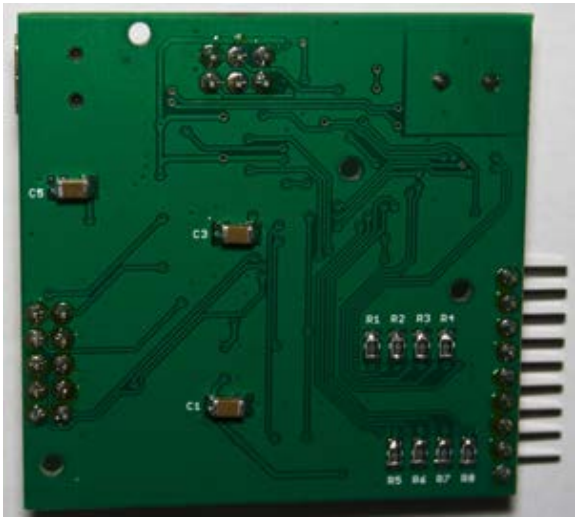
insideGadgets

Logic Observer v1.0



An 8 input CPLD based Logic Analyser with a selectable sample rate up to 50 MSPS with up to 131Kbit sample depth per input, controllable via USB with a GUI or command line interface and data format readable by Sigrok PulseView.

Features

- 2V to 5.5V input voltage for logic high reading
- 4.7M input impedance
- Sample rate of 50/10/5/2.5/1 MSPS and 500/250/100 KSPS
- Sample depth of 8K, 16K, 32K, 64K, 96K or 131K
- Selectable trigger pins
- High/low voltage triggering
- Trigger delay of 0 to 254ms
- Trigger count of 0 to 254 times
- Ability to re-program Altera CPLD / Atmel MCU
- Small 5x5cm form factor

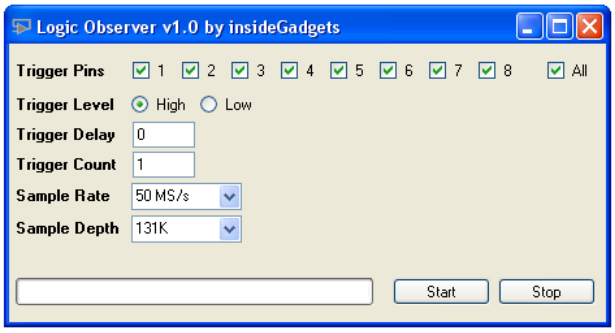
Specifications

PCB Board: 50mm x 50mm
Minimum input voltage: -0.5V
Maximum input voltage: 5.5V
Weight: 15 grams
CPLD: Altera 64 macro cell EPM3064ATC44
MCU: Atmel ATmega48/88/168/328 QFP
SRAM: Cypress 1Mbit SRAM CY7C1019DV33

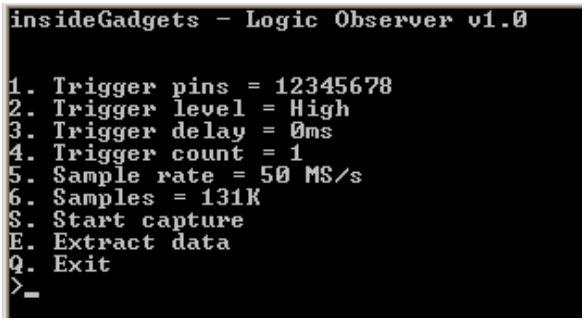
How to Use

You can either use the GUI or command line program.
1. Connect the USB cable to the board and install the driver found in the “Driver” folder.

GUI

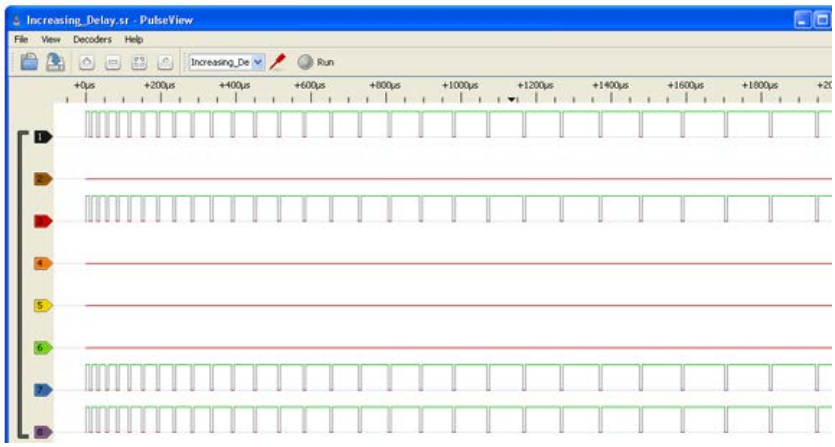


Command line



2. Run the Logic_Observer_v1.0_GUI.exe program for the GUI or the Logic_Observer_v1.0.exe for the command line program found in the "Logic_Observer_v1.0_ITF" folder.

3. Choose the trigger options and sample rate/depth that suits and press the Start button. The LED on the board will light and once a trigger matches your conditions the LED will turn off and the transfer will start automatically (the LED will be blinking). After 20 seconds, the transfer will be completed.



4. Load Sigrok PulseView and load the newly created .sr file.

Definition of settings

Trigger pins – Select which pins that we trigger off
Trigger level – Choose to trigger on a high or low voltage
Trigger delay – After we trigger, how long to wait until we start capturing
Trigger count – How many triggers we need to wait for until we start capturing
Sample rate – How many times per second to capture data
Sample depth – How long do we capture data for

How it works

The 8 inputs are fed into a buffer with 4.7M pull down resistors and the buffers outputs go to the SRAM and ATmega. The ATmega runs from a 20MHz crystal and controls the CPLD and buffer whilst communicating via USB using V-USB (http://www.obdev.at/vusb/). An analog switch allows us to select the input as the 50MHz oscillator or the ATmega's pin which we've configured as an output for the other clocks we generate from a timer. The 74HC144 buffer is used to convert 5V output logic of your programmer to 3.3V when reprogramming the ATmega.

Once we're waiting for a trigger, it's the ATmega's job to perform all the software selectable trigger features and the CPLD continues to increment the address lines of the SRAM until the ATmega is triggered. The CPLD then stores the start address minus 50, because the ATmega takes a few cycles to see the trigger event. When the CPLD's current address matches the incrementing address, it stops. The ATmega waits long enough for the capture to happen, returns to read mode and clocks out the data at a slower speed by setting the CPLD to read mode and toggling a next address pin to increment the address lines.

Programming the ATmega48/88/168/328 (optional)

This step is only necessary if you wish to update the firmware on the ATmega or if you have replaced the chip. You will need to solder in the 2x3 male ICP header (J1) if it's not populated already, you will require a programmer such as the USBtinyISP and your programmer will need to be supported by the AVRDUDE software - http://savannah.nongnu.org/projects/avrdude/

Firstly change your fuse bits so the ATmega uses a 20MHz crystal:
avrdude -p atmega48 -c usbtiny -U lfuse:w:0xde:m -U hfuse:w:0xd9:m -U efuse:w:0xff:m

Upload the \Logic_Observer_xxx_ATmega\main.hex file to the ATmega by running the following command: avrdude –p atmega48 –c usbtiny –U flash:w:main.hex

Programming the Altera CPLD (optional)

This step is only necessary if you wish to update the firmware on the CPLD or if you have replaced the chip. You will need to solder in the 2x6 male ICP header (J2) if it's not populated already, you will require a programmer such as the Mini USB Blaster and your programmer will need to be supported by the Quartus II software - http://savannah.nongnu.org/projects/avrdude/

You can open the \Logic_Observer_xxx_Quartus\ folder and open the .qpf project file.

Revision History

Rev. 2 – 26/04/2015

- Added pin state change detection in pin change interrupt so it only counts as a trigger if the pin has changed state.
- Added initial detection of pin state so we don't have to wait until the pin change interrupt is triggered to start capturing
- Moved clock stabilise delay to after starting a capture and waiting for a trigger as occasionally if a trigger occurred straight away, the first few readings would be random data

Rev. 1 – 28/02/2015

- Initial Revision

(c) 2015-2015 by insideGadgets
http://www.insidegadgets.com

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.
http://creativecommons.org/licenses/by-nc/3.0/