. Otherwise, a message shows the compilation result (length of compiled program).

Helps to program writing



AOUT1

AOUT

byte 0..255: analog value for output DAC

```
wakeup 2
wakeup 3
wakeup 4
```

The two fields below the main buttons can be opened to see a list of all the identifiers (resources). The left list contains all the identifiers (X1, X2, X3...) while the right side one contains only the base name (resource category). Selecting an item from either ones, the other synchronize and a short message appears just below. This kind of message can be obtained also by pressing the F1 key while the text cursor is over an identifier name. The center button labeled “V” inserts the shown identifier in the program text.

Program transfer to EZ-Red

After a cycle is compiled, it is possible to transfer it to the module in order to execute it or to store it permanently. To begin the operation, click the button Transfer; it can take a few seconds.

If, when clicking Transfer, the program needs compilation, this step is automatically performed.

The transfer operation requires the current running cycle to stop. If the “Transfer button” is clicked with the right-side mouse button, the newly transferred program is started immediately.

Flash-In and Recall

The Flash-In button stores the PLC program in the module internal flash memory, so that at every power-on the cycle is automatically executed.

The Recall button recalls the saved cycle, if it has been overwritten by a program transfer operation. Of course, if after having transferred a program, that program has also been stored with “Flash-In”, the recall operation will recall the last program sent.

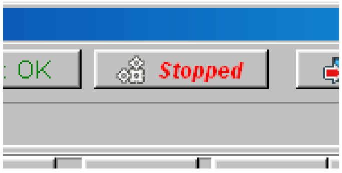
WARNING: flash memories can not be rewritten an infinite number of times; they can be written and rewritten many times, but it is good not to abuse of them.

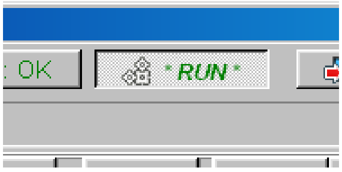
Program erasure (module flash erasure)

In order to delete a PLC cycle stored in flash memory, so it is not executed at module power-on, click with the right mouse button on “Recall”. A confirmation request is issued before proceeding.

Start and stop the cycle

The EZ-Red internal PLC runs independently from the program editing. However, it is often needed to start, stop or restart a cycle. The upper-right button, labeled “Stopped” or “*RUN*”, shows the running status and, if clicked, inverts it:





When the cycle starts to run from stopped state, all the internal parameters (flags, options, inputs thresholds...) are set to their default.

When a running cycle is stopped, all the outputs are left as they are. There is no way to restart from the stopped point: to do that, all the tasks must be suspended or put in step-mode – this is different than clicking the RUN/STOP button.

Debug

In order to discover program bugs, the behavior of the program and the resources (or variables) must be observed to note the point where an error reveals – where the result found is not the one expected.

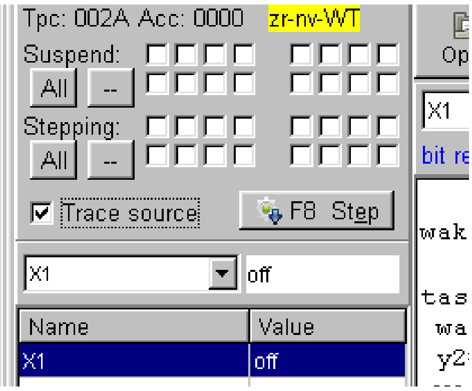
There are many techniques to this; a fairly common one is to execute slowly the program step-by-step to see the effects of every single operation. At every step inputs, outputs and other resources are checked to see if they reflect correctly the intended operations, and if the next, yet to be executed instruction, is correct.

Sadly the natural environment of EZ-Red can be difficult, because the module can execute different tasks concurrently, and because external devices can modify their behavior when not managed at the correct speed. The EZ-Red debugging system can suspend or set to stepping mode one or more tasks, in order to ease the debugging of complex cycles.

As a practical example, take the following program (connect a wire from Y1 to X1):

```
; debug example

wakeup 2

task1:
  wait x1
  y2=on
  wait 750
  y2=off

task2:
  y1<=off
  wait 500
  y1<=on
  wait 500
```

The cycle contains two tasks: task1 waits for input 1 (x1) to raise, then generates a pulse of ¾ of a seconds (750 milliseconds) on Y2 (digital output 2) and the restarts. Task2 blinks Y1 (digital output 1) with a frequency of 1 hertz. To verify that this is true, write the program and click with right-button of the mouse to compile, send to the modules and run. Make sure that the checkbox "Trace source" is ticked:



Tpc: 002A Acc: 0000 zr-nv-WT

Suspend: □□□□ □□□□ □□□□ □□□□

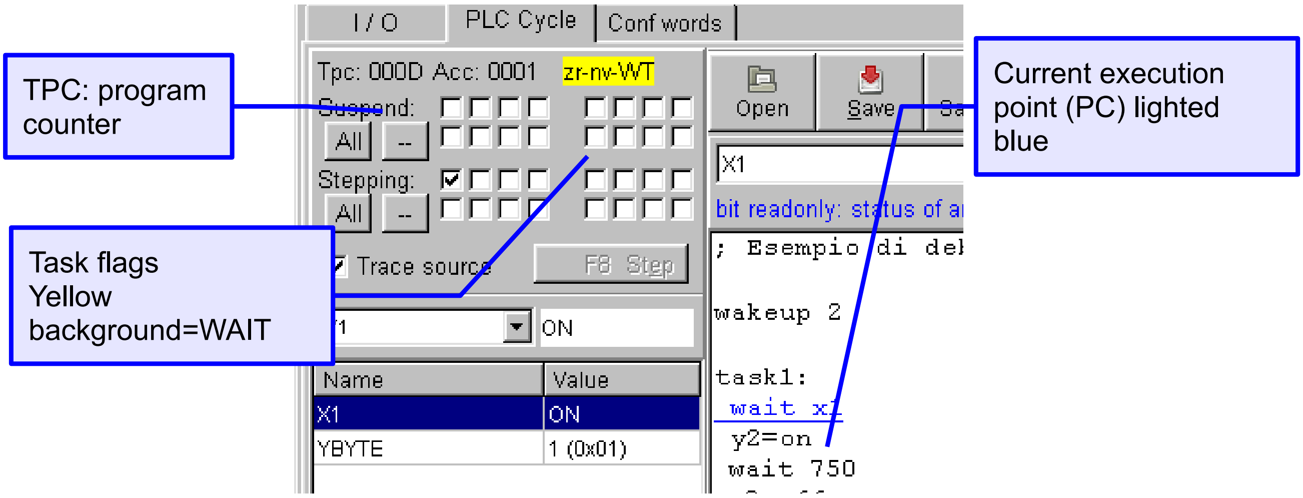
Stepping: □□□□ □□□□ □□□□ □□□□

☑ Trace source

X1 off

Name	Value
X1	off

The writing/editing area animates, while it shows the current execution point:



TPC: program counter

Task flags
Yellow
background=WAIT

Current execution point (PC) lighted blue

Tpc: 000D Acc: 0001 zr-nv-WT

X1

bit readonly: status of a

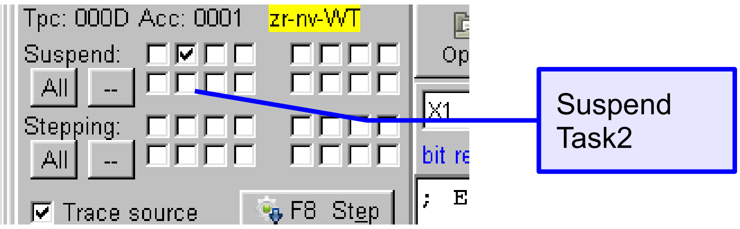
ON

Name	Value
X1	ON
YBYTE	1 (0x01)

```
; Esempio di del

wakeup 2

task1:
 wait x1
 y2=on
 wait 750
```

The execution point should stay most of the time on the “wait x1” instruction, but occasionally moves away.

As task1 waits for a front on X1 generated by task2, try to suspend task2 by clicking in the related checkbox:



Tpc: 000D Acc: 0001 zr-nv-WT

X1

Suspend Task2

Now task2 does no more generate pulses on X1, so task1 waits indefinitely. Clicking again on the suspend checkbox of task2 makes task1 cycle again. Similarly, it is possible to suspend task1, and see that task2 continues to run, but the output Y2 stays still.

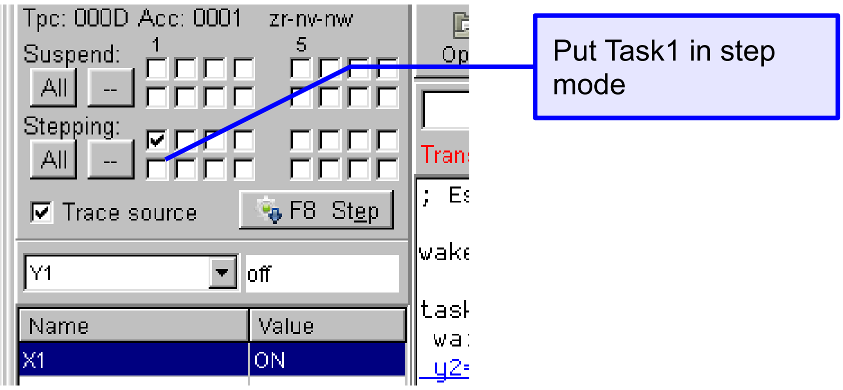
Task suspension

Using the checkboxes in the Suspend group it is possible to suspend one or more tasks, or reactivate them deleting the tick mark. The "All" button suspends all the tasks, while the "--" button enables all them together. To suspend a task is useful to avoid interferences by other parts during analysis of a fragment of code.

A task can be suspended or reactivated by TSmon, but other running tasks can do this as well, using instructions like SUSPEND, WAKEUP and RESTART.

Step mode for a task

When a task is in Step-Mode actually stops and waits for TSmon; TSmon can signal to execute a single step, the task executes a step and then stops again. Acting this way it is possible to analyze, slowly and carefully, a task. Now in the example program reactivate all the tasks e tick the STP box of the first task:

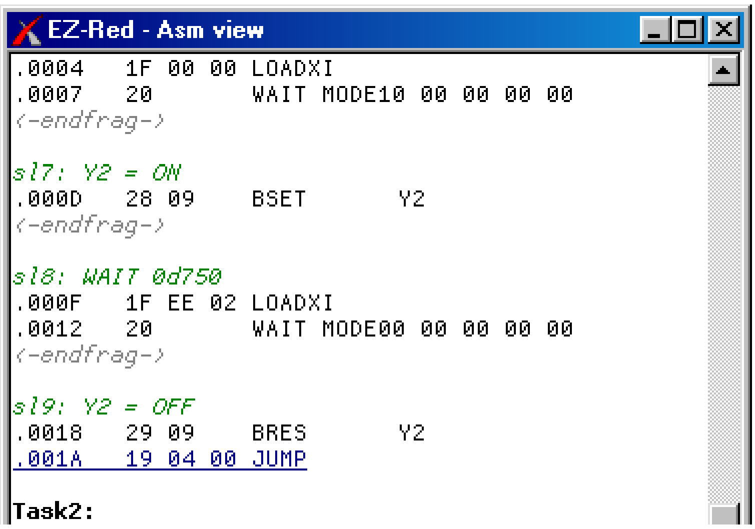


Tpc: 000D Acc: 0001 zr-nv-nw

Name	Value
X1	ON

Put Task1 in step mode

After having ticked the checkbox, the task Task1 stops. At this point press F8 (or click on "F8 Step") repeatedly to execute instructions one by one. After every key press a different line of source code (the program) will be underlined and highlighted in blue color. It can happen that no lines are highlighted; this means that EZ-Red is executing a hidden instruction, for example the implicit jump to the beginning of the task. Click the right mouse button in the editing area and select "Show ASM window":



```
.0004   1F 00 00 LOADXI
.0007   20       WAIT MODE10 00 00 00 00
<-endfrag->

sl7: Y2 = ON
.000D   28 09    BSET       Y2
<-endfrag->

sl8: WAIT 0d750
.000F   1F EE 02 LOADXI
.0012   20       WAIT MODE00 00 00 00 00
<-endfrag->

sl9: Y2 = OFF
.0018   29 09    BRES       Y2
.001A   19 04 00 JUMP

Task2:
```

The window shows the machine code generated for the program; the green lines like "sl9: y2=off" indicate the original text of the program, followed by the machine instructions, written in black, generated by the compiler.

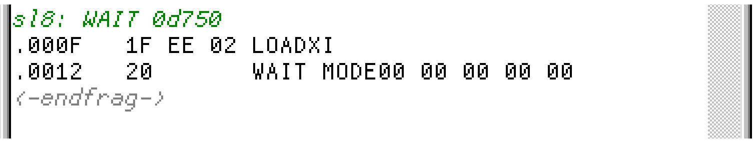
Here too the execution line is highlighted in blue, and shows that the PLC is waiting to execute a JUMP (goto): it is the implicit jump to restart the task. By pressing F8 the instruction will be executed. Please note that while Task1 is stopped waiting for the user to step it, Task2 is running normally and generates pulses.

When the EZ-Red is executing a WAIT instruction, the Step button is disabled for the duration of the WAIT and the flags written on a yellow background confirm that the task is waiting and can not prosecute.

When the WAIT terminates the execution point moves to the next instruction ("y2=on" in this case), and the task stops again. It can happen that a WAIT lasts forever, if the necessary conditions to terminate it do not happen: in this case the task must be manually unblocked, for example reactivating other tasks, or acting on the external circuitry, or manually modifying some internal resources.

Step mode on more than one task

If two or more tasks are put together in step-mode, the execution point keeps to jump from one task to another. This is correct, because it reflects the way the PLC executes concurrent tasks. There can also be the impression that the same instruction is executed more than once. In reality, such an impression is given when a single high-level statement is compiled to several low-level instructions, and the PLC switches task at the low-level instruction. This can be verified in the ASM window:

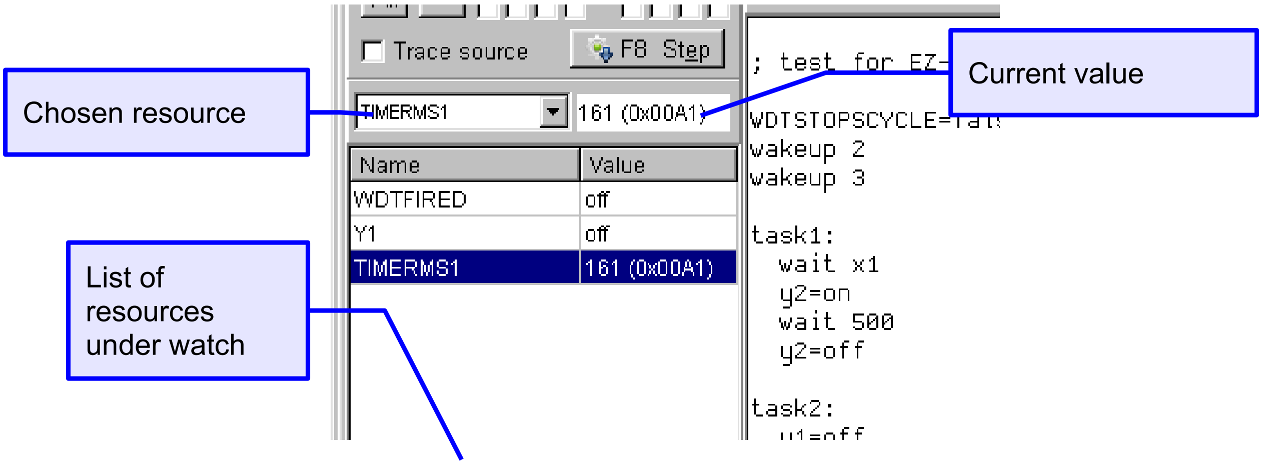


```
sl8: WAIT 0d750
.000F   1F EE 02 LOADXI
.0012   20       WAIT MODE00 00 00 00 00
<-endfrag->
```

The WAIT command is made, at low level, by two distinct instructions; the PLC executes one, then one from each other task, and then comes back to this task for the second instruction giving the false impression that the original WAIT were executed twice.

Watching and modifying resources

Just below the "F8 Step" button are located a few controls to show and modify EZ-Red resources:



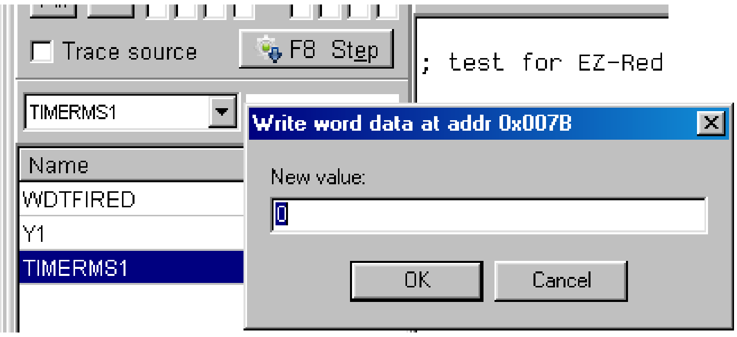
TIMERMS1 | 161 (0x00A1)

Name	Value
WDTFIRED	off
Y1	off
TIMERMS1	161 (0x00A1)

```
; test for EZ-
WDTSTOPSCYCLE=fal
wakeup 2
wakeup 3

task1:
  wait x1
  y2=on
  wait 500
  y2=off

task2:
  y1=off
```

The box "Chosen resource" contains, in alphabetical order, all the EZ-Red identifiers plus those declared in the program source text. By clicking on the triangle on the right, the box opens and, browsing the listed items, their current value is showed in the field "Current value".

Clicking on the "Current value" box with the right mouse button, the current resource is added to the listing below; this way it is possible to watch different resources together. A double click, instead, modifies the value of the resource. If it is a boolean (bit) value, the bit is inverted; otherwise a pop-up window shows up, asking for a new value to assign:



```
; test for EZ-Red
```

TIMERMS1

Name
WDTFIRED
Y1
TIMERMS1

New value: 0

It is possible to enter numbers in decimal (normal) format, or in hexadecimal format preceded by notation "0x" like "0x0a" (which is 10 in decimal).

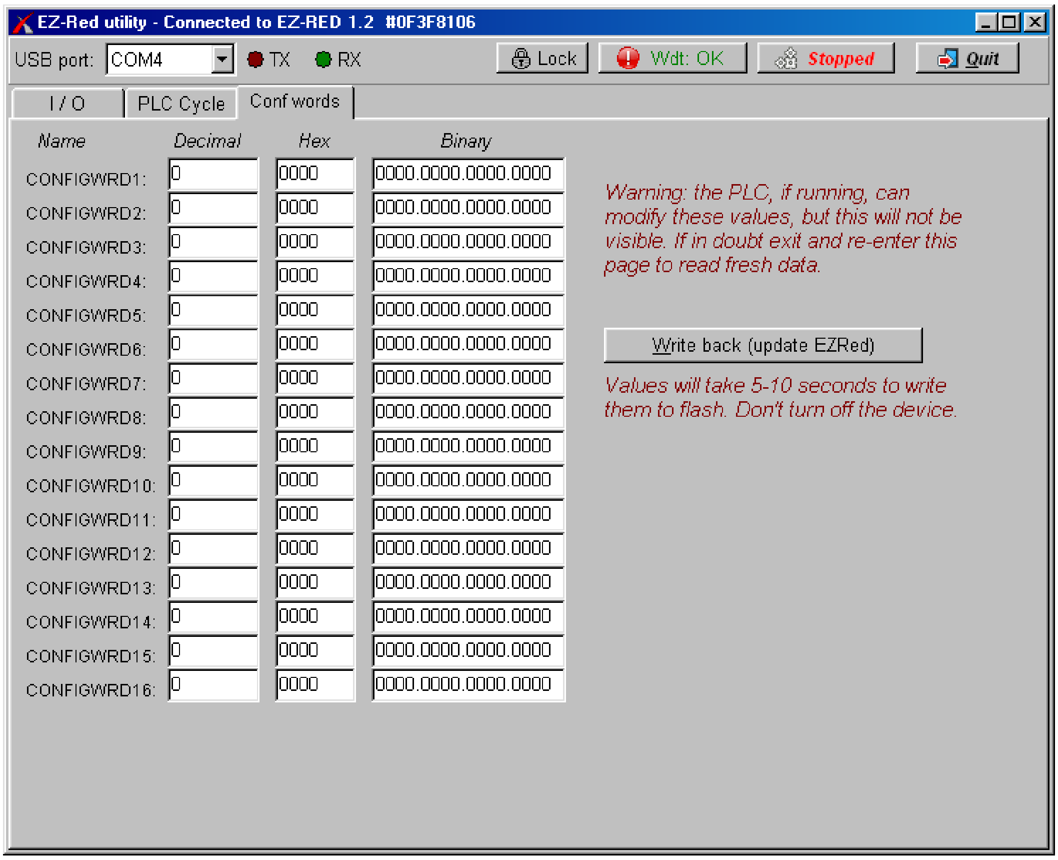
List of resource under watch

This table (list) shows a number of resources and their respective value, in real time. Clicking a single item in the table makes that item to be loaded in the boxes "Chosen resource" and "Current value", allowing to write a new value in it by double-clicking "Current value". A double-click in the table has the same effect.

When a row of the table is selected, it is possible to click the right mouse button to pop up a menu; the actions in this menu can also be invoked directly with the following shortcut keys:

Key combination	Operation invoked
Ctrl+Del	Delete the row
Ctrl+UP (up arrow)	Move the row up
Ctrl+PgUP (page up)	Move the row all to the top
Ctrl+Dn (down arrow)	Move the row down
Ctrl+PgDn (page down)	Move the row all to the down

"Conf words" tab (configuration variables)



EZ-Red utility - Connected to EZ-RED 1.2 #0F3F8106

USB port: COM4

Name	*Decimal*	*Hex*	*Binary*
CONFIGWRD1:	0	0000	0000.0000.0000.0000
CONFIGWRD2:	0	0000	0000.0000.0000.0000
CONFIGWRD3:	0	0000	0000.0000.0000.0000
CONFIGWRD4:	0	0000	0000.0000.0000.0000
CONFIGWRD5:	0	0000	0000.0000.0000.0000
CONFIGWRD6:	0	0000	0000.0000.0000.0000
CONFIGWRD7:	0	0000	0000.0000.0000.0000
CONFIGWRD8:	0	0000	0000.0000.0000.0000
CONFIGWRD9:	0	0000	0000.0000.0000.0000
CONFIGWRD10:	0	0000	0000.0000.0000.0000
CONFIGWRD11:	0	0000	0000.0000.0000.0000
CONFIGWRD12:	0	0000	0000.0000.0000.0000
CONFIGWRD13:	0	0000	0000.0000.0000.0000
CONFIGWRD14:	0	0000	0000.0000.0000.0000
CONFIGWRD15:	0	0000	0000.0000.0000.0000
CONFIGWRD16:	0	0000	0000.0000.0000.0000

Warning: the PLC, if running, can modify these values, but this will not be visible. If in doubt exit and re-enter this page to read fresh data.

Values will take 5-10 seconds to write them to flash. Don't turn off the device.

In this tab it is possible to review and modify the configuration variables - resources which maintain their value even at power-off, i.e. across sessions. These variables are accessible from the PLC cycle too, which can also modify them. These variables can be used to customize the same PLC cycle for different situations, perhaps supplying the customer with an ad-hoc written program to manage them, or they can be used as a mean to store session informations to be utilized in the future.

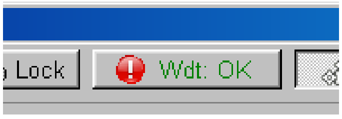
When the tab is recalled there is a short pause while TSmon fetches the variables from EZ-Red, then the values are showed. Every value is a 16-bit word, displayed in decimal, hexadecimal and binary notations. Data can be modified in any text box, as the corresponding ones are update consequently.

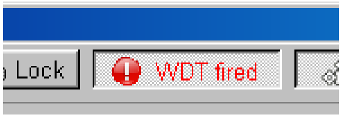
When the settings are done, the "Write back" button must be clicked to transfer data to the module.

Beware: if a PLC cycle is executing, and the cycle modifies these values, the updated data are not displayed - data are only loaded when the "Conf words" tab becomes visible.

Beware: after clicking the "Write back" button, a few seconds are required for the EZ-Red to write the new values in its permanent memory. If the module is turned off meanwhile, no data will be saved and the variables will retain the previous content.

Watch-dog



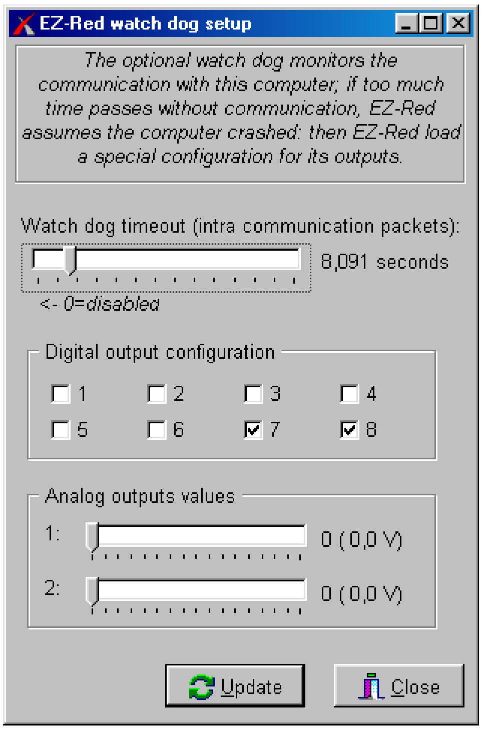


The "Wdt: OK" / "WDT fired" button shows the watch-dog status and allows to modify its settings.

When the label shows "OK", the watch-dog has not fired, as in the left image above. If the watch-dog has been triggered, the button is labeled "fired".

It is possible to click the button in three different ways:

1. When the button is released (Wdt: OK), if left-clicked it stops the PLC cycle and sets the "alarm" condition (watch-dog fired). Please note that TSmon performs the two operations separately in order to do a real "emergency stop", because the watch-dog could be programmed to not stop the cycle.
2. When the button is pressed (WDT fired), if left-clicked it resets the watch-dog, bringing back the normal status of the device. This does not mean that the cycle is restarted - simply the alarm condition is canceled.
3. When clicked with the right mouse button, it pops up a dialogue window to allow to view and modify the watch-dog settings:



The optional watch dog monitors the communication with this computer; if too much time passes without communication, EZ-Red assumes the computer crashed: then EZ-Red load a special configuration for its outputs.

Watch dog timeout (intra communication packets): 8,091 seconds

<- 0=disabled

Digital output configuration

☐ 1 ☐ 2 ☐ 3 ☐ 4
☐ 5 ☐ 6 ☑ 7 ☑ 8

Analog outputs values

1: 0 (0,0 V)

2: 0 (0,0 V)

In tis window it is possible to program a time-out for the commands sent by the computer. If the device does not receive data from the computer for more than the specified time, or if two consecutive messages from the PC are separated by more time than specified, the watch-dog triggers. When it triggers, it sets on the power outputs the pattern specified here under "Digital output configuration" and, on the analog outputs, the values written under "Analog outputs values". After having changed these settings, click Update to send the configuration to the EZ-Red. Click Close instead to leave the settings untouched.

watch-dog test

To test how the watch-dog works, stop the cycle if it is running, turn off all the outputs, and set the "Watch dog timeout" to a few seconds like 20 or 30.

Then, exit from TSmon (Quit button). After the specified time the watch-dog will fire: the red led near the outputs connector will flash, and the outputs will be loaded with the watch-dog settings.

Access protection

The device has two mechanisms to protect from unauthorized accesses: a password for the PLC cycle stored in flash, and a special bit DISABLEUSB which inhibits communication from the PC.

PLC flash memory protection password

The non-volatile memory (flash) storing the PLC cycle executed at power-on can be protected by a password consisting in a hexadecimal number from 0 to FFFF.FFFF (over 4 billions combinations). When the password is in effect, the PLC program can not be modified without knowing the password.

To set up the password, click the "Lock" button and write two times the hexadecimal password (the second time is for confirmation). When the password is set, the Lock button remains in pressed position. When the button is down, a click on it permits to unlock temporarily the device, by supplying the correct password (the one stored inside the device). The device remains unlocked only until a power-off.

To delete permanently the password, a new one must be set with value "0" (zero).

DISABLEUSB

This bit can be set by the PLC cycle, and inhibits the communication with the PC, apart from some commands like "device id" (model and version request), set/reset password (to unlock), and the emergency procedure (watch-dog + cycle stop).

The communication can be re-enabled if the PLC cycle resets the bit (may be with a special configuration applied on the inputs), or if the PC sends the appropriate command to unlock. In the latter case, if a password is stored, the PC must send the correct one and the device gets unlocked (temporarily); if no passwords is stored, the PC can send anything as password - the EZ-Red will be unlocked (temporarily).

The TSMon program can not work properly if DISABLEUSB is set: the connected device is recognized, but no other communications can take place. It is necessary to click the "Lock" button, CANCEL the insertion of a new password, then click again "Lock" and enter the device password (or any number if no password is set).

XON Electronics Srl
www.xonelectronics.it
info@xonelectronics.it

Internet product page is http://www.xonelectronics.it/prodotti/industriali/EZ-Red

Please report any error or imprecision to web@xonelectronics.it