UMONS
Université de Mons

Hardware Software Platforms

Lecteur de fréquences

Bourgeois J.
Grugeon L.

POLYTECH MONS

Cahier de charges et Méthodologie

- Fréquence reçue *f* comprise entre 50 Hz et 50 kHz
- Registre d'écriture de 8 bits

- **Division** de la gamme ($\Delta F = 49\,950$ Hz) en 256 plages uniformément réparties
- **Codage** de *f* par la fréquence f_k associée à la plage dans laquelle *f* se situe
- **Ecriture** de la valeur f_k dans le registre

- **Affichage** via un terminal du numéro de plage *k* (0 , ... , 255) et de f_k (50 , 245 , ... , 50 000)

Hardware

```
entity FreqIn is
   port  (
         clk             : in std_logic;
         reset_in        : in std_logic;
         enable_in       : in std_logic;
         reg0_value_out  : out std_logic_vector(7 downto 0);
         gpio_in         : in std_logic
   );

end entity FreqIn;

architecture RTL of FreqIn is

   type     state_type is (s0,s1,s2,s3);
   signal   state : state_type;
```

```
process(clk, reset_in, enable_in, gpio_in)

   variable count : natural := 0;
   constant f_clk : natural := 50000000;

begin

   if reset_in = '0' then
      count := 0;
      state <= s0;

   elsif rising_edge(clk) then

      case state is

         when s0 =>
            count := 0;
            if enable_in = '1' then
               if gpio_in = '1' then
                  state <= s1;
               end if;
            end if;

         when s1 =>
            count := count + 1;
            if gpio_in = '0' then
               state <= s2;
            end if;

         when s2 =>
            count := count + 1;
            if gpio_in = '1' then
               reg0_value_out <= std_logic_vector(to_unsigned(((f_clk/count)-50)/195,8)
               count := 0;
               if enable_in = '1' then
                  state <= s1;
               else
                  state <= s0;
               end if;
            end if;

         when others =>
            count := 0;
            state <= s0;
            --test
      end case;

   end if;
```

$$k = FLOOR\ [(f - 50) / 195]$$

Hardware : Test Bench

```
architecture TB of FreqIn_TB is

    constant PERIOD          : time := 20 ns;
    signal clk               : STD_LOGIC := '0';
    signal reset_in          : std_logic := '0';
    signal senable_in        : std_logic := '0';
    signal sreg0_value_out   : std_logic_vector(7 downto 0);
    signal sgpio_in          : std_logic;

    component FreqIn is
       port  (

             clk             : in std_logic;
             reset_in        : in std_logic;
             enable_in       : in std_logic;
             reg0_value_out  : out std_logic_vector(7 downto 0);
             gpio_in         : in std_logic

             );
    end component;
```

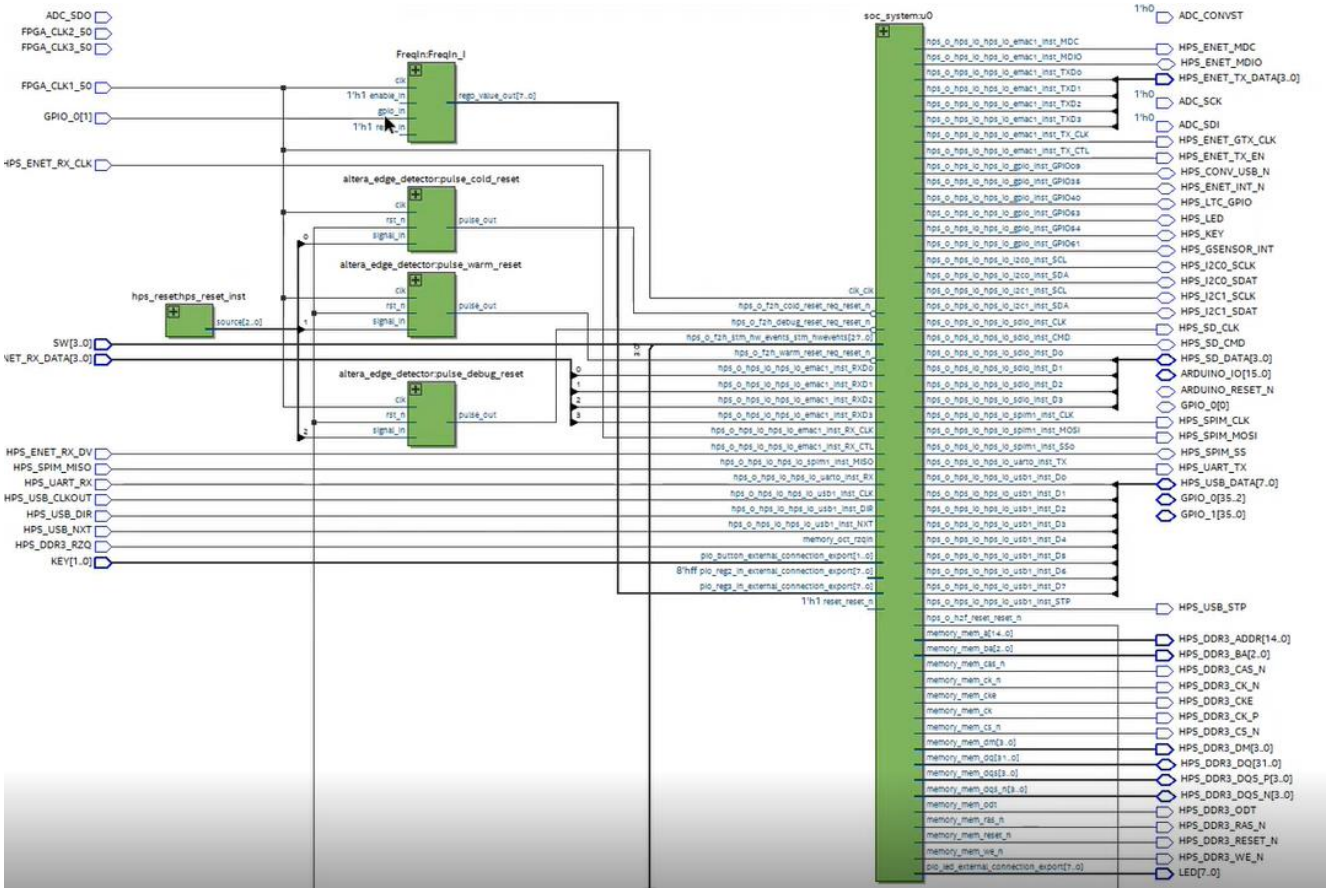
```
begin

    i1: FreqIn
    port map (
       clk             =>clk,
       reset_in        =>reset_in,
       enable_in       =>senable_in,
       reg0_value_out => sreg0_value_out,
       gpio_in         =>sgpio_in
    );


    reset_in_P: process
    begin
       reset_in <= '0';
       wait for PERIOD ;
       reset_in <= '1';
       wait;
    end process;

    clk_P: process
    begin
       clk <= '0';
       wait for PERIOD/2;
       clk <= '1';
       wait for PERIOD/2;
    end process;
```

```
stimulus: process
begin
   sgpio_in <= '0';
   if reset_in = '0' then
      wait until reset_in = '1';
   end if;


   for i in 0 to 2 loop
      senable_in <= '1';
      sgpio_in <= '1';
      wait for 1 ms;
      senable_in <= '0';
      sgpio_in <= '0';
      wait for 20 ms - 1 ms;
   end loop;


   for i in 0 to 2 loop
      senable_in <= '1';
      sgpio_in <= '1';
      wait for 1 ms;
      senable_in <= '0';
      sgpio_in <= '0';
      wait for 3.33 ms - 1 ms;
   end loop;
```

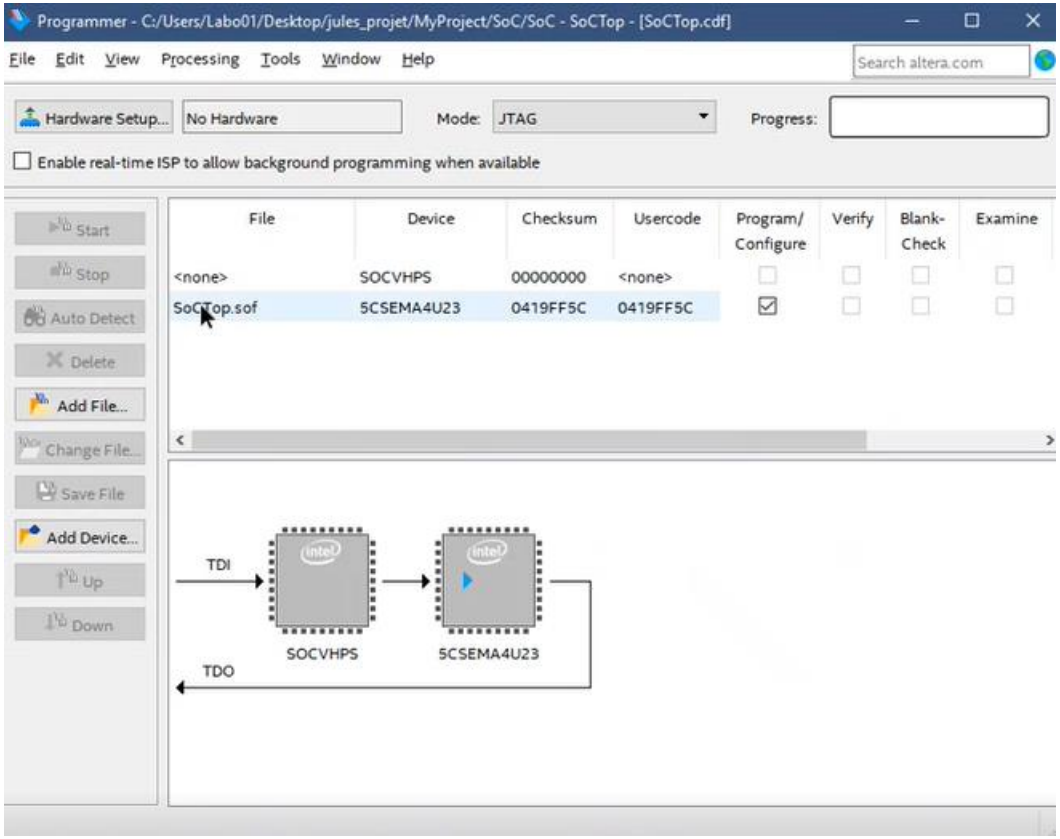
Hardware : Vue RTL

Succession de :	0	et	1
Fréquence :	50 Hz		300 Hz
Numéro de plage :	k(50) = 0		k(300) = 1

Hardware : PINs & Blocs



Hardware : FPGA

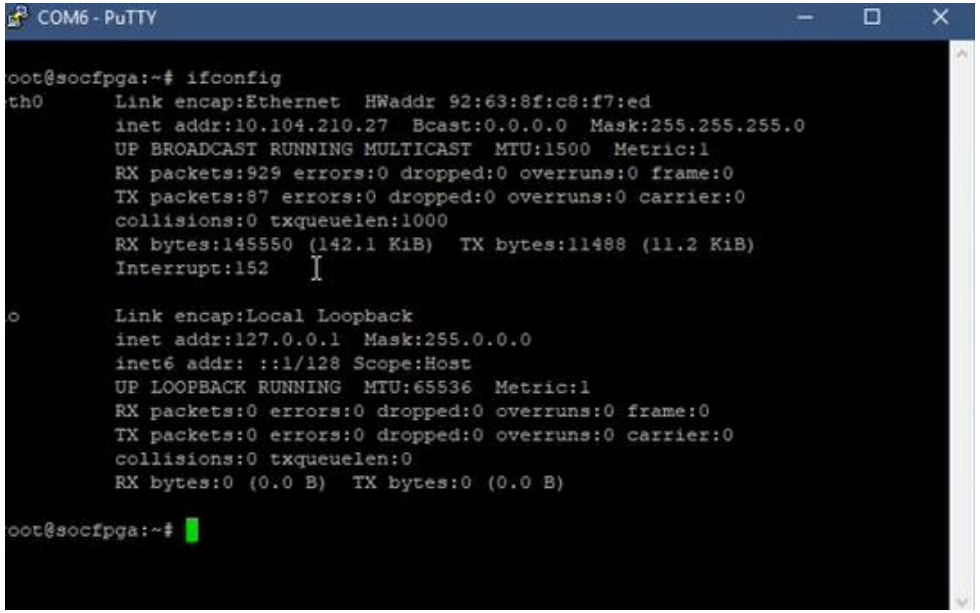


Software

```c
if( virtual_base == MAP_FAILED ) {
   printf( "ERROR: mmap() failed...\n" );
   close( fd );
   return( 1 );
}


printf("\n\n\n----------2----------x \n\n" );

iter = atoi(argv[1]);

printf("\n\n\n----------3----------%d \n\n", iter );

h2p_lw_reg3_addr=virtual_base + ( ( unsigned long )( ALT_LWFPGASLVS_OFST + PIO_REG3_IN_BASE ) & ( unsigned long )( HW_REGS_MASK ) );

printf("\n\n\n----------4---------- \n\n" );

for(;x<iter; x++){
   //printf("\n\n\n----------4---------- \n\n" );
   k_1 = *((uint32_t *)h2p_lw_reg3_addr)*195+50;
   printf( "Numero de la plagr  : %d\n", *((uint32_t *)h2p_lw_reg3_addr));
   printf( "Frequence :  %d\n", k_1);
   sleep(1);
}
printf("\n\n\n----------5---------- \n\n" );
if( munmap( virtual_base, HW_REGS_SPAN ) != 0 ) {
   printf( "ERROR: munmap() failed...\n" );
   close( fd );
   return( 1 );
}
close( fd );

return( 0 );
```

$$f_k = (k \,.\, 195\,) + 50$$

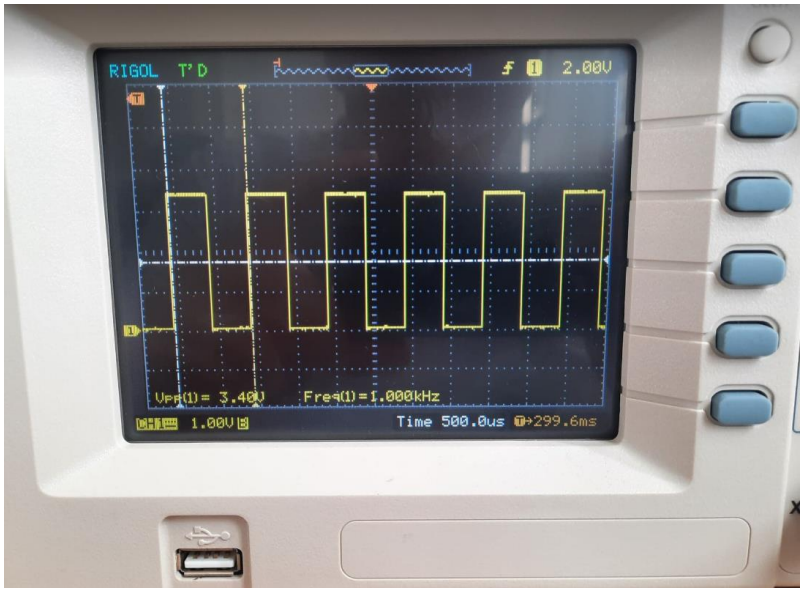
PuTTY : Connexion et Adresse



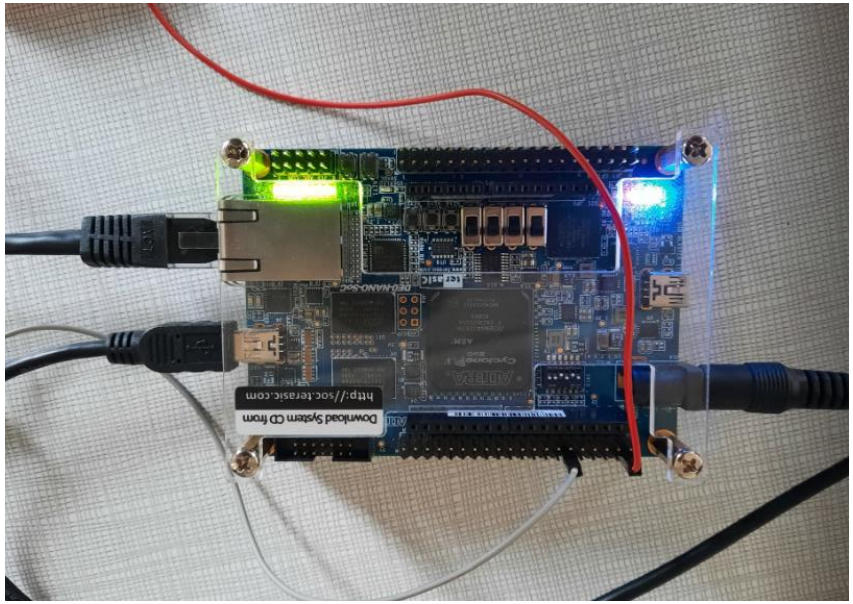
SoC : Téléchargement de l’exécutable

```
~/Desktop/jules_projet/MyProject/source
abo01@LABO06 ~
$ cd Desktop/jules_projet/MyProject/source

abo01@LABO06 ~/Desktop/jules_projet/MyProject/source
$ ls
_history  HPS-FPGA-TEMPLATE  main.c.std    main.err   main.o     main-std.c   Makefile
ps_0.h    main.c             main.c-ori.c  main.esym  main.xsym  main-test.c

abo01@LABO06 ~/Desktop/jules_projet/MyProject/source
$ make clean
m -f HPS-FPGA-TEMPLATE *.a *.o *~

abo01@LABO06 ~/Desktop/jules_projet/MyProject/source
$ make
rm-linux-gnueabihf-gcc -static -g -Wall  -IC:/intelFPGA/18.1/embedded/ip/altera/hps/altera_hps/hwlib/include -c main.c
o main.o
rm-linux-gnueabihf-gcc -g -Wall     main.o -o HPS-FPGA-TEMPLATE

abo01@LABO06 ~/Desktop/jules_projet/MyProject/source
$ scp HPS-FPGA-TEMPLATE root@10.104.210.27:/home/root
ould not create directory '/home/Labo01/.ssh'.
he authenticity of host '10.104.210.27 (10.104.210.27)' can't be established.
CDSA key fingerprint is SHA256:dwbpGGyAksQiqqLe0QwW4CTuT9HRlZgkm2NCKdWKwYA.
re you sure you want to continue connecting (yes/no)? yes
ailed to add the host to the list of known hosts (/home/Labo01/.ssh/known_hosts).
oot@10.104.210.27's password:
PS-FPGA-TEMPLATE                                                    100% 9036     8.8KB/s   00:00

abo01@LABO06 ~/Desktop/jules_projet/MyProject/source
```

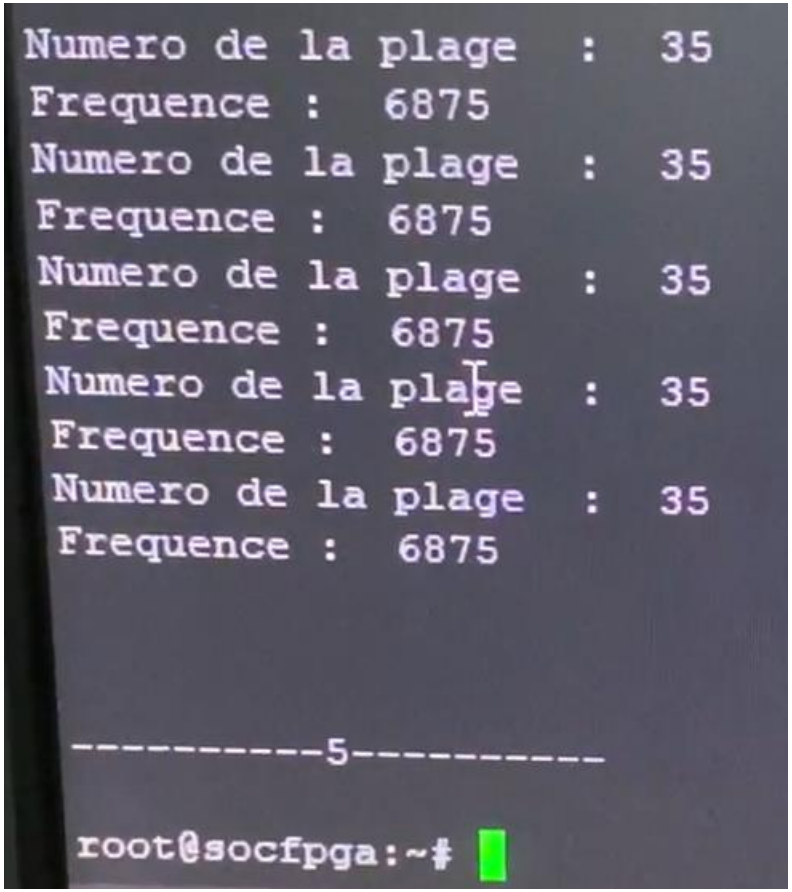
Signal d’entrée



Carte FPGA



Résultats



$$k = FLOOR\,[(7000 - 50) / 195]$$
$$= 35$$

$$f_k = (35 \,.\, 195) + 50$$
$$= 6875 \text{ Hz}$$