In-System Sources and Probes Editor in Quartus Prime.

Summary: *Procedure for the use of In-System Sources and Probes Editor*

Table of Contents

1 Description of the tool

In-System Sources and Probes Editor expands the access to verify a design by easily check the intern signals anytime and provides Quartus With a total dynamic debug environment. In cooperation with SignalProbe and SignalTap II logic analyzer, you get one powerful debug environment where you can-Generate stimuli and receive feedback directly from the logic. To run the virtual input into the design, you can use Virtual JTAG IP core or In-System Memory Content Editor.

The In-System Sources and Probes Editor consists of ALTSOURCE_PROBE IP core and an interface to control the instances of this according to the operation. Each core provides in and outputs, where you through the ports can run Selected Signals and through out-gates read selected signals. ALTSOURCE_PROBE IP core establishes a register chain to the in or out of the gates in the design. JTAG is used to shift data between ALTSOURCE_PROBE IP Core instances. ALTSOURCE_PROBE IP kernel handles communication between JTAG and registers in the design to provide a simple building block for Simulation and probing of their own design. In-System Sources and Probes Editor provides one-cycel read and write to logical nodes. Troubleshooting the low level by changing the inputs. This thanks to access to the logical nodes. Along with the SignalTap II Logic Analyzer you can force the device until the trigger conditions reach and trough this way isolate a fault.

You can easily with In-System Sources and Probes Editor add control signals in the design to virtual stimuli and this can simplify the following:

1. Create virtual buttons
2. Create a virtual front to communicate within your design
3. Emulate External data sensor
4. Monitor and change the run-time time constants on the fly

Tcl commands are supported by In-System Sources and Probes Editor as an interface between ALTSOURCE_PROBE IP core instances to increase the degree of automation.

Hardware and software requirements

1. Quartus Prime or Quartus Prime Lite Edition with TalkBack activated.
2. Download cable.
3. Altera development kit or similar with a JTAG connection.

Supported device families

1. Arria
2. Stratix
3. Cyclone
4. MAX

Signals that you want to look at are connected to an instance of In-System Sources and Probes IP core. After compilation, you can check each instance in the In-System Sources and Probes Editor view or the Tcl interface. Before the use of the In-System Sources and Probes Editor one must instantiate In-System Sources and Probes IP core, preferably over IP directory.

In-System Sources and Probes Editor does't support simulation so it must be removed before creating a netlist.

In-System Sources and Probes Editor provides complete control over ALTSOURCE_PROBE IP cores in the structure. The editor allows you to view all instances controlable in real time.

The In-System Sources and Probes Editor has three views:

1. The JTAG chain Configuration - Programming view. Here you can select the hardware and the device to be programmed.
2. Instance Manager - Information on the generated instances, as well as control of data required for the In-System Sources and Probes Editor.

   This view has some buttons/sub views:

- Read Probe Data
- Continuously Read Probe Data
- Stop Continuously Reading Probe Data
- Write Source Data
- Probe Read Interval
- Event Log
- Write Source Data

3. In-System Sources and Probes Editor - Logs all the data read from the selected instance. Allows modification of source data that is written to the device.

2 Use the following steps when using the tool:

Programming the Unit

1. In Quartus : open Tools > IP Catalog.
2. Select In-System Sources and Probes IP core.
3. Choose a name for the IP core.
4. Generate the IP core. Then use the generated core, which is based on your specifications, to instantiate the In-System Sources and Probes IP core.
5. Compile the project.
6. Open the In-System Sources and Probes Editor.
7. Select the correct hardware in the JTAG Chain Configuration, and the right device.
8. Look for SRAM object file (.sof), which contains the instance or In-System Sources and Probes instance.
9. Program the device by clicking on Program Device.

3 Demonstration

Here is a short demonstration of a very simple system with only 4 buttons and 4 LEDs. Start with a new project. Download In-System Sources and Probes IP core in the IP catalog according to figure 1.

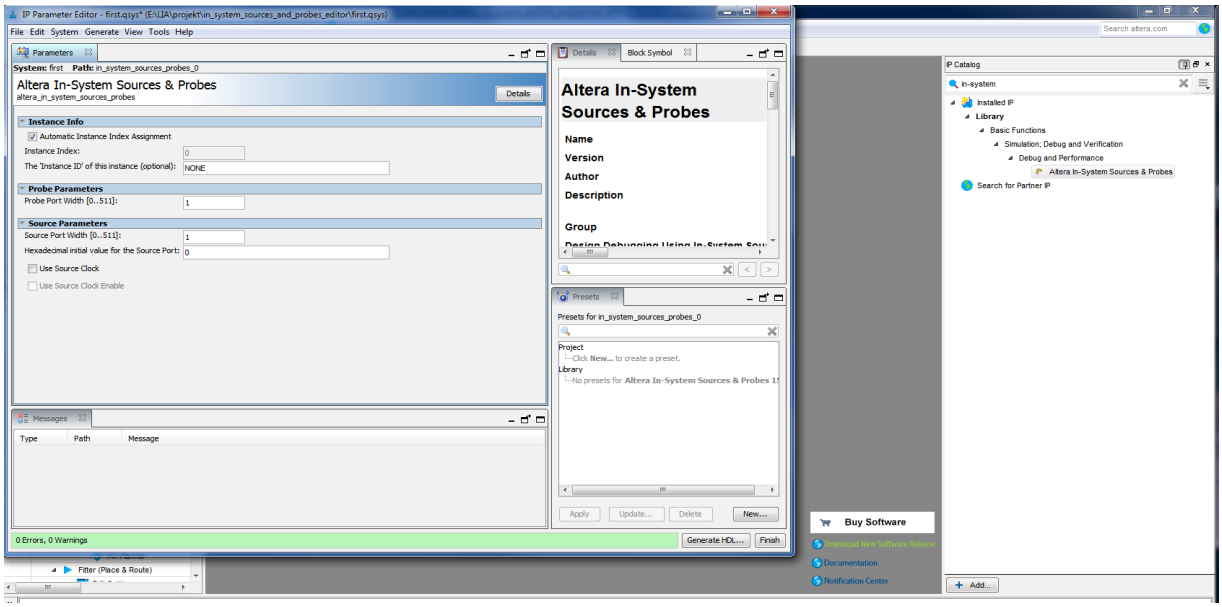


Figure 1: The figure shows a screenshot where I -System Sources and probes IP Core IP catalogue is included in the project.

Then the files must be included to the project. Usually a reminder appears as in figure 2.

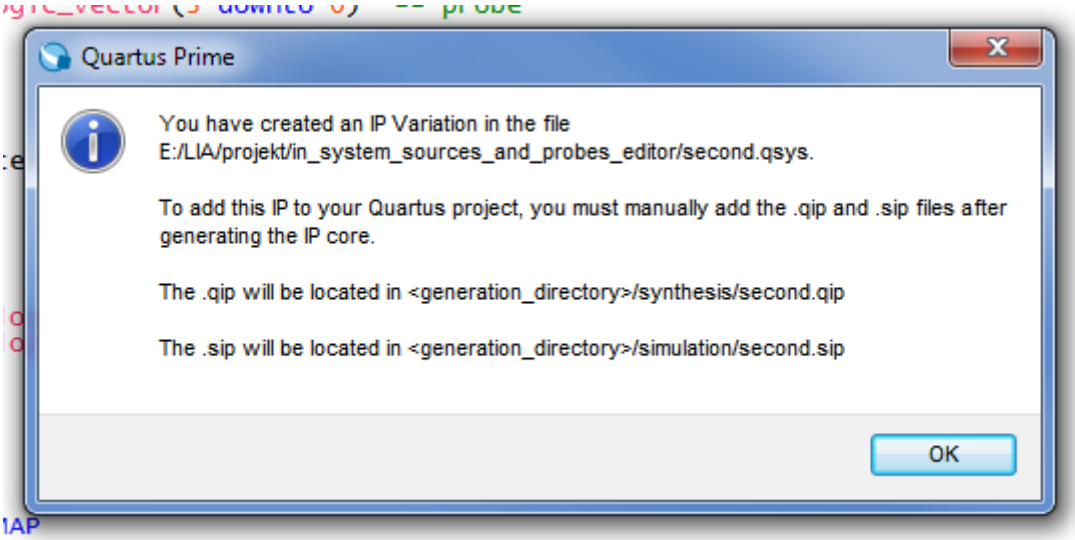
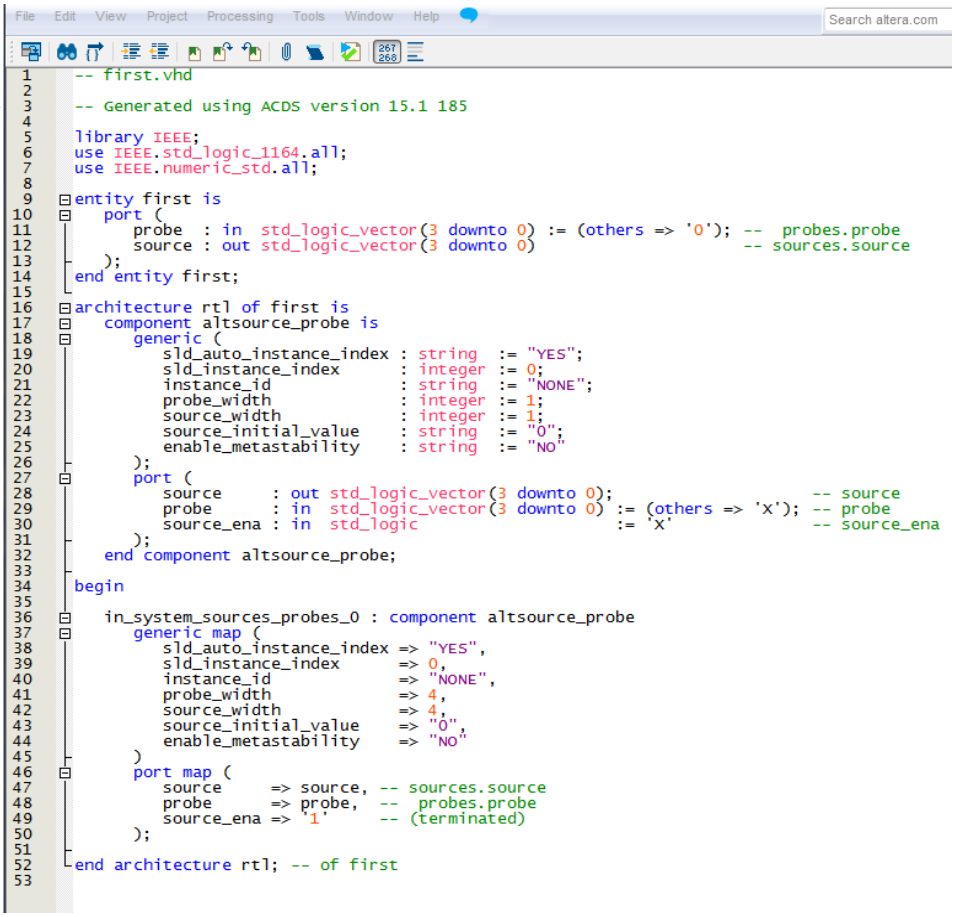


Figure 2: Reminder.

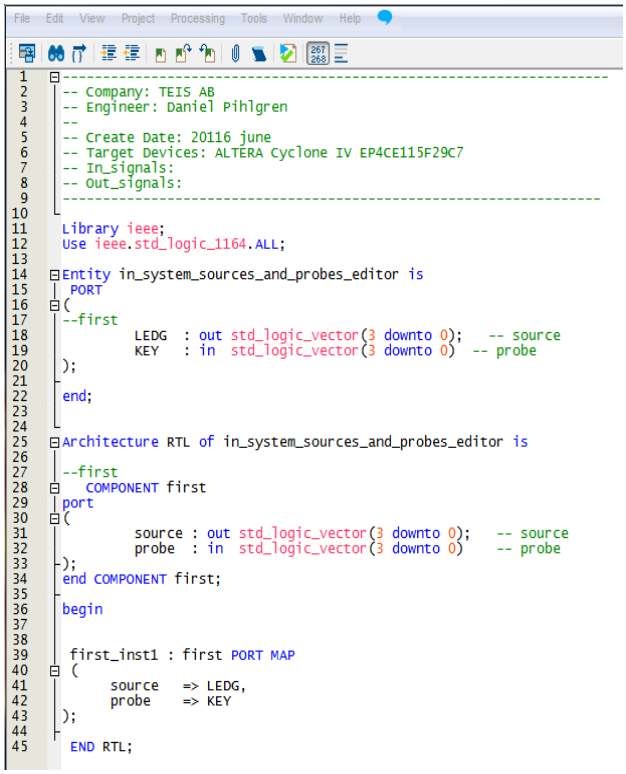
When the core is generated, the generated file must be included with the project trough the Settings menu. Then instantiated and this can be seen in Figure 3.



```
-- first.vhd

-- Generated using ACDS version 15.1 185

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity first is
    port (
        probe  : in  std_logic_vector(3 downto 0) := (others => '0'); --  probes.probe
        source : out std_logic_vector(3 downto 0)                     -- sources.source
    );
end entity first;

architecture rtl of first is
    component altsource_probe is
        generic (
            sld_auto_instance_index : string  := "YES";
            sld_instance_index      : integer := 0;
            instance_id             : string  := "NONE";
            probe_width             : integer := 1;
            source_width            : integer := 1;
            source_initial_value    : string  := "0";
            enable_metastability    : string  := "NO"
        );
        port (
            source     : out std_logic_vector(3 downto 0);                    -- source
            probe      : in  std_logic_vector(3 downto 0) := (others => 'X'); -- probe
            source_ena : in  std_logic                    := 'X'              -- source_ena
        );
    end component altsource_probe;

begin

    in_system_sources_probes_0 : component altsource_probe
        generic map (
            sld_auto_instance_index => "YES",
            sld_instance_index      => 0,
            instance_id             => "NONE",
            probe_width             => 4,
            source_width            => 4,
            source_initial_value    => "0",
            enable_metastability    => "NO"
        )
        port map (
            source     => source, -- sources.source
            probe      => probe,  --  probes.probe
            source_ena => '1'     -- (terminated)
        );

end architecture rtl; -- of first
```

Figur 3: Generated file.

Thereafter instantiated in the top_file as in figure 4.



```
----------------------------------------------------------------------
-- Company: TEIS AB
-- Engineer: Daniel Pihlgren
--
-- Create Date: 20116 june
-- Target Devices: ALTERA Cyclone IV EP4CE115F29C7
-- In_signals:
-- Out_signals:
----------------------------------------------------------------------

 Library ieee;
 Use ieee.std_logic_1164.ALL;

Entity in_system_sources_and_probes_editor is
 PORT
(
 --first
          LEDG  : out std_logic_vector(3 downto 0);   -- source
          KEY   : in  std_logic_vector(3 downto 0)  -- probe
 );

 end;


Architecture RTL of in_system_sources_and_probes_editor is

 --first
   COMPONENT first
 port
(
          source : out std_logic_vector(3 downto 0);   -- source
          probe  : in  std_logic_vector(3 downto 0)    -- probe
);
 end COMPONENT first;

 begin


 first_inst1 : first PORT MAP
 (
       source   => LEDG,
       probe    => KEY
 );

  END RTL;
```

Figur 4: Top_level file.

Make Pin Assignment and compile. Once the compilation is complete, open the In-System Sorces and Probes Editor from the Tools menu, as in Figure 5

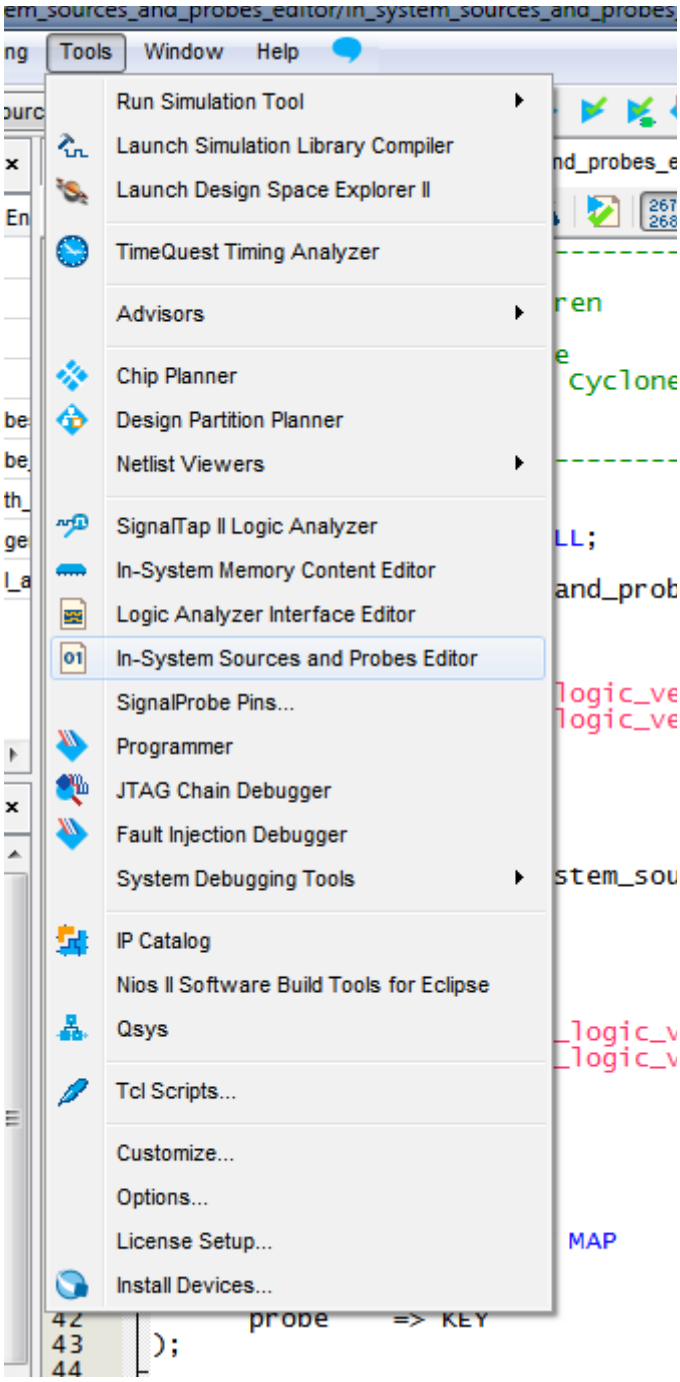


Figure 5: Select In-System Sources and Probes Editor from the Tools menu.

Now is the time to select the file you want to program the hardware with. First you have to select the correct port for the JTAG to talk to the hardware, then you have the .sof file included. Also, make sure the right device is selected. See Figure 6.

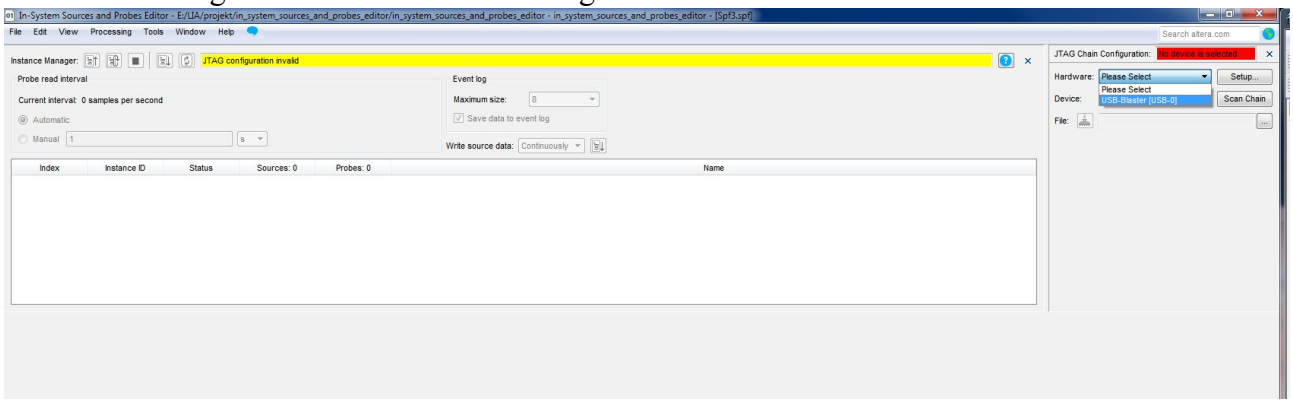


Figure 6: Select the hardware to talk with the device.

Figure 7 shows the download of the .sof file to the hardware.

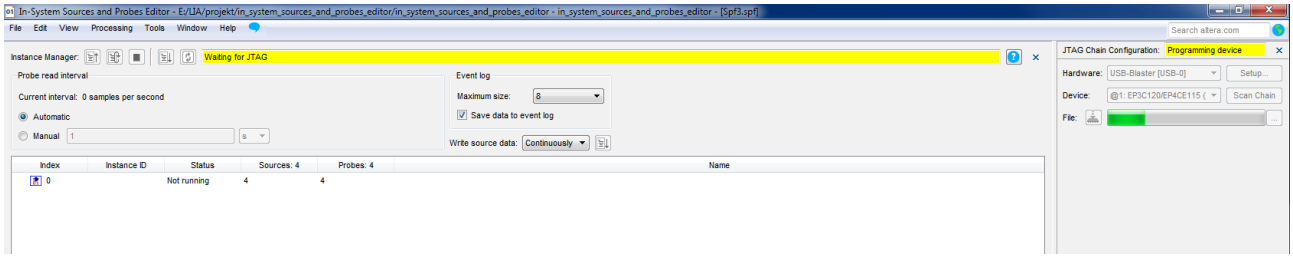


Figure 7: Download of configuration file to the hardware.

In Figure 8, the system is downloaded and the instance found. Here you select Continuous Read Probe Data.

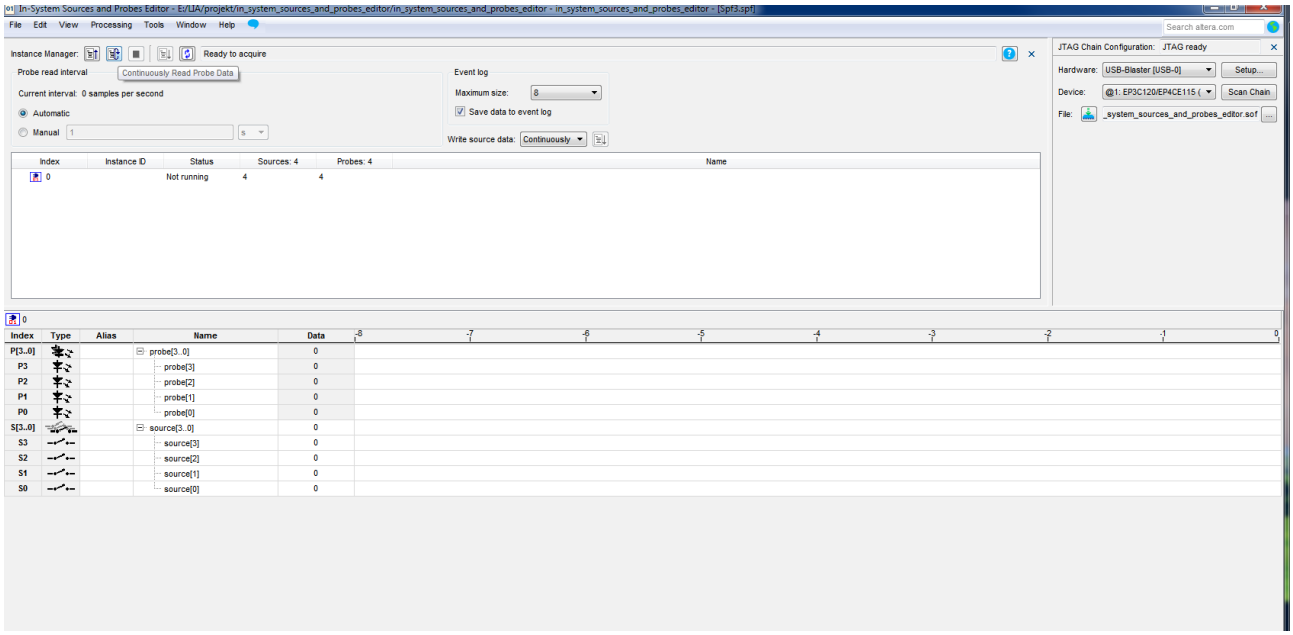


Figure 8: Programming trough the JTAG view.

Below is the continuous reading active and key0 is pressed. Moreover, as seen, ledg 3 is lit from writing directly to source[3] in In-System Sources and Probes Editor itself. More of this can be seen in the next picture.

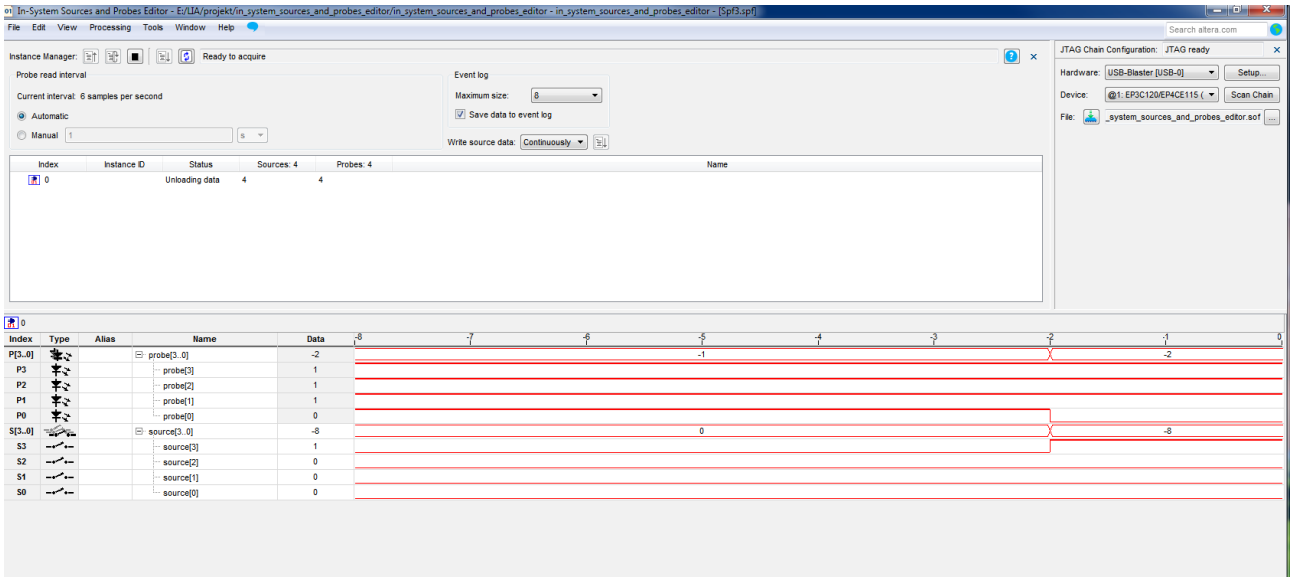


Figure 9: Continuous reading data from a probe and writing to a source.

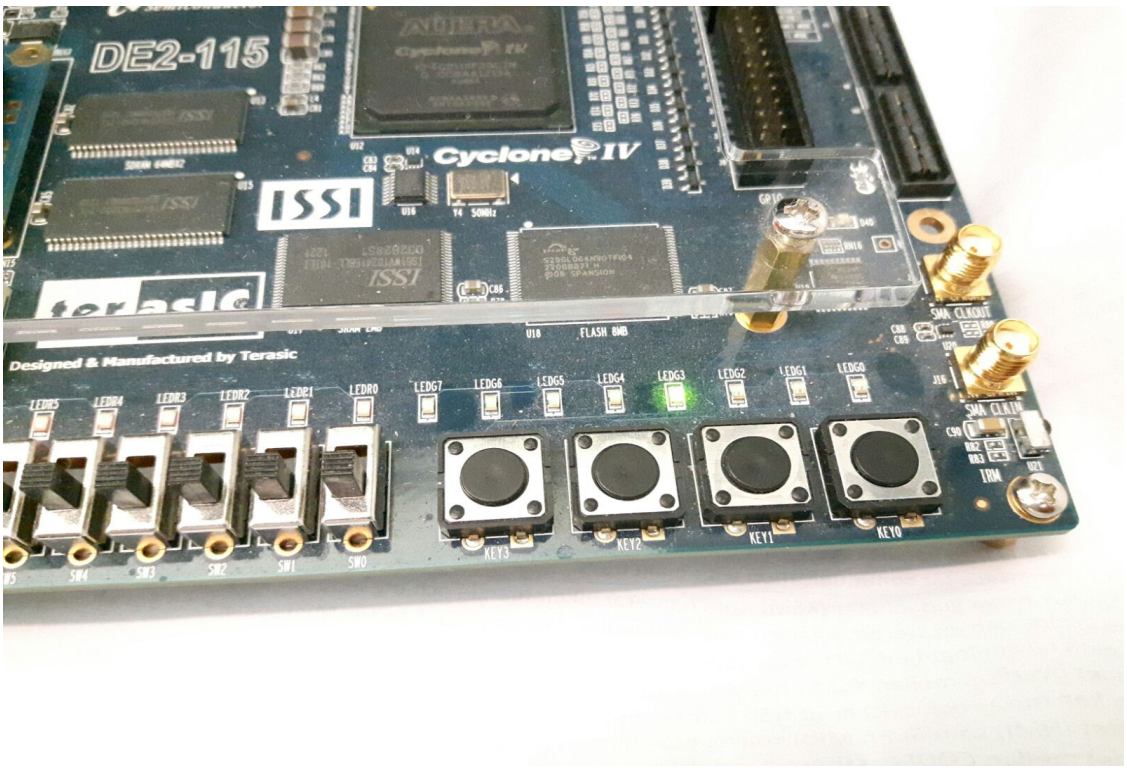


Figure 10: LEDG3 illuminated trough the In-System Sources and Probes Editor.