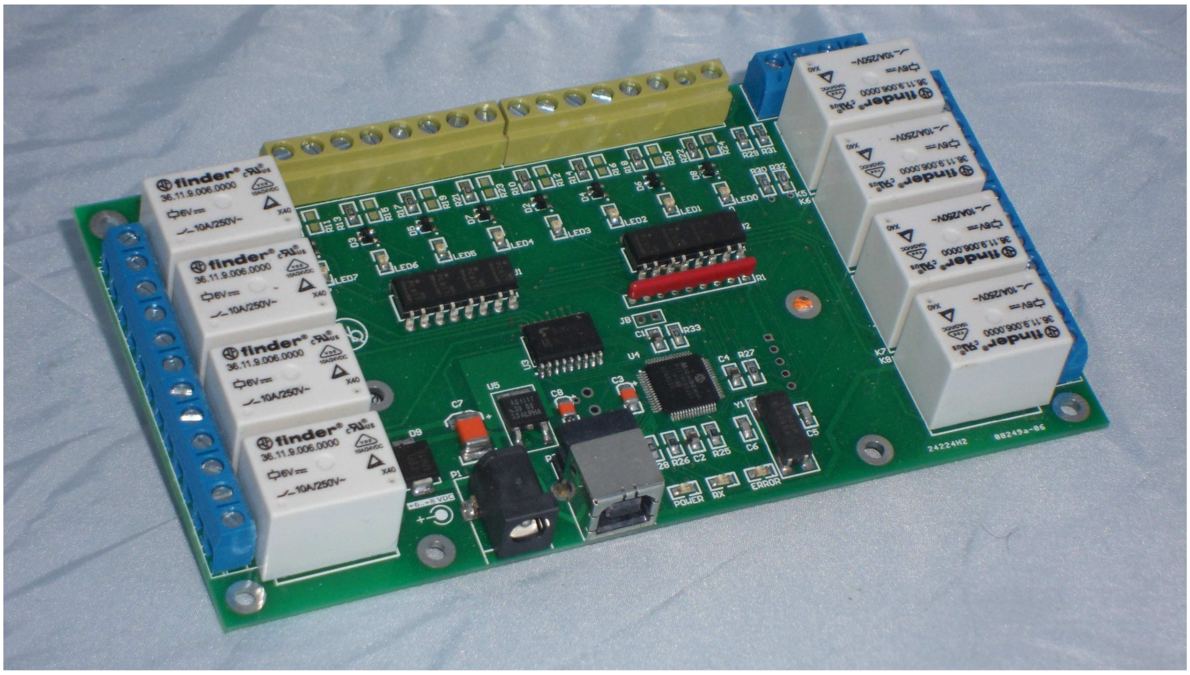
USB-IO

USB board to control up to

- 8 digital optically-isolated inputs
- 8 digital relay outputs
- 2 analog inputs

Software User Manual



ATTENTION: USB-IO board are delicate electronic devices, handle with care avoid electrostatic discharge. Hold the board by the edges to avoid touching the integrated circuits on it.

ATTENTION

handling precautions!

Version 1.0 - December 2013

USB-IO Concept

USB-IO interface adapter allows connecting various hardware devices to your PC. Easy installation and plug-and-play support makes USB-IO board an ideal solution to add USB interface to your HW.

The USB-IO has 8 SPDT relays outputs, 8 optically isolated inputs and 2 analog inputs. This device has many applications and we couldn't possibly list them all, but here are a few examples: The relay can be used to turn on and off a sprinkler system, automating a hot tub, or low-voltage lights. The inputs can be used to monitor magnetic window switches or garage door position. The analog inputs can be used to monitor some analog levels.

Helpful the provided demonstration software, connect the USB-IO to a USB host port and control IO’s is really simple. Or with the Helpful provided Class file for Visual Studio, develop your own application.

USB-IO Caracteristics

- Full HID USB2.0 compliant; driver natively include under Windows or Linux OS
- High-Speed USB 2.0 device, USB 1.1 compatible
- 8 Relay output’s 10A / 250VAC
- 8 optically isolated input’s 0/10VDC – User modifiable*
- 2 x 10 bits analog input’s 0/10VDC – User modifiable*
- 1 Kbytes of user Flash memory
- Demonstration software and DLL are available FREE OF CHARGES (Windows 2000 / XP / VISTA / Seven) – Visual Studio - Framework .NET 4.0 © MICROSOFT
- PCB dimensions : 150x83 mm - RoHS compliant
- Weight : 181 grams

* need solders tools to change SMD components

Support

Commercial department :
commercial@eko-fpga.com

WEB Site :
http://www.eko-fpga.com

Technical department :
technique@eko-fpga.com

The board is adaptable according to your specifications.
If the volume of your application requires it, do not hesitate to contact us.

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***technique@eko-fpga.com***. Please detail any errors you find and include your mailing address so that we can send you any manual updates.

TABLE OF CONTENTS

GLOSSARY

USB-IO	: USB IO controller board
$xx	: HEXADECIMAL notation
%xxxxxxxx	: BINARY notation

DOCUMENT REVISION HISTORY

Paragraphs	Rev.	Date	Modifications
	1.0	02.12.13	First edition of the document.

NOTA: all the pages carry the index corresponding to their edition even if their contents were not modified.

1 Preface

USB-IO differs from most of other low-cost adapter by the event-driven interface. It means that the USB-IO adapter can notify your PC software immediately after the input value has changed. The event-driven interface is very flexible. The conditions of an event generation can be configured for every USB-IO module independently. For example, ADC module can send events when the voltage on the analog input descends below predefined value or rises above any other else.

Would you like to analyze changes over time? No problem, USB-IO adapter can send events periodically with the predefined repeat interval. The polling interface is supported by the USB-IO adapter.

This is a fully assembled and tested unit. The board requires external power of 6-8VDC / 500mA.

The relays can control loads of up to 10 amps and 240VAC. Please see below for more detailed specifications.

USB-IO uses the Human Interface Device (HID) USB profile. It does not require USB drivers and so is immediately Plug-and-Play with present and future Windows©, Linux© and Mac© operating systems.

Many links are available on the Web. You can start by:

- http://www.lvr.com/hidpage.htm
- http://labs.mike-obrien.net/Document.aspx?id=hidlibrary

Control software and class file, writing in VB.NET language (Framework .NET 4.0), is free of charge:

You can download the demonstration software on http://www.eko-fgpa.com

2 Software installation

These paragraphs are intended to detail the software installation steps as well as describe what is being installed.
The software provided with this board need to be downloading and ***must be installing onto your hard disk prior to use.*** To do this, perform the following steps as appropriate for your operating system.

Win2000/XP/Vista/Seven

a. download the install file from http://www.eko-fpga.com
b. run the installation
c. Follow the on-screen prompts to install the software for this board.

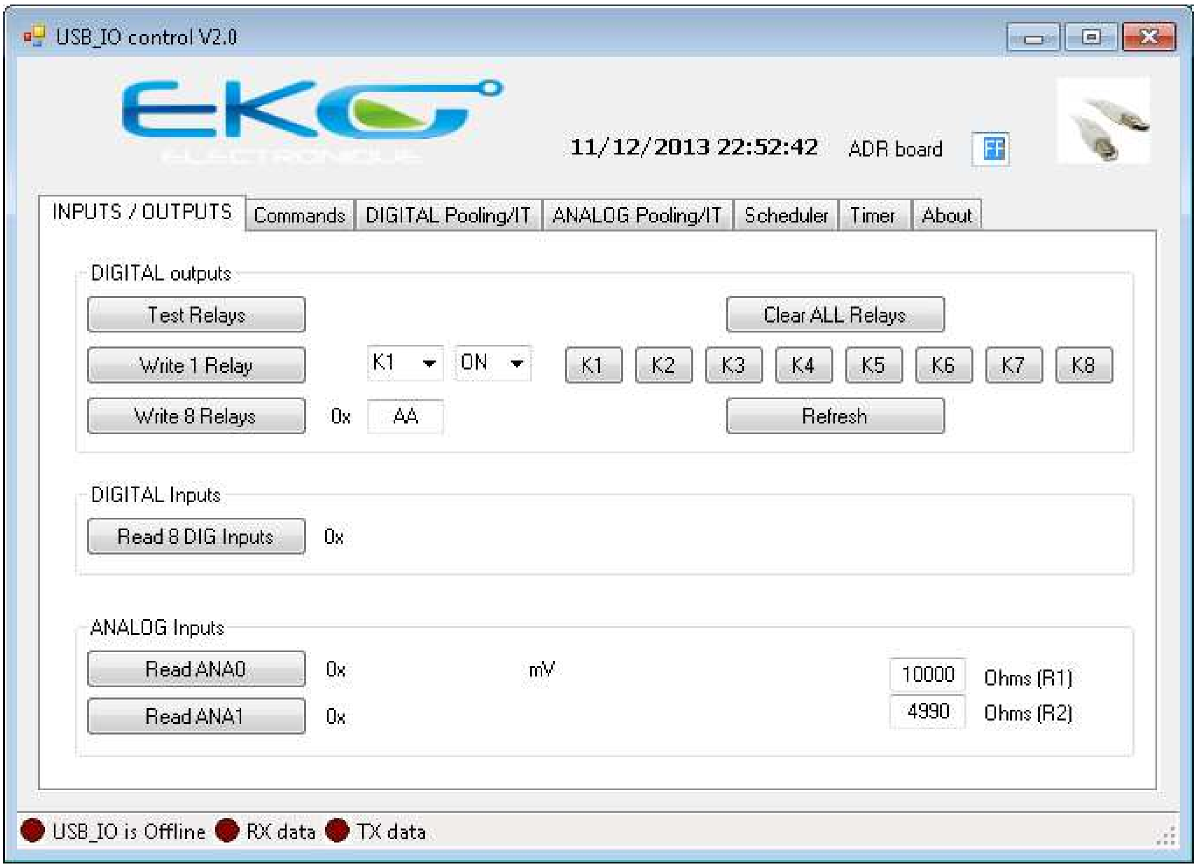
If you don't have the Microsoft .NET4.0 Framework already installed, the installation software will try to download it.

The demonstration software, sources project (VB.NET Framework 4.0) and DLL are provided as free of charges.

3 Software description

This software is provided as simple example to control the inputs / outputs of the board. A DLL is provided to help developers to design their own specific control software.

3.1 INPUTS/OUTPUTS tab



11/12/2013 22:52:42

ADR board FF

K1 ON

0x AA

0x

0x

0x

10000 Ohms (R1)

4990 Ohms (R2)

● USB_IO is Offline ● RX data ● TX data

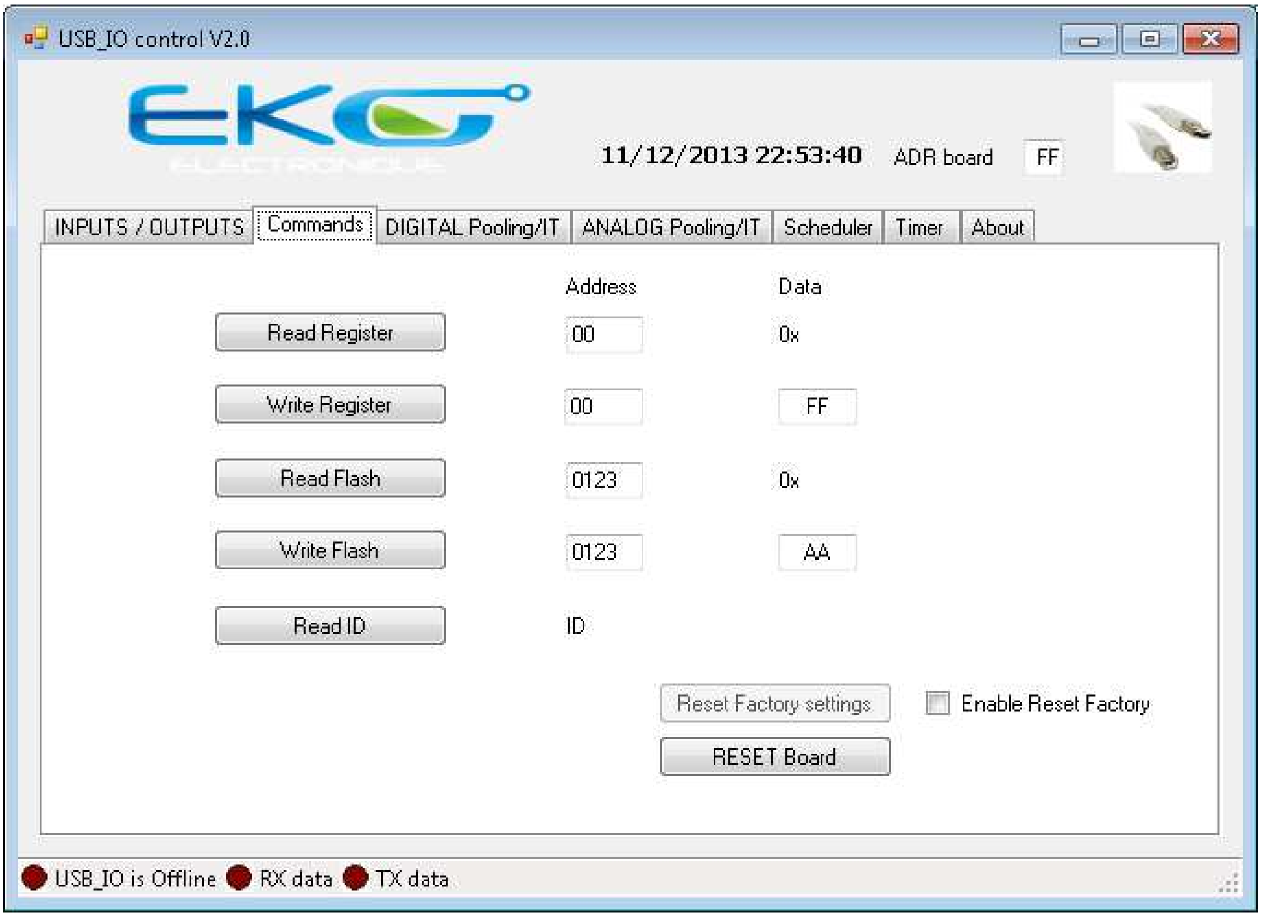
From this tab, you can read and write all inputs / outputs.

The DIGITAL outputs area demonstrate access to unit output or to all outputs with only one command.

The DIGITAL inputs area is an HEXA value, bit coded. 0x11 means input 0 and 4 : ON

The ANALOG inputs area is 10 bits HEXA value. If you set up correctly the R1/R2 resistors divider, then the ANA0 value is converted in milli-Volts value too.

3.2 Commands tab



	Address	Data
Read Register	00	0x
Write Register	00	FF
Read Flash	0123	0x
Write Flash	0123	AA
Read ID	ID	

□ Enable Reset Factory

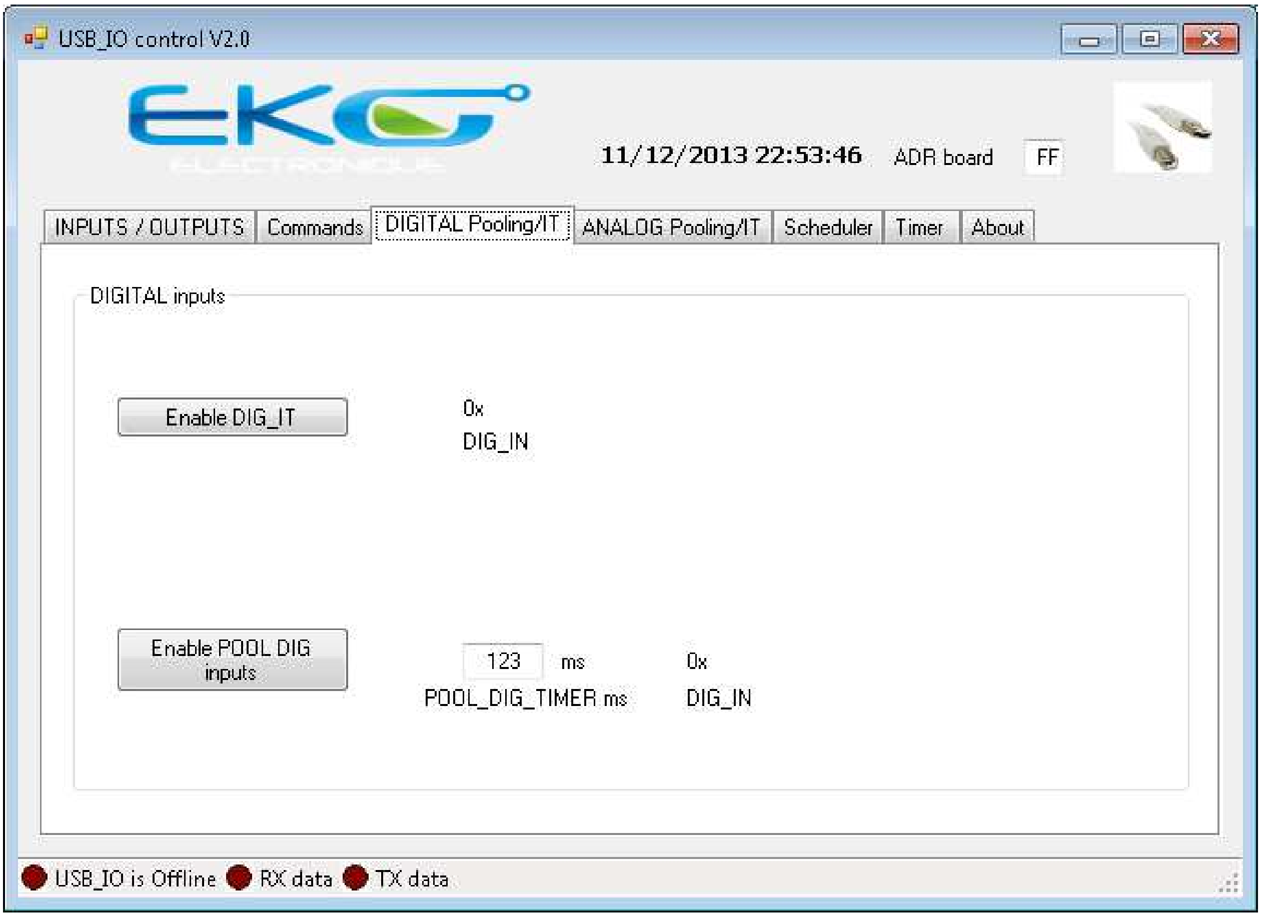
From this tab, you can read and write all registers and FLASH user area.

Have a look to §4 for the mapping registers array.

All values are HEX encoded.

If you check the "Enable Reset Factory" then you click on "Reset Factory settings" button, all settings will be overridden by the original factory configuration.

3.3 DIGITAL Pooling/IT tab

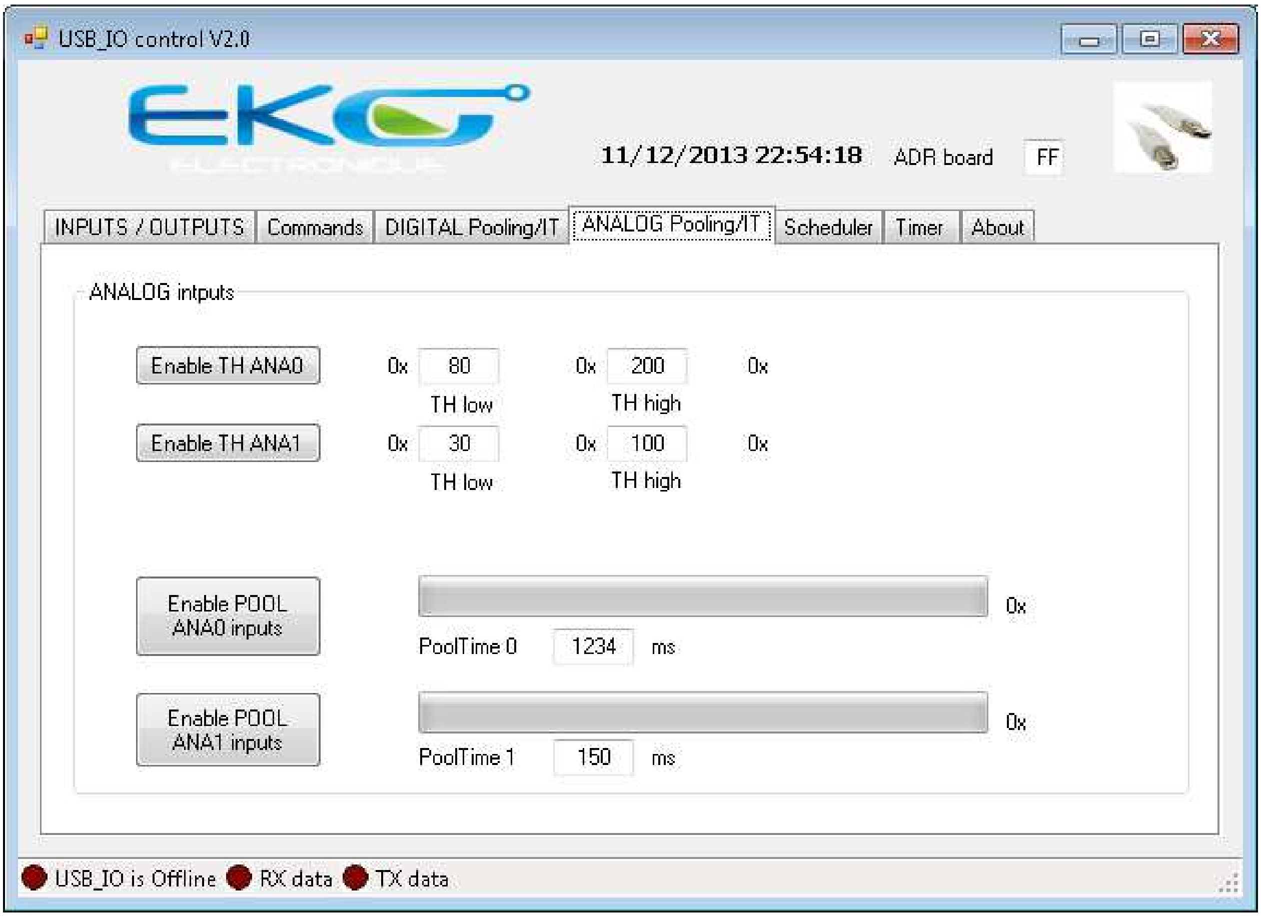


From this tab, you can play with the DIGITAL interrupts features of the board.
Two interrupts modes are available :

- edge interrupts : on any rising or falling edge, on any digital input, interrupt is fired then a read is executed

- pooling timer : on timer tick, interrupt is fired then a read is executed

3.4 ANALOG Pooling/IT tab



0x 80 TH low, 0x 200 TH high, 0x

0x 30 TH low, 0x 100 TH high, 0x

PoolTime 0: 1234 ms, 0x

PoolTime 1: 150 ms, 0x

From this tab, you can play with the ANALOG interrupts features of the board.
Two interrupts modes are available :

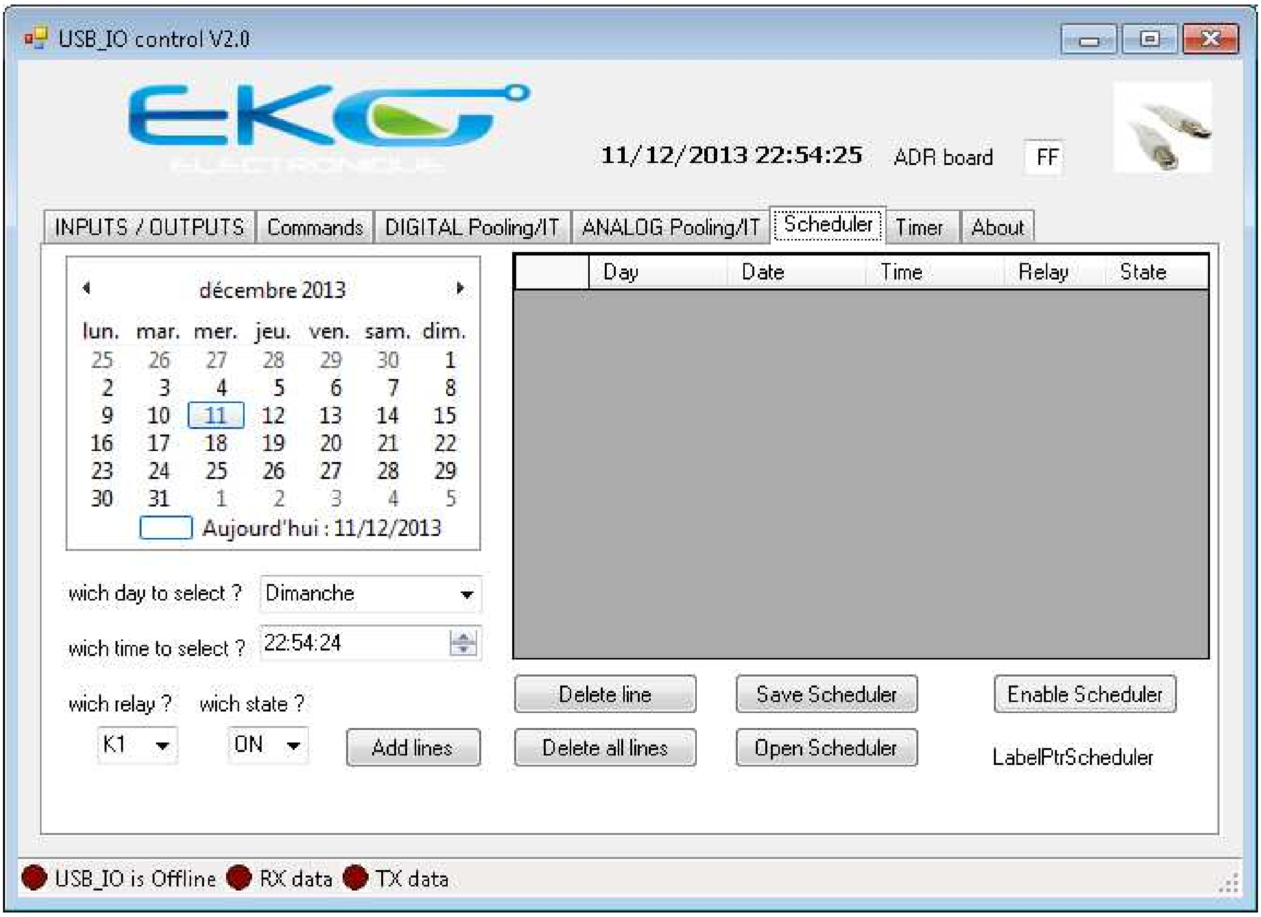
- threshold interrupts : set a low and up threshold, armed the board by enabling the threshold, then an interrupt is fired when a threshold is done.
- pooling timer : on timer tick, interrupt is fired then a read is executed

Each analog input has his own Threshold and Timer features.

Warning : Read carefully the USB_IO programming manual and the analog inputs characteristics.

DO NOT REVERSE POLARITY - DO NOT EXCED 3.6V on ANx cpu inputs.

3.5 Scheduler tab



11/12/2013 22:54:25 ADR board FF

décembre 2013

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Aujourd'hui : 11/12/2013

wich day to select ? Dimanche

wich time to select ? 22:54:24

wich relay ? K1

wich state ? ON

	Day	Date	Time	Relay	State

From this tab, you can play with the Scheduler features of the PC software.

4 Registers

4.1 Mapping Register

@	*Name*	*Access*	*Factory settings*	
0x00	ADDRESS	R/W	0xFF	Address of the USB-IO controller - FUTUR USE
	reserved			
0x10	CONFIG_DIG	R/W	0x00	Configuration bits for the DIGITALS IO's
	reserved			
0x20	OUTPUTS_REL	R/W	0x00	States of the 8 relay outputs
0x22	POOL_DIG_TIMERH	R/W	0x00	Set the pooling time of digitals inputs MSB byte
0x23	POOL_DIG_TIMERM	R/W	0x00	Set the pooling time of digitals inputs Middle byte
0x24	POOL_DIG_TIMERL	R/W	0x64	Set the pooling time of digitals inputs LSB byte (100 milliseconds)
	reserved			
0x30	CONFIG_ANA	R/W	0x00	Configuration bits for the ANALOGS inputs
0x31	THH_ANA0	R/W	0x02	Threshold High for the analog0 input – MSB
0x32	THL_ANA0	R/W	0x00	Threshold Low for the analog0 input – LSB
0x33	TLH_ANA0	R/W	0x00	Threshold Low for the analog0 input – MSB
0x34	TLL_ANA0	R/W	0x80	Threshold Low for the analog0 input – LSB
0x35	THH_ANA1	R/W	0x01	Threshold High for the analog1 input – MSB
0x36	THL_ANA1	R/W	0x00	Threshold High for the analog1 input – LSB
0x37	TLH_ANA1	R/W	0x00	Threshold Low for the analog1 input – MSB
0x38	TLL_ANA1	R/W	0x30	Threshold Low for the analog1 input – LSB
0x3C	POOL_ANA0_TIMERH	R/W	0x00	Set the pooling time of analog inputs MSB byte
0x3D	POOL_ANA0_TIMERM	R/W	0x03	Set the pooling time of analog inputs Middle byte
0x3E	POOL_ANA0_TIMERL	R/W	0xE8	Set the pooling time of analog inputs LSB byte (1000 milliseconds)
0x3F	POOL_ANA1_TIMERH	R/W	0x00	Set the pooling time of analog inputs MSB byte
0x40	POOL_ANA1_TIMERM	R/W	0x03	Set the pooling time of analog inputs Middle byte
0x41	POOL_ANA1_TIMERL	R/W	0xE8	Set the pooling time of analog inputs LSB byte (1000 milliseconds)
	reserved			
0xFF	*reserved*			

For detailed registers and commands descriptions, please have a look to the "USB-IO Manual Programming" document.

END OF DOCUMENT