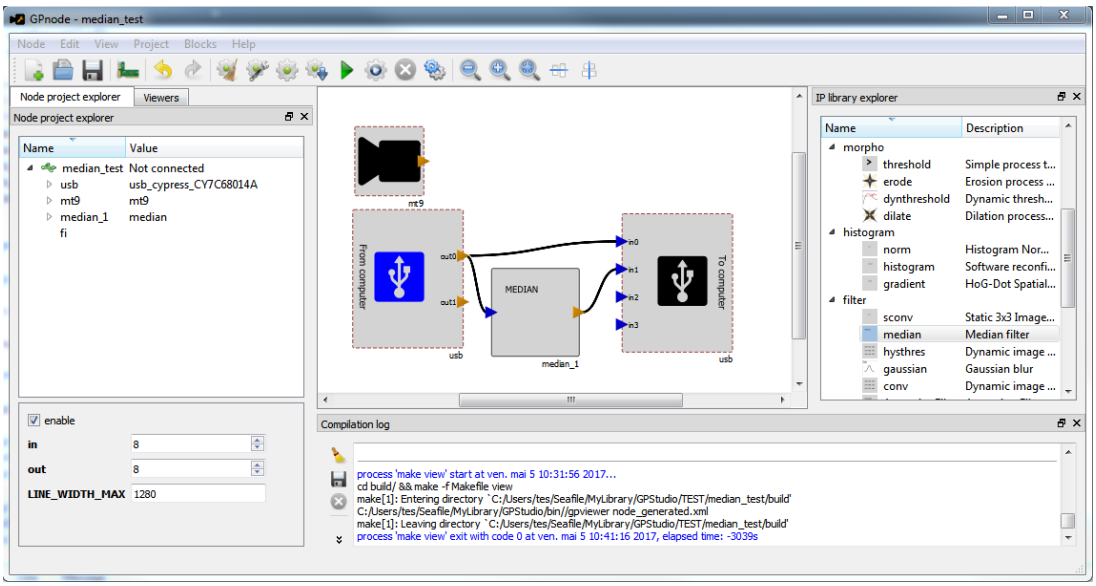
Contents

1 Send an image to the smartcam from the computer

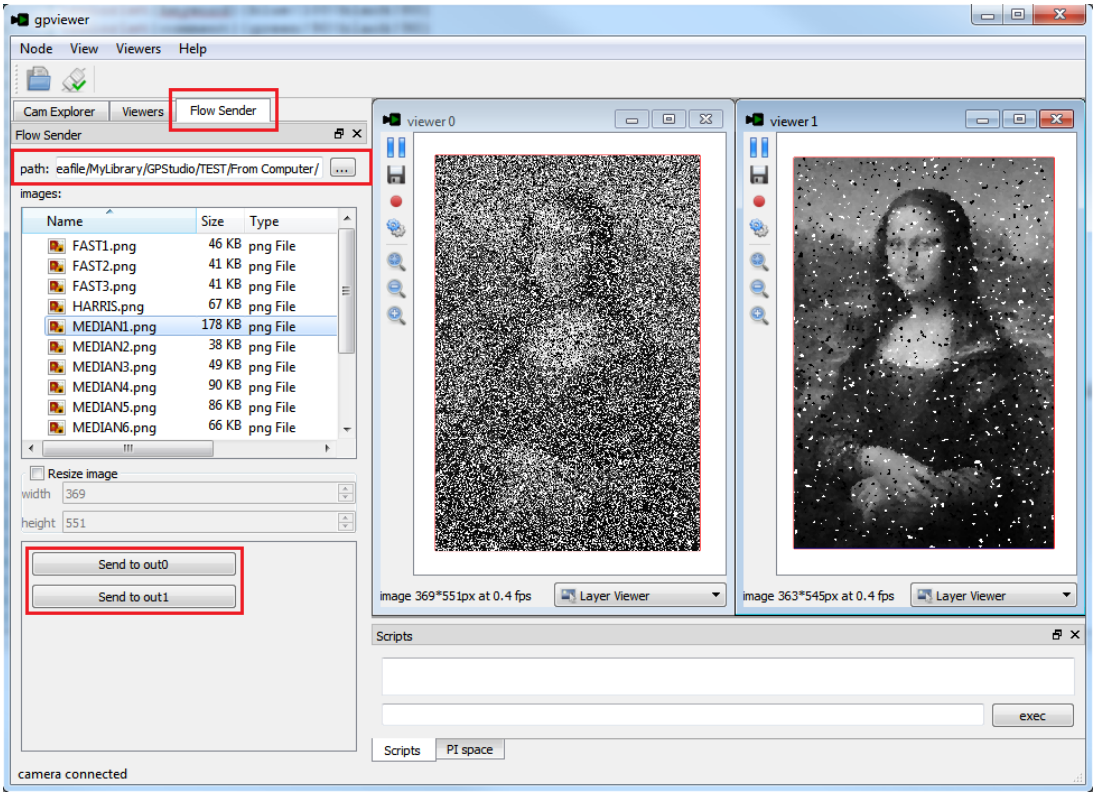
Sometimes, it may be interesting to send an image from the computer to the smart camera.

GPStudio provides this opportunity thanks to a simple option.



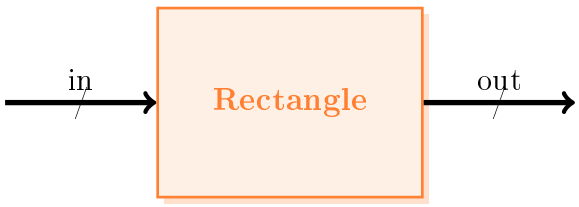
- First of all, create your project in GPnode as showed on the previous figure and save it.
- Generate, compile, send and run the project.
- After having generate, compile, send and run the project, you are directed to the GPviewer. In the GPviewer, select "Flow sender".
- Select the path to the images that have to be sent to the smart camera.
- Choose the image and send it to the out0 port or the out1 port.

The following figure shows the result.



2 Draw a rectangle or several points with gpviewer

2.1 Rectangle



The input flow contains the pixels from the sensor coded on 1 byte. The output flow contains the rectangle coordinates.

The rectangle coordinates are sent on a serial format via a 64 bits frame, with the following disposition :

$$< x; y; w; h >$$

Each component is 16 bits wide and the sending order is left to right, Less Significant Byte first.

To parameter the ip ;

- You have to write the follow script in the **.proc file** for the output flow.

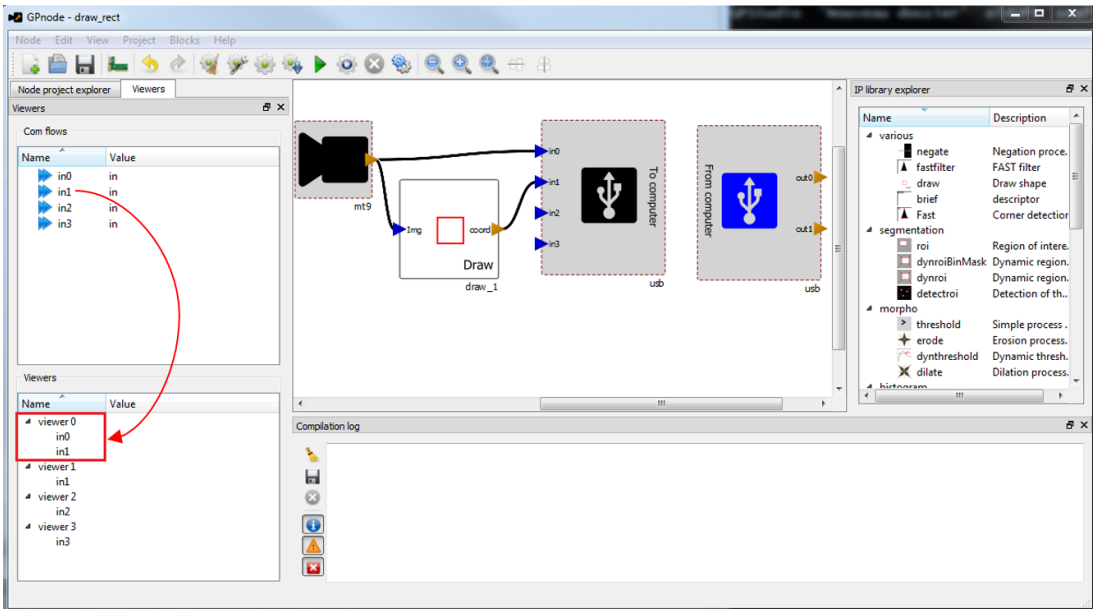
```
<flow name="coord" size="8" type="out">
    <properties>
        <property name="datatype" type="flowtype" value="
            features"/>
        <property name="featuretype" type="featuretype" value="
            rect"/>
    </properties>
</flow>
```

- To implement the **_process.vhd**, read the following script.

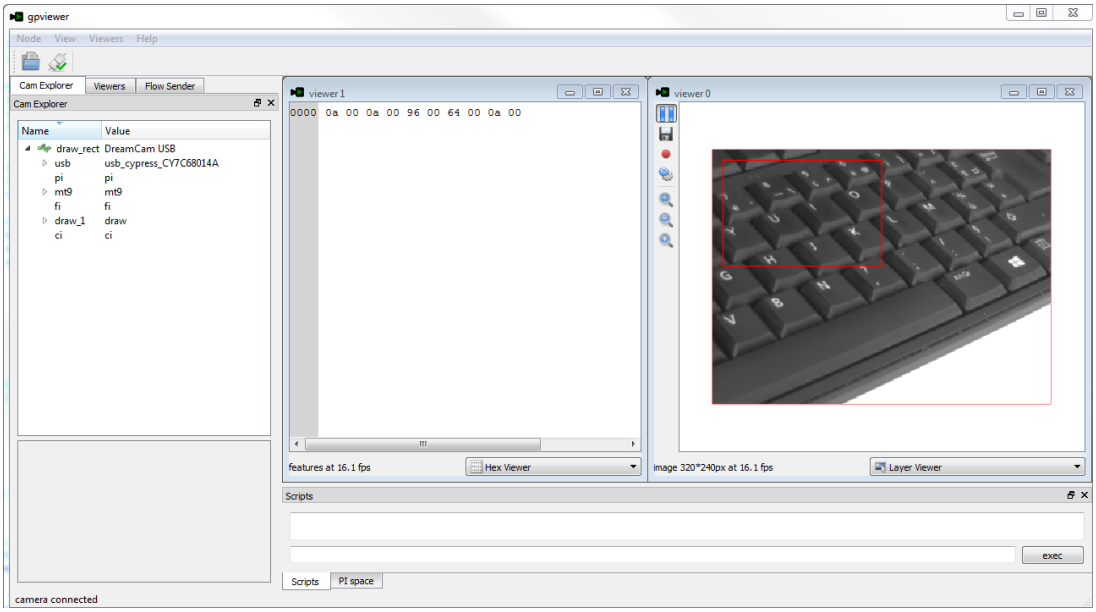
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;
library std;

entity draw_process is
        generic (
            CLK_PROC_FREQ : integer;
            IMG_SIZE      : integer;
            COORD_SIZE    : integer
        );
        port (
            clk_proc      : in std_logic;
            reset_n       : in std_logic;

            ------------------ dynamic parameters ports ------------------
            status_reg_enable_bit     : in std_logic;
            inImg_size_reg_in_w_reg   : in std_logic_vector(11 downto 0);
            inImg_size_reg_in_h_reg   : in std_logic_vector(11 downto 0);

            ------------------------------ Img flow ------------------------------
            Img_data                  : in std_logic_vector(IMG_SIZE-1
                downto 0);
            Img_fv                    : in std_logic;
            Img_dv                    : in std_logic;

            ------------------------------ coord flow ------------------------------
            coord_data                : out std_logic_vector(COORD_SIZE-1
                downto 0);
            coord_fv                  : out std_logic;
            coord_dv                  : out std_logic
        );
end draw_process;

architecture rtl of draw_process is

--process data_process vars
signal enabled  : std_logic;

--Coord over serial line related vars
signal frame_buffer               : std_logic_vector(63 downto 0);
signal frame_buffer_has_been_filled : std_logic;
signal frame_buffer_has_been_sent   : std_logic;
signal frame_buffer_position        : unsigned(6 downto 0);

begin
    data_process : process (clk_proc, reset_n)

    begin
         if(reset_n='0') then
            --Cleaning frame buffer
            frame_buffer    <= (others=>'0');
            --Cleaning signals used to fill buffer
            frame_buffer_has_been_filled <= '0';
            frame_buffer_has_been_sent   <= '0';
            coord_fv    <= '0';
            coord_dv    <= '0';
            coord_data  <= (others=>'0');
            --Cleaning flags
            enabled     <= '0';

                elsif(rising_edge(clk_proc)) then
                   coord_fv    <= '0';
                   coord_dv    <= '0';
                   coord_data  <= (others=>'0');
```

```
64
65                      if(Img_fv = '0') then
66                          --
67                          if(frame_buffer_has_been_filled = '0')then
68                              --We send frame coordinates only if there is
                                    something to send
69                              if(enabled = '1' and
                                    frame_buffer_has_been_sent = '0')then
70                                  frame_buffer <= (others => '0');
71
72                                  --filling buffer with matching coordinates
73                                  frame_buffer(15 downto 0)  <=
                                        std_logic_vector(to_unsigned(10,16));
                                        -- x
74                                  frame_buffer(31 downto 16) <=
                                        std_logic_vector(to_unsigned(10,16));
                                        -- y
75                                  frame_buffer(47 downto 32) <=
                                        std_logic_vector(to_unsigned(100,16));
                                         -- w
76                                  frame_buffer(63 downto 48) <=
                                        std_logic_vector(to_unsigned(100,16));
                                         -- h
77
78                                  -- Get buffer ready to send
79                                  frame_buffer_has_been_filled <= '1';
80                                  frame_buffer_position        <= (others
                                        =>'0');
81                              end if;
82                          else
83                              --send coord
84                              coord_fv <= '1';
85                              coord_dv <= '1';
86                              coord_data <= frame_buffer(to_integer(
                                    frame_buffer_position)+7 downto to_integer
                                    (frame_buffer_position));
87
88                              if(frame_buffer_position >= 65)then
89                                  frame_buffer_has_been_filled <= '0';
90                                  frame_buffer_has_been_sent   <= '1';
91                              else
92                                  frame_buffer_position <=
                                        frame_buffer_position + to_unsigned
                                        (8,8);
93                              end if;
94                          end if;
95                          enabled  <= status_reg_enable_bit;
96                      else
97                          coord_fv         <= '0';
98                          coord_dv         <= '0';
99                          coord_data       <= (others=>'0');
100                         frame_buffer_has_been_sent       <= '0';
101                     end if;
102                 end if;
103     end process;
104 end rtl;
```

- The last manipulation to do before running the compilation is to add the output flow containning the coordinates in the viewer that receives the output flow from the sensor. As showed at the follow figure.



After having run the compilation, we get the follow result in gpviewer.

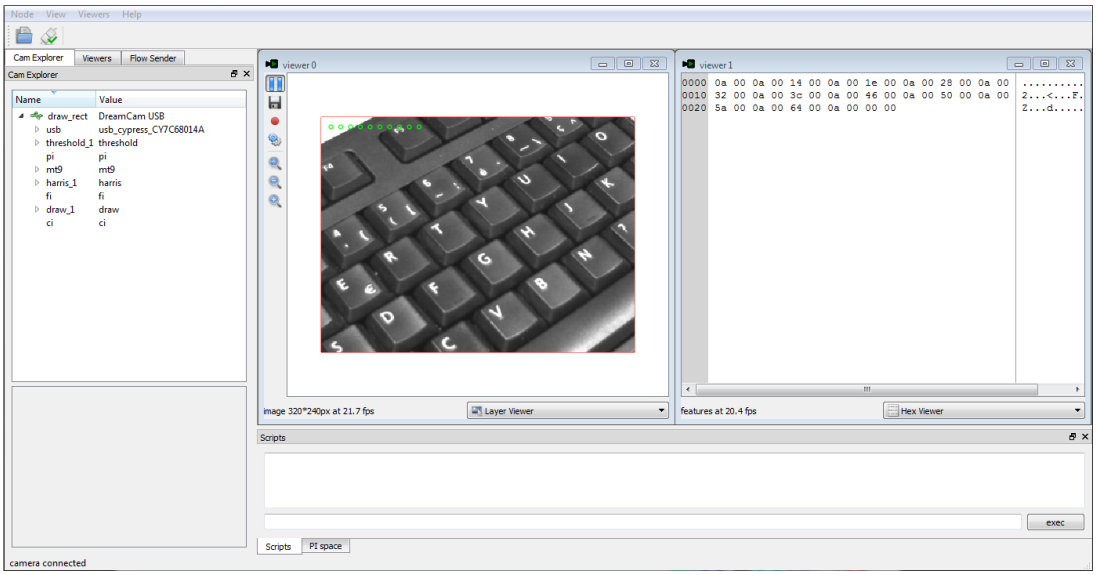


2.2 Several points

To draw several points in a viewer of gpviewer, just modify the value of the featuretype.

```
<flow name="coord" size="8" type="out">
    <properties>
        <property name="datatype" type="flowtype" value="features"/>
        <property name="featuretype" type="featuretype" value="point
            "/>
    </properties>
</flow>
```

After having run the compilation, we get the follow result in gpviewer.



For the both options, draw a rectangle or several points, the main structure of the file is the same. Let see the follow pseudo-code.

Initialization:

```
signal frame_buffer
signal frame_buffer_filled
signal frame_buffer_sent
signal frame_buffer_position

if (reset = 0) then
    frame_buffer ←' 0...0';
    frame_buffer_filled ←' 0';
    frame_buffer_sent ←' 0';
else if (rising_edge(clk)) then
    out_fv ←' 0';
    out_dv ←' 0';
    out_data ←' 0...0';
    if (in_fv = 1) then
        \\ your script
    else if (in_fv = 0) then
        if (frame_buffer_filled = 0) then
            if (frame_buffer_sent = 0) then
                Filling of the frame buffer;
                frame_buffer_filled ←' 1';
                frame_buffer_position ←' 0...0';
            end
        else
            out_fv ←' 1';
            out_dv ←' 1';
            out_data ← frame_buffer((frame_buffer_position)
                        + 7 downto (frame_buffer_position));
            if (frame_buffer_position ≥ Value[a]) then
                signal frame_buffer_sent ←' 1';
                signal frame_buffer_filled ←' 0';
            else
                Incrementation of the frame_buffer_position value;
            end
        end
    else
        out_fv ←' 0';
        out_dv ←' 0';
        out_data ←' 0...0';
        frame_buffer_sent ←' 0';
    end
end
```

[a] $Value = 32n + 1$, where n is the number of points that you want to send on the output flow. If you want to draw a rectangle, $n = 2$.