

Enseignes et afficheurs à LED

Commande d'une matrice à LED avec une FPGA



Dr. Mamadou Lamine NDIAYE

Commande d'une matrice à LED avec une carte FPGA



Mamadou Lamine NDIAYE

Commande d'une matrice à LED avec une carte FPGA



- Présentation de la matrice à LED
- Contrôle par FPGA de la matrice à LED
- Simulation des signaux de contrôle
- Test sur la carte Adafruit

Présentation de la matrice à LED



Matrice 16X32(Adafruit)

- 512 Led RGB rangées sous forme de matrice 16X32
- Nécessite 12 pins (6 données et 6 contrôle) et une alimentation de 5V (2A minimum suggérée)

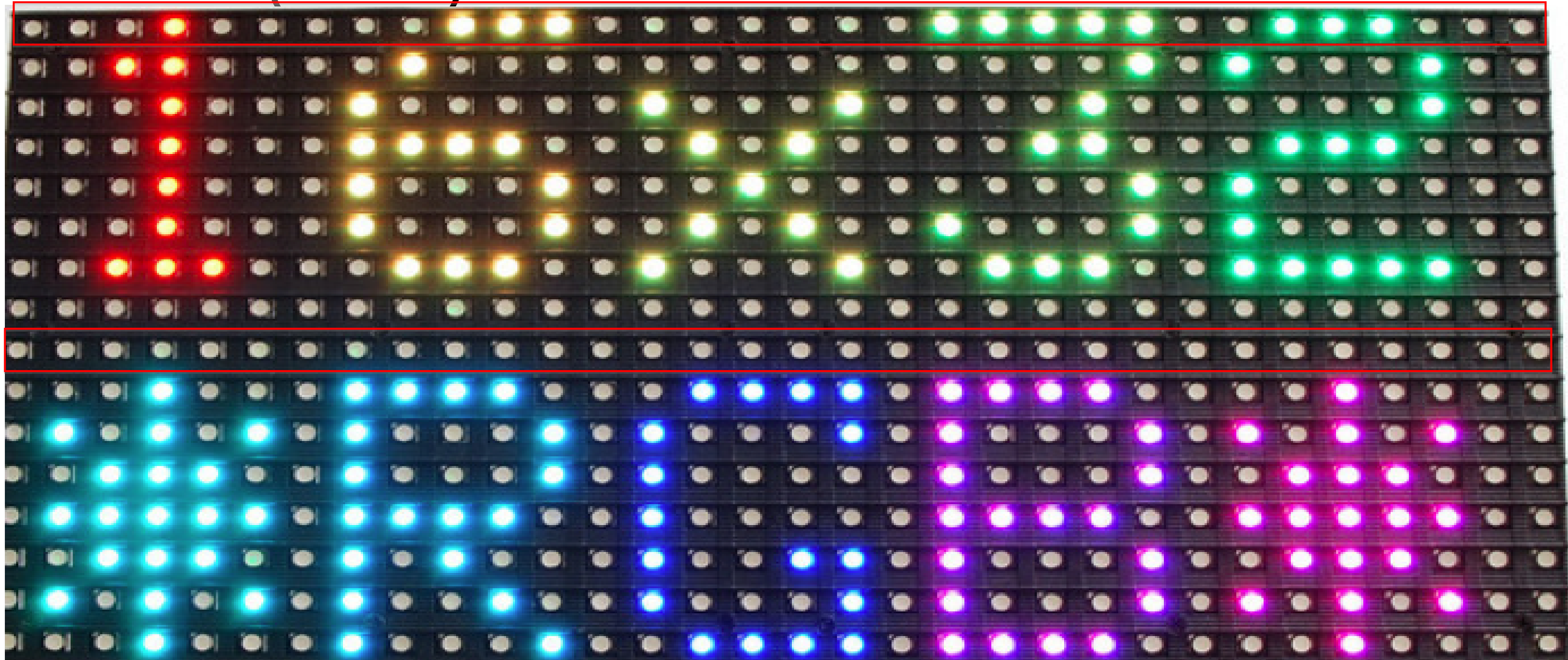
Matrice 32X32(Adafruit)

- 1024 Led RGB rangées sous forme de matrice 32X32
- Nécessite 13 pins (6 données et 7 contrôle) et une alimentation de 5V (4A minimum suggérée)
- Contrôle par Microcontrôleur et carte FPGA

Présentation de la matrice à LED



Matrice 16X32(Adafruit)



Présentation de la matrice à LED



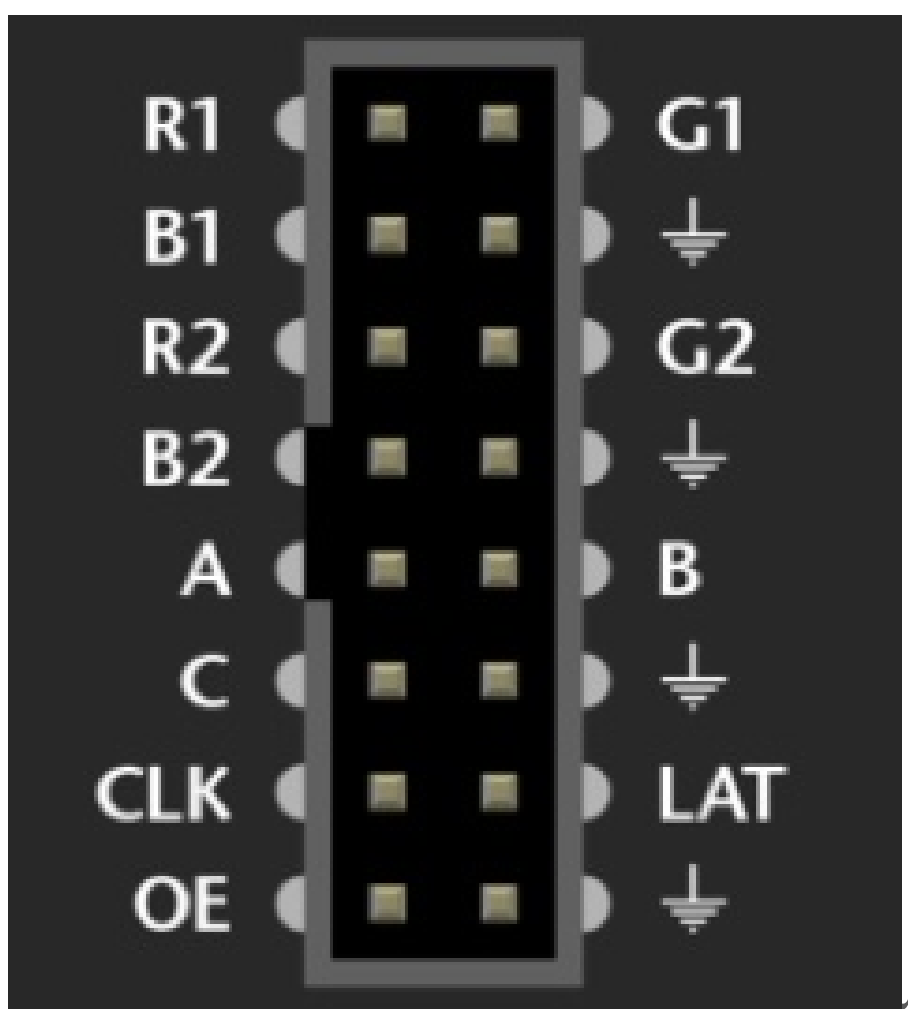
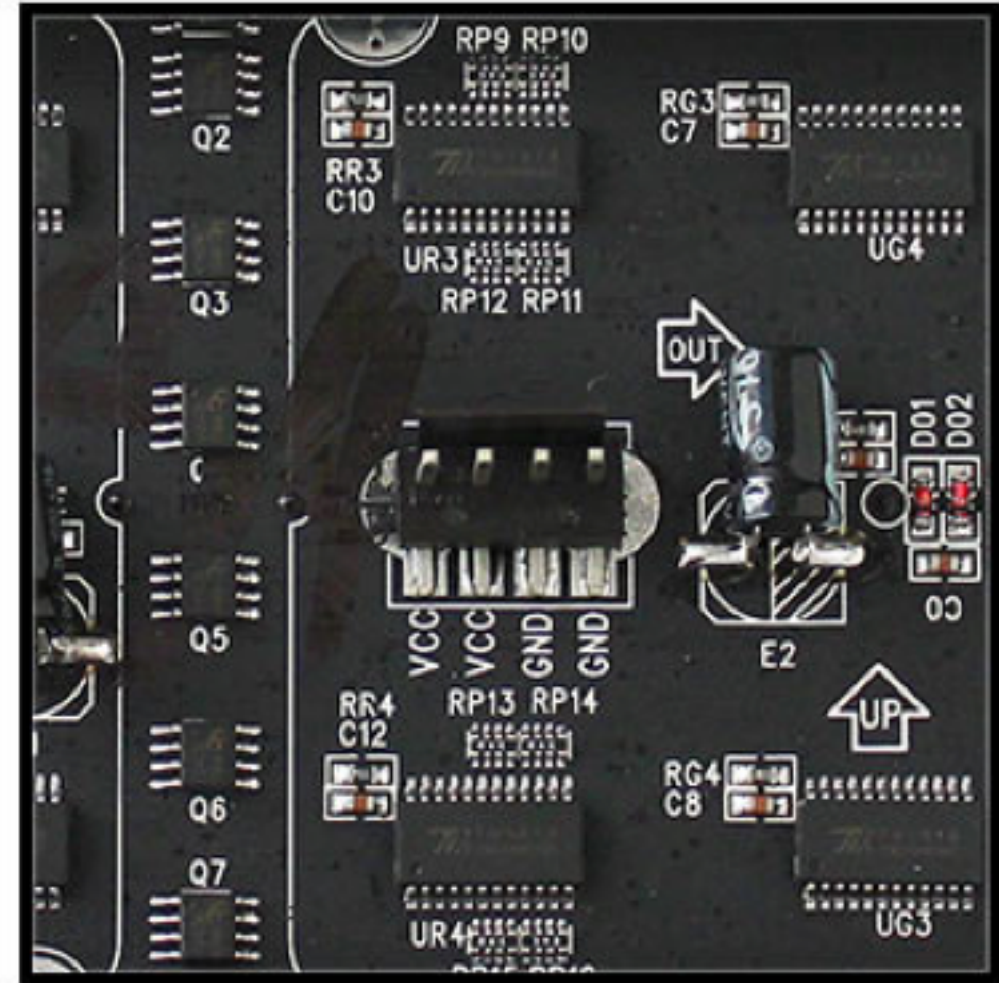
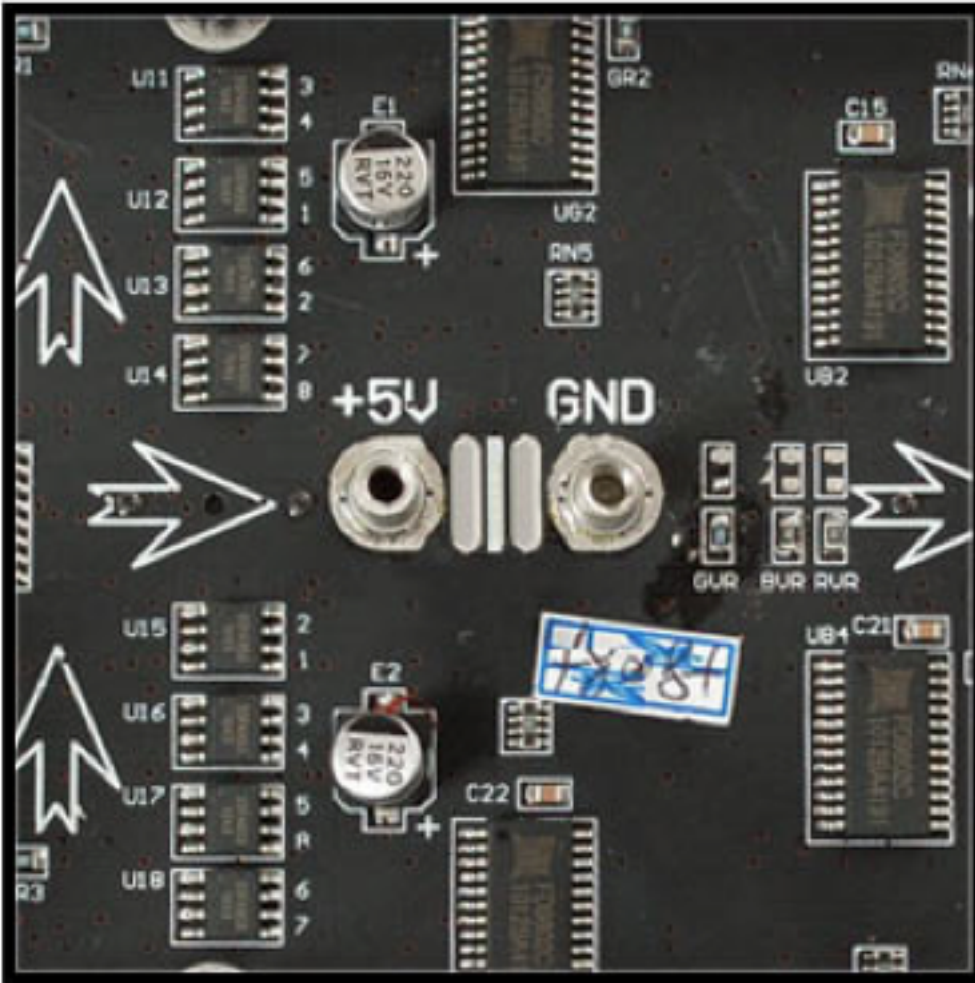
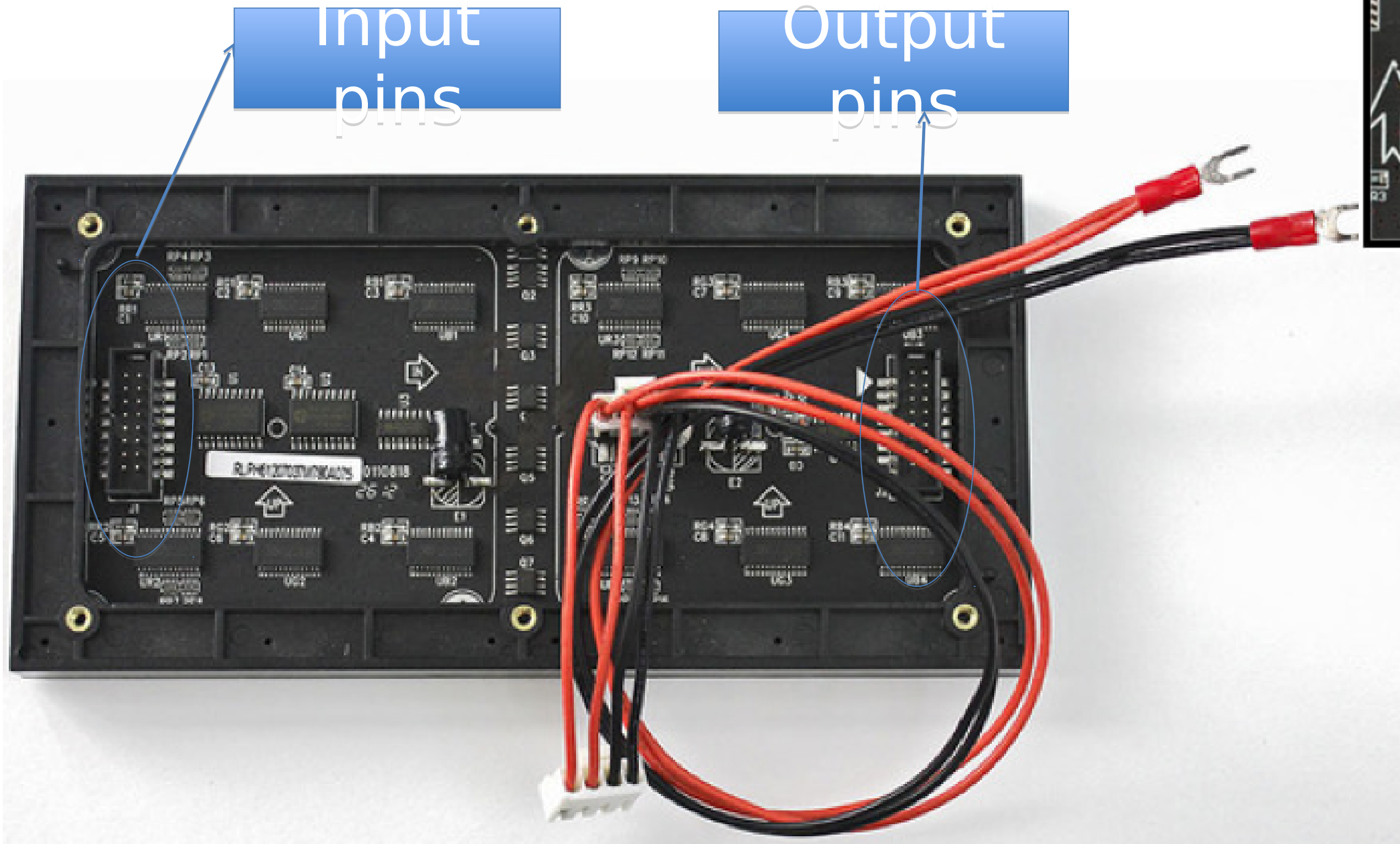
Matrice 32X32(Adafruit)



Matrice à LED RGB 16X32 de Adafruit

Alimentation

● Existe en 2 types de connecteur pour l'alimentation

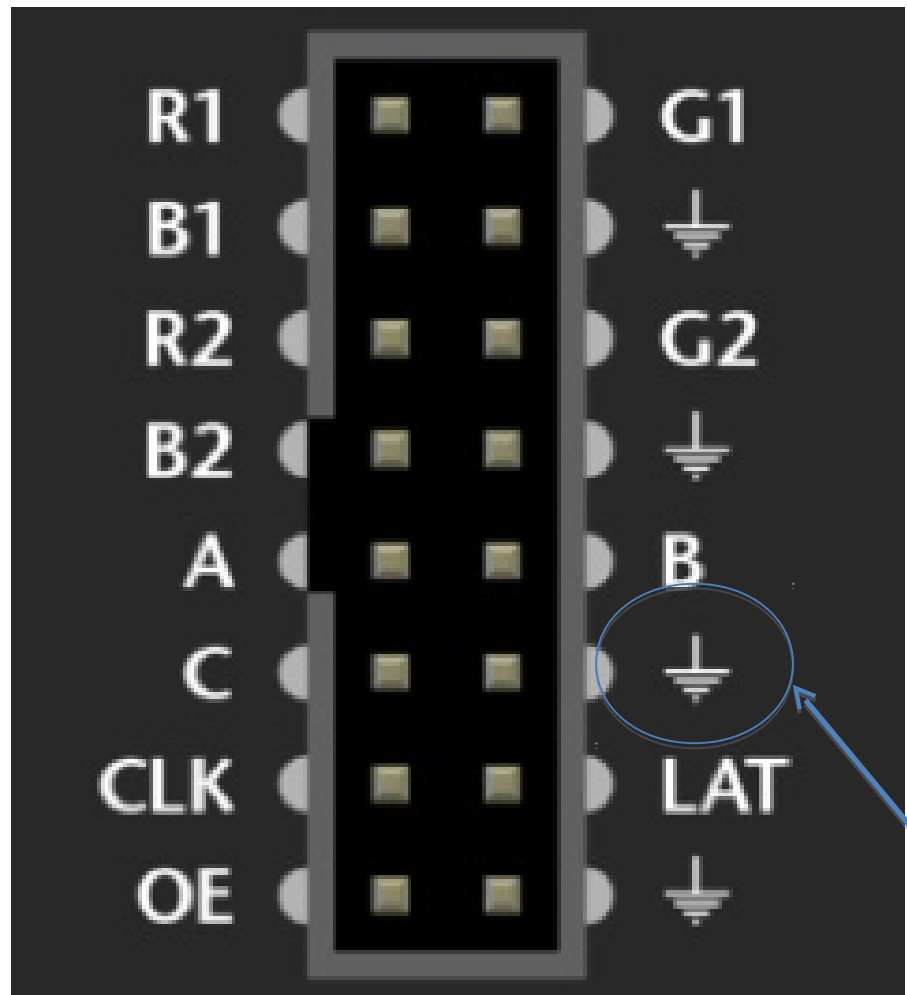
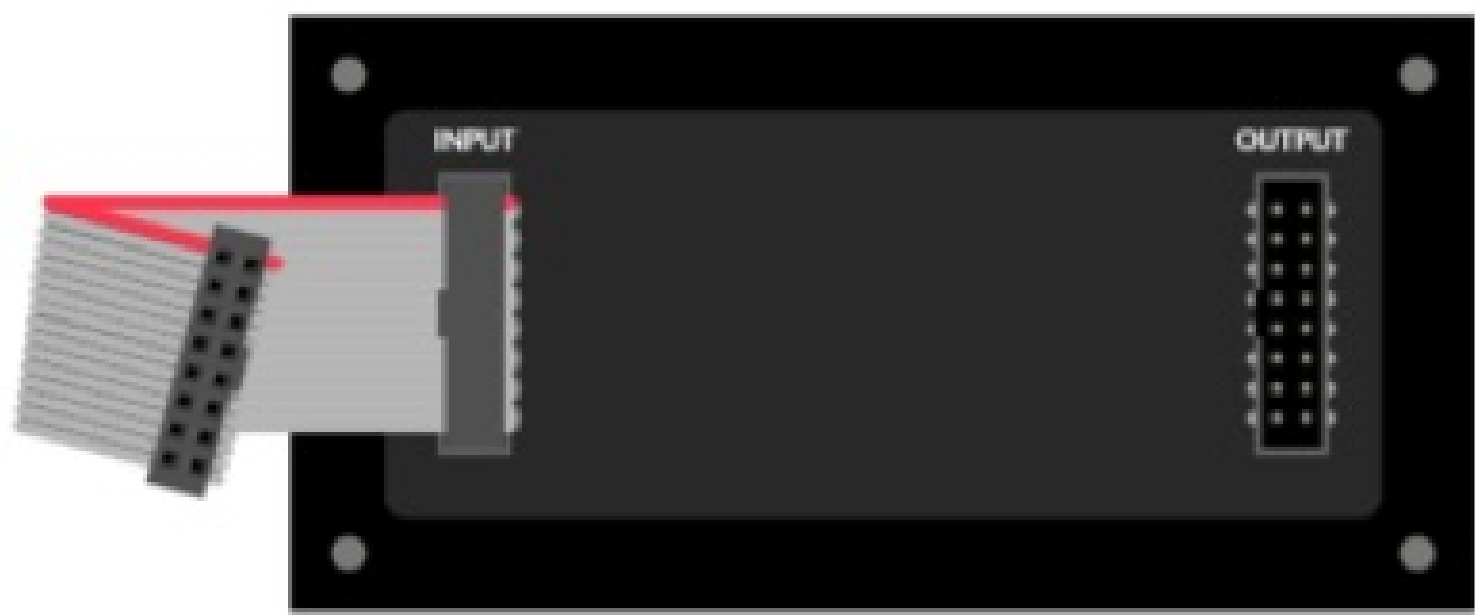
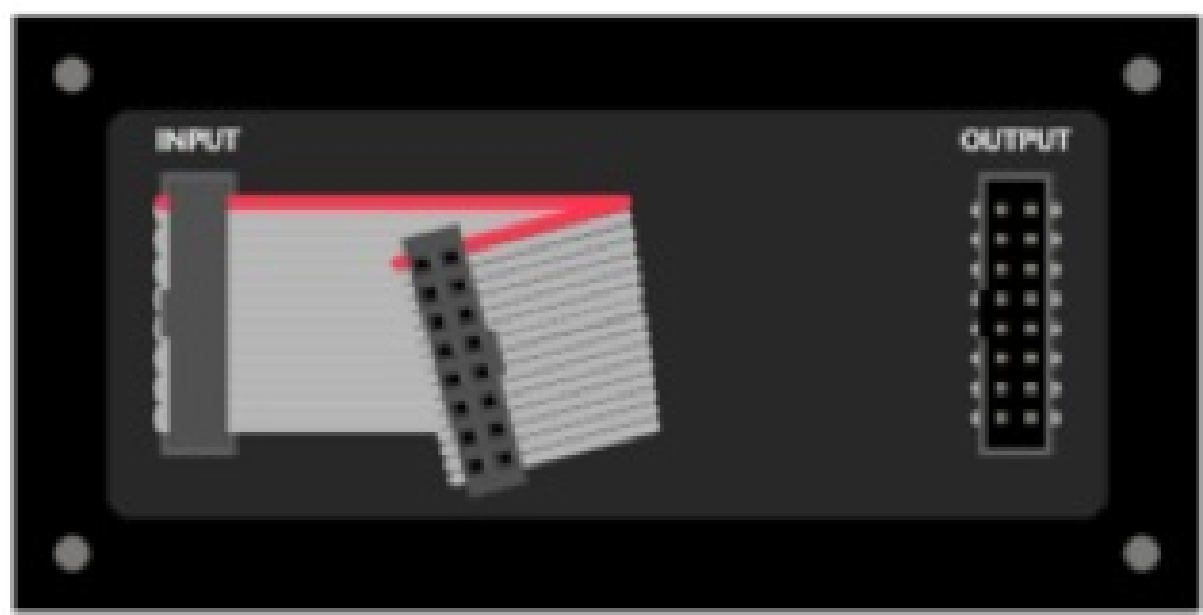


Input pins

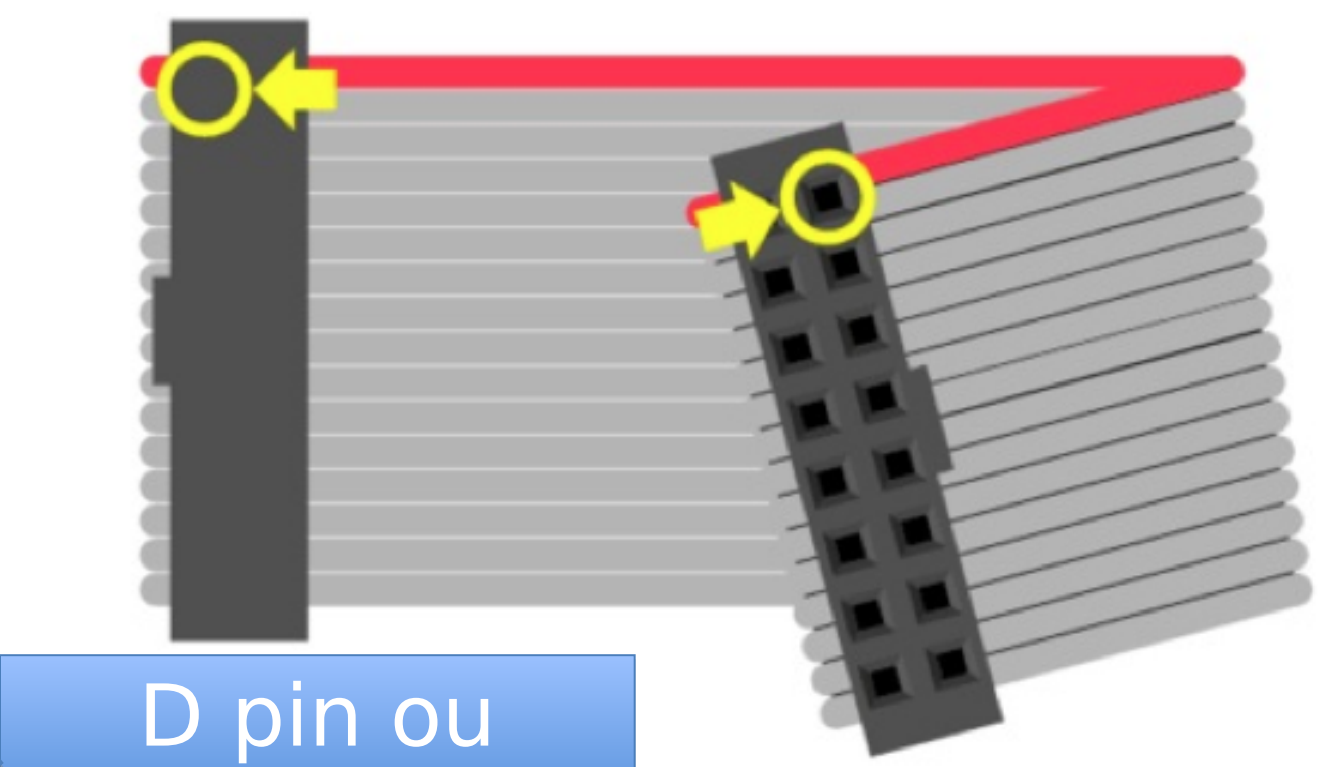
Matrice à LED RGB 16X32 de Adafruit

Connecteur IDC

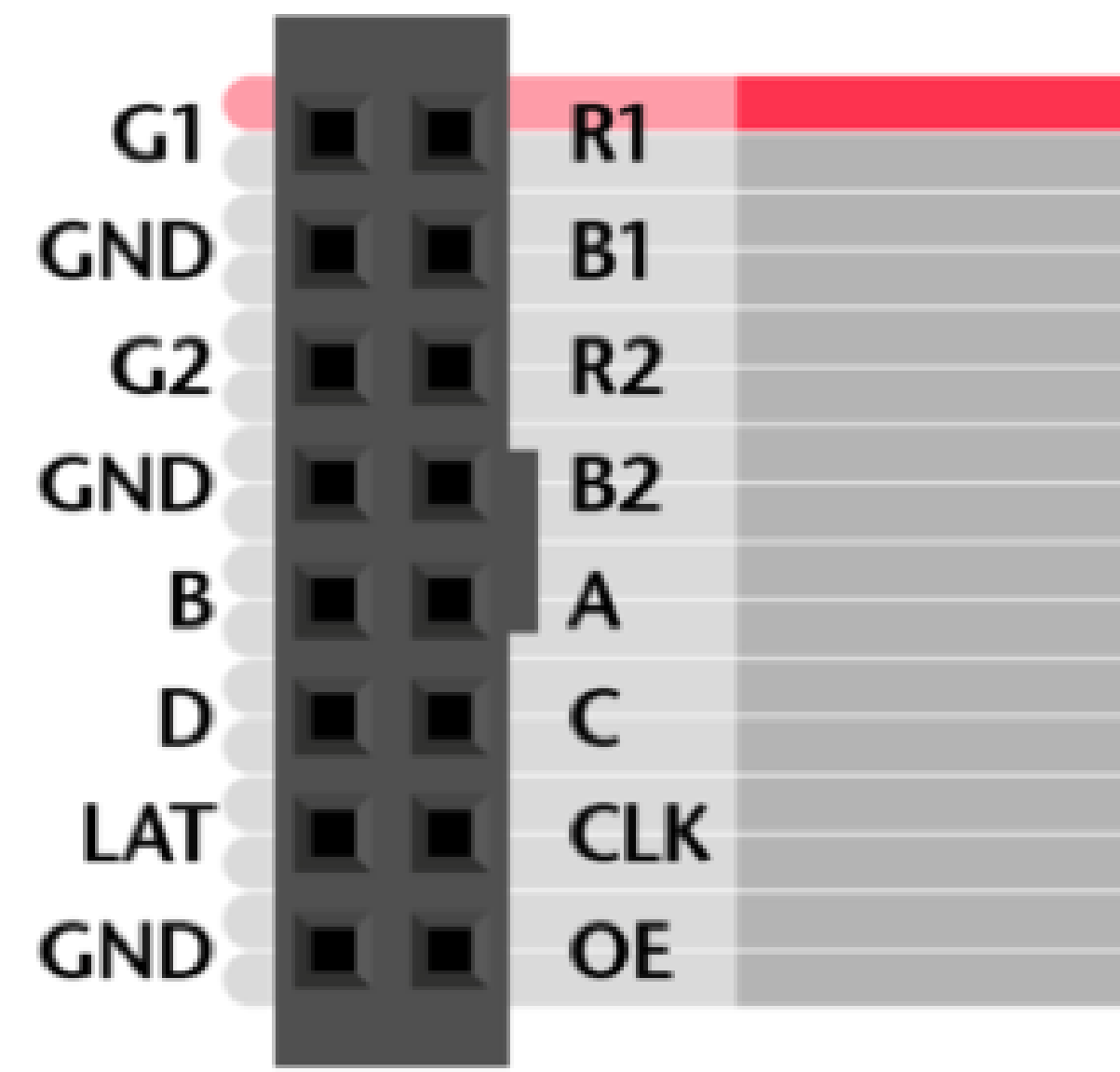
● Existe 2 connecteurs (input et output)



Input pins



D pin ou Masse

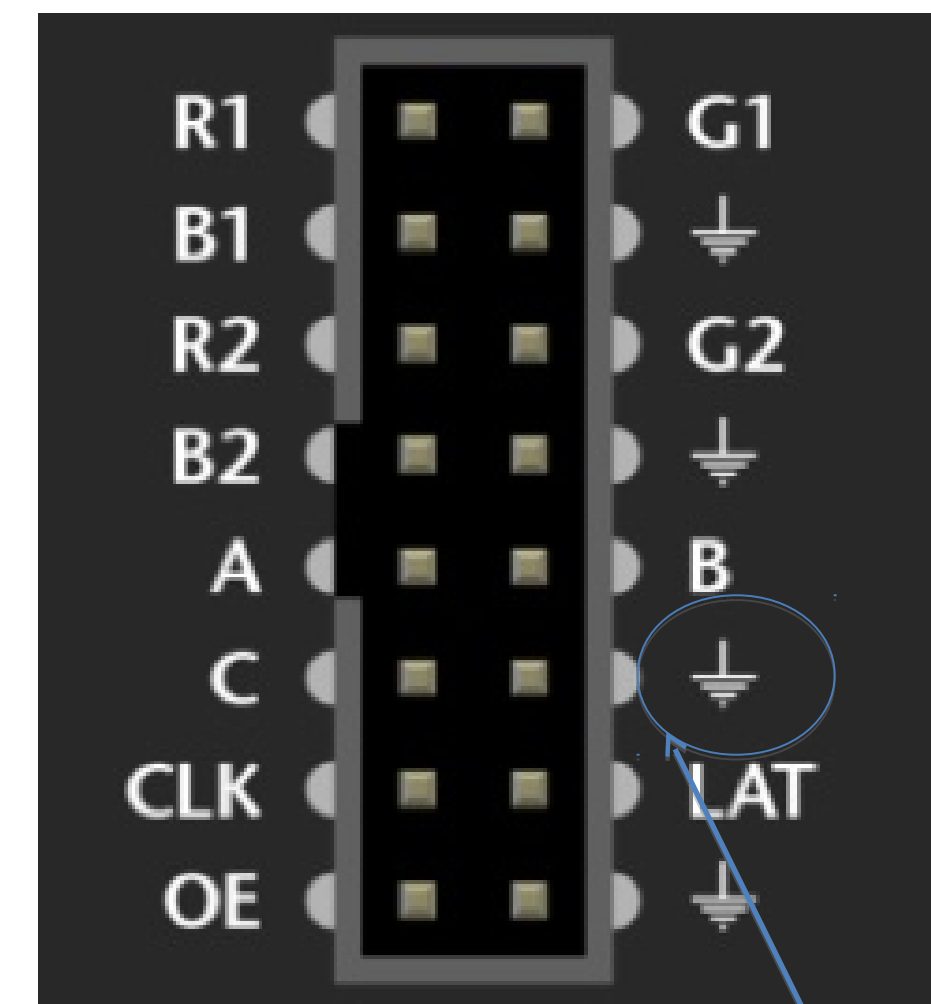


Contrôle de la Matrice à LED RGB 16X32 de Adafruit



Spécifications des pins IDC

- Les pins **R1**, **G1** and **B1** délivrent les données de couleurs pour la première moitié des leds (1-8x32).
- Les pins **R2**, **G2** and **B2** délivrent les données de couleurs pour la deuxième moitié des leds (9-16x32).
- Les pins **A**, **B**, **C** permettent de sélectionner les 8 sections de leds à commander. Pour un 16X32 **D** doit être à la masse.
- **LAT** (latch) signale la fin de la ligne des données,
- **CLK** (clock) rythme le contrôle de chaque bit de données.
- **OE** (output enable) permet d'activer l'affichage ou de verrouiller.



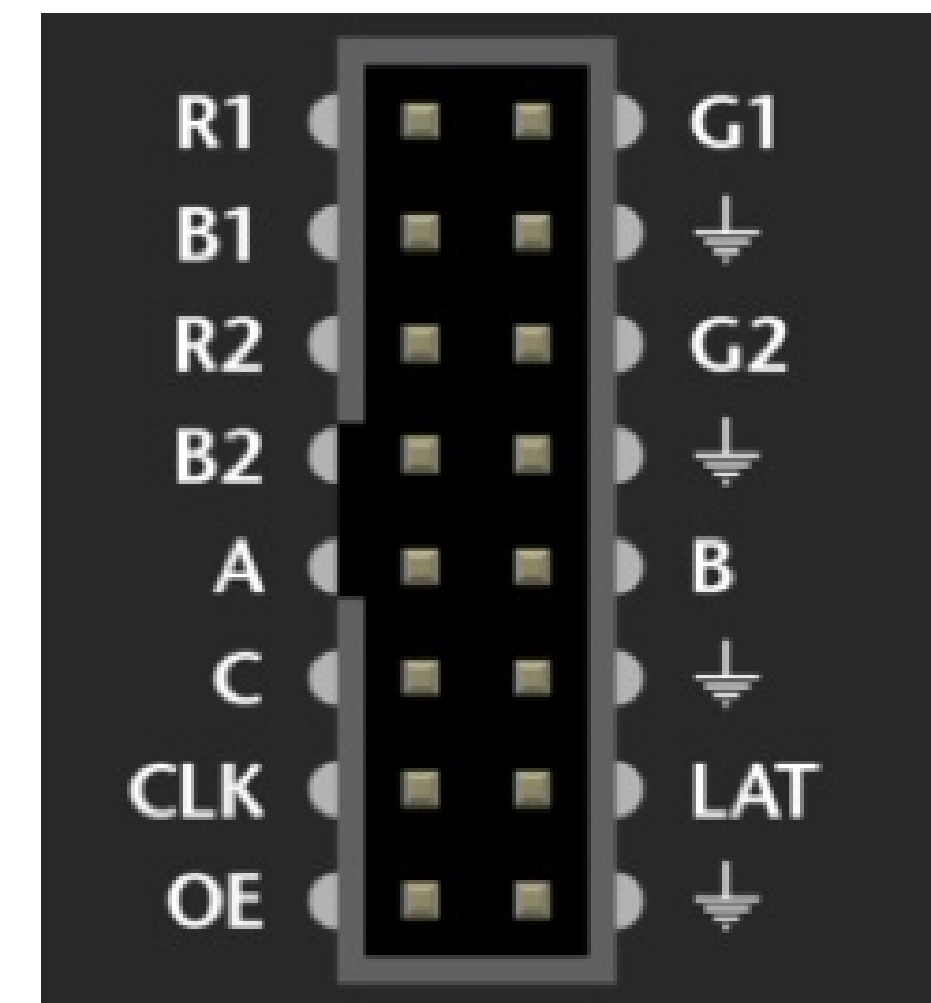
D pin ou
Masse

Contrôle de la Matrice à LED RGB 16X32 de Adafruit



Assignation des pins IDC pour test sur DE0 ALTERA

Matrix Pin	FPGA Pin	Pin Name
R1	PIN_E7	GPIO 18
G1	PIN_E10	GPIO 27
B1	PIN_E8	GPIO 20
R2	PIN_F9	GPIO 22
B2	PIN_C9	GPIO 24
G2	PIN_B11	GPIO 29
A	PIN_E11	GPIO 26
B	PIN_D11	GPIO 31
C	PIN_C11	GPIO 28
CLK	PIN_A12	GPIO 30
OE	PIN_D12	GPIO 32
LAT	PIN_B12	GPIO 33

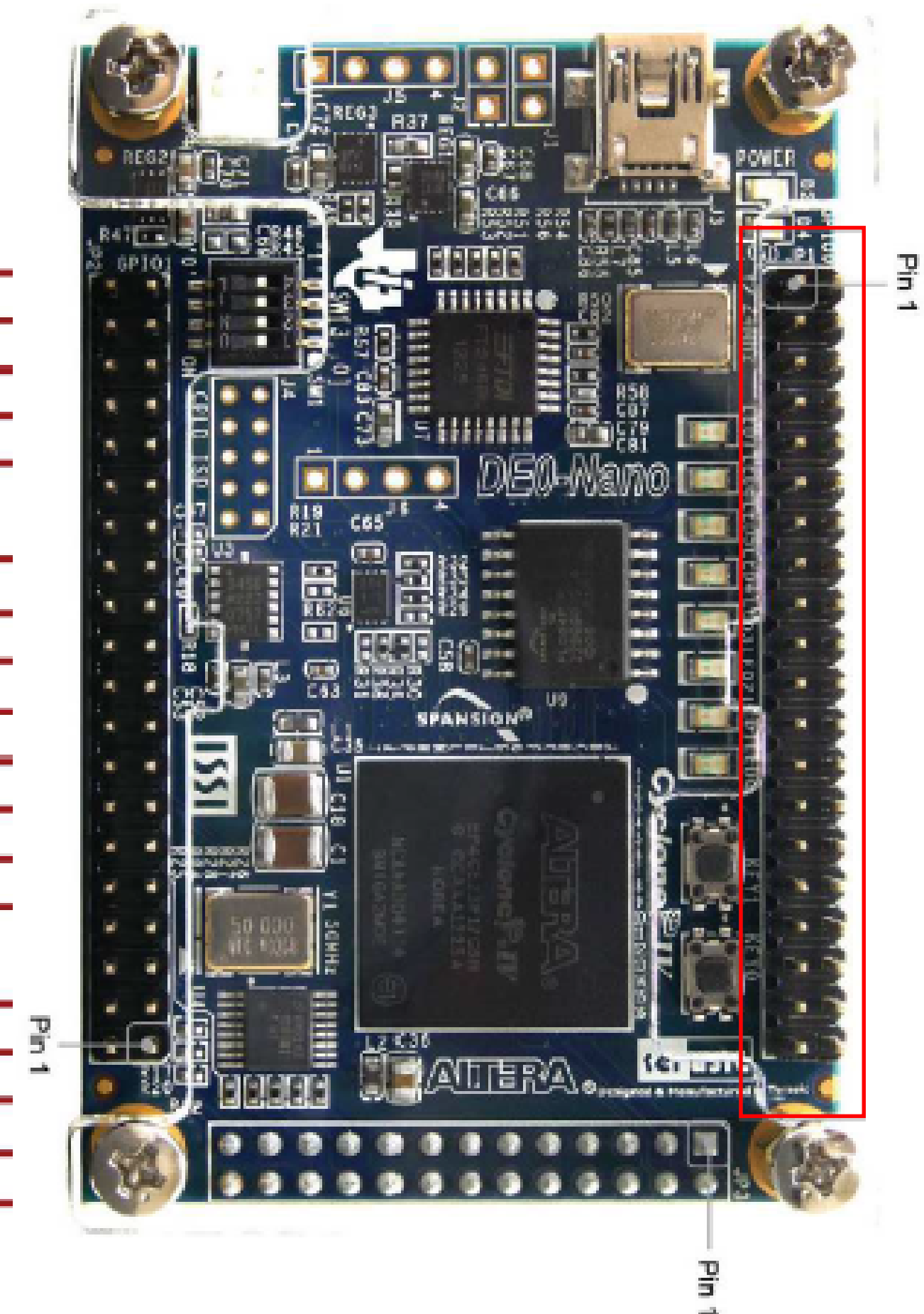
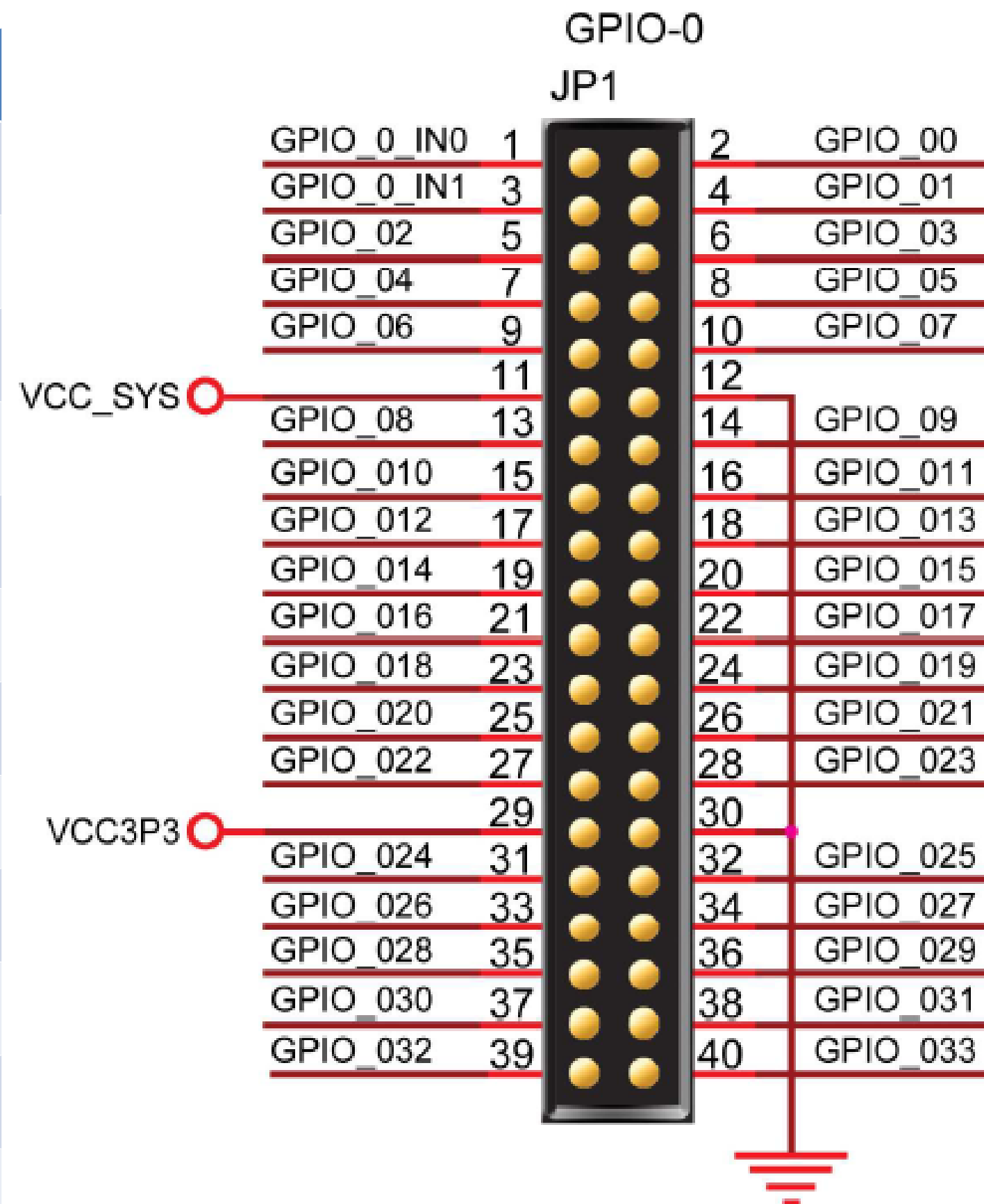


Contrôle de la Matrice à LED RGB 16X32 de Adafruit



Assignation des pins IDC pour test sur DE0 ALTERA

Matrix Pin	FPGA Pin	Pin Name
R1	PIN_E7	GPIO 18
G1	PIN_E10	GPIO 27
B1	PIN_E8	GPIO 20
R2	PIN_F9	GPIO 22
B2	PIN_C9	GPIO 24
G2	PIN_B11	GPIO 29
A	PIN_E11	GPIO 26
B	PIN_D11	GPIO 31
C	PIN_C11	GPIO 28
CLK	PIN_A12	GPIO 30
OE	PIN_D12	GPIO 32
LAT	PIN_B12	GPIO 33



Premier test de la carte Adafruit



Activer l'afficheur pendant 10ms toutes les 0,5 s.

- OE (output enable) permet d'activer l'affichage de la matrice.
- Génération d'un signal de période 500ms
- Durée niveau bas : 10 ms (activation)
- Durée niveau haut : 490 ms (verrouillage)

Deuxième test de la carte Adafruit

Activer l'afficheur pendant 10ms toutes les 0,5s.

- OE (output enable) permet d'activer l'affichage de la matrice.
- Génération d'un signal de période 500ms
- Durée niveau bas : 10 ms (activation)
- Durée niveau haut : 490 ms (verrouillage)

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use ieee.numeric_std.all;
```

```
entity test0 is  
    port (  
        clk_in  : in  std_logic;  
        oe      : out std_logic  
    );  
End test0 ;
```

```
architecture Behavioral of test0 is
```

```
begin  
    process (clk_in)  
        variable cpt : integer := 0;  
    begin  
        if (cpt < 1000000) then  
            oe <= '0';  
        elsif (cpt < 50000000) then  
            oe <= '1';  
        else  
            cpt := 0;  
        end if;  
        cpt := cpt + 1;  
    end process;  
end Behavioral;
```

Deuxième test de la carte Adafruit

Activer l'afficheur pendant 10ms toutes les 0,5s.

```
2 use IEEE.STD_LOGIC_1164.ALL;
3 use ieee.numeric_std.all;
4
5 entity test1 is
6     port (
7         clk_in    : in  std_logic;
8         oe        : out std_logic
9     );
10 end test1;
11
12 architecture Behavioral of test1 is
13 begin
14     process (clk_in)
15         variable cpt : integer := 0;
16     begin
17         if (cpt < 10) then    -- cpt < 1000000
18             oe <= '0';
19         elsif (cpt < 500) then -- cpt < 50000000
20             oe <= '1';
21         else
22             cpt := 0;
23         end if;
24         cpt := cpt + 1;
25     end process;
26 end Behavioral;
```

- Paramètres de simulation
 - Durée 0 à 100 us
 - Période clk_in 20 ns
 - Multiplication par 100000 pour les tests

Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA

The screenshot displays the Quartus II 32-bit software interface. The main window shows the 'Run Simulation Tool' menu open, with 'RTL Simulation' selected. The 'Messages' window at the bottom shows the following log:

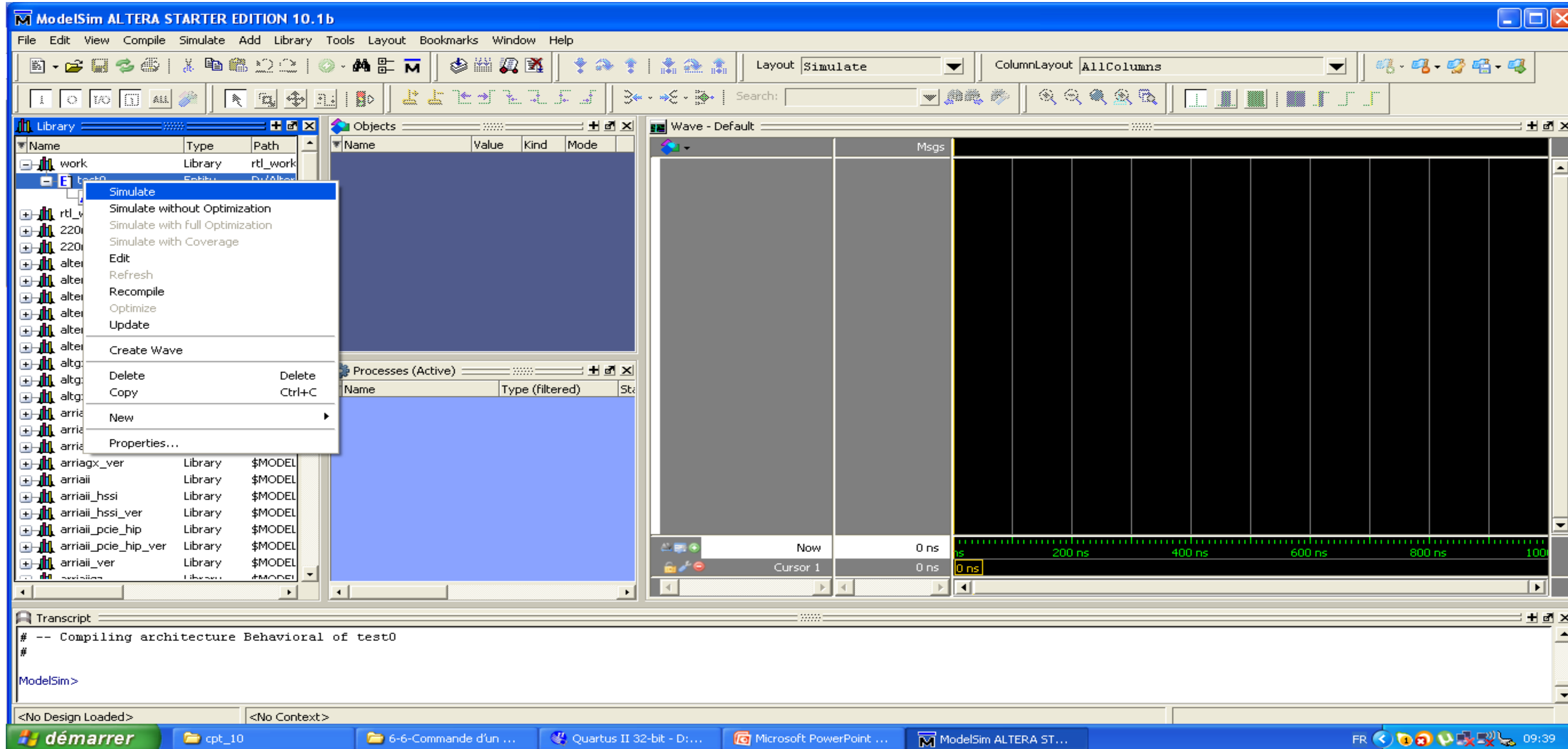
```
Info (204019): Generated file test0_b_1200mv_0c_vhd_slow.sdo in folder "D:/Altera_Project/energia/test0/simulation/modelsim/" for EDA simulation tool
Info (204019): Generated file test0_min_1200mv_0c_vhd_fast.sdo in folder "D:/Altera_Project/energia/test0/simulation/modelsim/" for EDA simulation tool
Info (204019): Generated file test0_vhd.sdo in folder "D:/Altera_Project/energia/test0/simulation/modelsim/" for EDA simulation tool
Info: Quartus II 32-bit EDA Netlist Writer was successful. 0 errors, 0 warnings
Info (293026): Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info (293000): Quartus II Full Compilation was successful. 0 errors, 30 warnings
```

The 'Messages' window also shows a list of tasks: Compile Design, Analysis & Synthesis, Edit Settings, View Report, Analysis & Elaboration, and Partition Merge. The 'Messages' window is currently displaying the 'Info' messages.

Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA



Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA

ModelSim ALTERA STARTER EDITION 10.1b

File Edit View Compile Simulate Add Library Tools Layout Bookmarks Window Help

ColumnLayout AllColumns

Search:

Library

Name	Type	Path
work	Library	rtl_work
test0	Entity	D:/Alter...
rtl_w		
220n		
220n		
alter		
alter		
alter		
alter		
alter		
alter		
alter		
altgx		
altgx		
altgx		
arria		
arria		
arria		
arriagx_ver	Library	\$MODEL
arriaii	Library	\$MODEL
arriaii_hssi	Library	\$MODEL
arriaii_hssi_ver	Library	\$MODEL

Objects

Name	Value	Kind	Mode
clk_in	U	Signal	In
rst	U	Signal	In
clk_out	U	Signal	Out
rgb1	UUU	Signal	Out
rgb2	UUU	Signal	Out
led_addr	UUU	Signal	Out
lat	U	Signal	Out
oe	U	Signal	Out

Processes (Active)

Name	Type (filtered)	Sta
line__21	VHDL Process	Act

Wave - Default

Now 0 ns

Cursor 1 0 ns

0.00 ns 0.2 ns 0.4 ns 0.6 ns 0.8 ns 1 ns

Transcript

```
# Loading ieee.numeric_std(body)
# Loading work.test0(behavioral)
VSIM 3>
```

Now: 0 ps Delta: 0 sim:/test0

Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA

ModelSim ALTERA STARTER EDITION 10.1b

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

ColumnLayout AllColumns

Search:

Library

Name	Type	Path
work	Library	rtl_work
test0	Entity	D:/Alter
behavioral	Architecture	
rtl_work	Library	D:/Alter
220model	Library	\$MODEL
220model_ver	Library	\$MODEL
altera	Library	\$MODEL
altera_Insim	Library	\$MODEL
altera_Insim_ver	Library	\$MODEL
altera_mf	Library	\$MODEL
altera_mf_ver	Library	\$MODEL
altera_ver	Library	\$MODEL
altgxb	Library	\$MODEL
altgxb_lib	Library	\$MODEL
altgxb_ver	Library	\$MODEL
arriagx	Library	\$MODEL
arriagx_hssi	Library	\$MODEL
arriagx_hssi_ver	Library	\$MODEL
arriagx_ver	Library	\$MODEL
arriaii	Library	\$MODEL
arriaii_hssi	Library	\$MODEL
arriaii_hssi_ver	Library	\$MODEL

Objects

Name	Value	Kind	Mode
clk_in	U	Signal	In
rst	U	Signal	In
clk_out	U	Signal	Out
rgb1	UUU	Signal	Out
rgb2	UUU	Signal	Out
led_addr	UUU	Signal	Out
lat	U	Signal	Out
oe	U	Signal	Out

Processes (Active)

Name	Type (filtered)	Sta
line__21	VHDL Process	Act

Wave - Default

Msgs

Context Menu:

- Edit
- View
- Radix
- Format
- Combine Signals...
- Group...
- Ungroup
- Force...
- NoForce
- Clock...
- Properties...

Transcript

```
wave create -pattern none -portmode out -language vhdl /test0/oe
# test0
VSIM 11>
```

Now: 0 ps Delta: 0 sim:/test0

Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA

The screenshot displays the ModelSim ALTERA Starter Edition 10.1b interface. The main window shows a simulation setup for a testbench. The 'Create Pattern Wizard' dialog is open, allowing the user to generate a waveform for a signal. The dialog is configured to create a clock pattern for the signal 'Edit:/test0/clk_in'. The 'Patterns' list includes Clock, Constant, Random, Repeater, and Counter. The 'Signal Name' field is set to 'Edit:/test0/clk_in'. The 'Start Time' is 0 and the 'End Time' is 100, with a 'Time Unit' of 'us'. The 'Wave - Default' window shows a waveform for the 'clk_in' signal, which is a clock signal starting at 0 ns and ending at 100 ns. The 'Transcript' window shows the command 'wave create -pattern none -portmode out -language vhdl /test0/oe' and the prompt 'VSIM 11>'. The status bar at the bottom shows 'Now: 0 ps Delta: 0' and 'sim:/test0'.

Premier test de la carte Adafruit

Simulation avec ModelSim ALTERA

The screenshot shows the ModelSim ALTERA Starter Edition 10.1b interface. The main window displays a simulation setup for a testbench named 'test0'. A dialog box titled 'Edit:/test0/clk_in <Pattern : clock>' is open, allowing the user to specify the clock pattern attributes. The dialog box contains the following fields:

- Initial Value: 0
- Clock Period: 20 ns
- Time Unit: ns
- Duty Cycle: 50

The dialog box also includes buttons for '< Previous', 'Finish', and 'Cancel'. The background shows the 'Objects' window with a list of signals and their values, and the 'Wave - Default' window showing a timing diagram for the signals.

The transcript window at the bottom shows the following commands:

```
wave create -pattern none -portmode out -language vhdl /test0/oe
# test0
VSIM 11>
```


Premier test de la carte Adafruit

Simulation avec ModelSim ALTERA

The screenshot displays the ModelSim ALTERA Starter Edition 10.1b interface. The 'Run' menu is open, showing options like 'Run 100', 'Run -All', 'Continue', and 'Run -Next'. The 'Wave - Default' window shows a timing diagram with signals: 'Edit:/test0/clk_in' (0), 'sim:/test0/clk_out' (U), 'sim:/test0/rgb1' (UUU), and 'sim:/test0/oe' (U). The 'Processes (Active)' window shows 'line__21' as a VHDL Process. The 'Transcript' window shows the following commands:

```
add wave -position end sim:/test0/rgb1
add wave -position end sim:/test0/lat
add wave -position end sim:/test0/oe

VSIM 12>
```

The status bar at the bottom indicates 'Now: 0 ps Delta: 0' and 'rst'. The Windows taskbar at the bottom shows the system tray with the time '09:44'.

Premier test de la carte Adafruit



Simulation avec ModelSim ALTERA

The screenshot displays the ModelSim ALTERA Starter Edition 10.1b interface. The 'Simulate' menu is open, showing options like 'Run', 'Step', and 'Restart...'. The 'Run' submenu is also visible, with 'Run -All' selected. The 'Wave - Default' window shows a timing diagram with a cursor at 61.287 ns. The 'Transcript' window at the bottom shows the following commands:

```
add wave -position end sim:/test0/rgb1
add wave -position end sim:/test0/lat
add wave -position end sim:/test0/oe
VSIM 12>
```

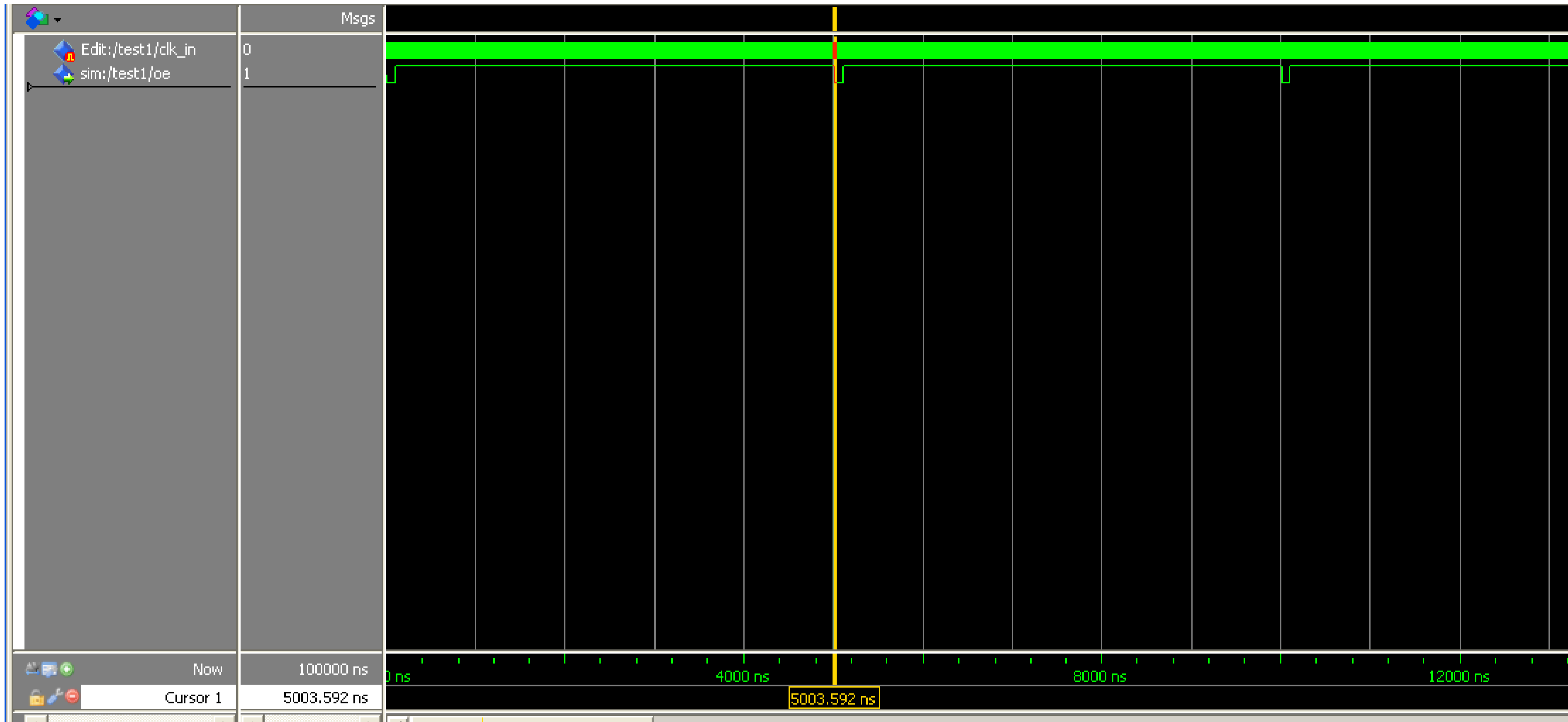
The signal list in the center shows the following signals:

Name	Kind	Mode
clk_in	U	Signal In
clk_out	U	Signal In
rgb1	UUU	Signal Out
rgb2	UUU	Signal Out
led_addr	UUU	Signal Out
lat	U	Signal Out
oe	U	Signal Out

Premier test de la carte Adafruit



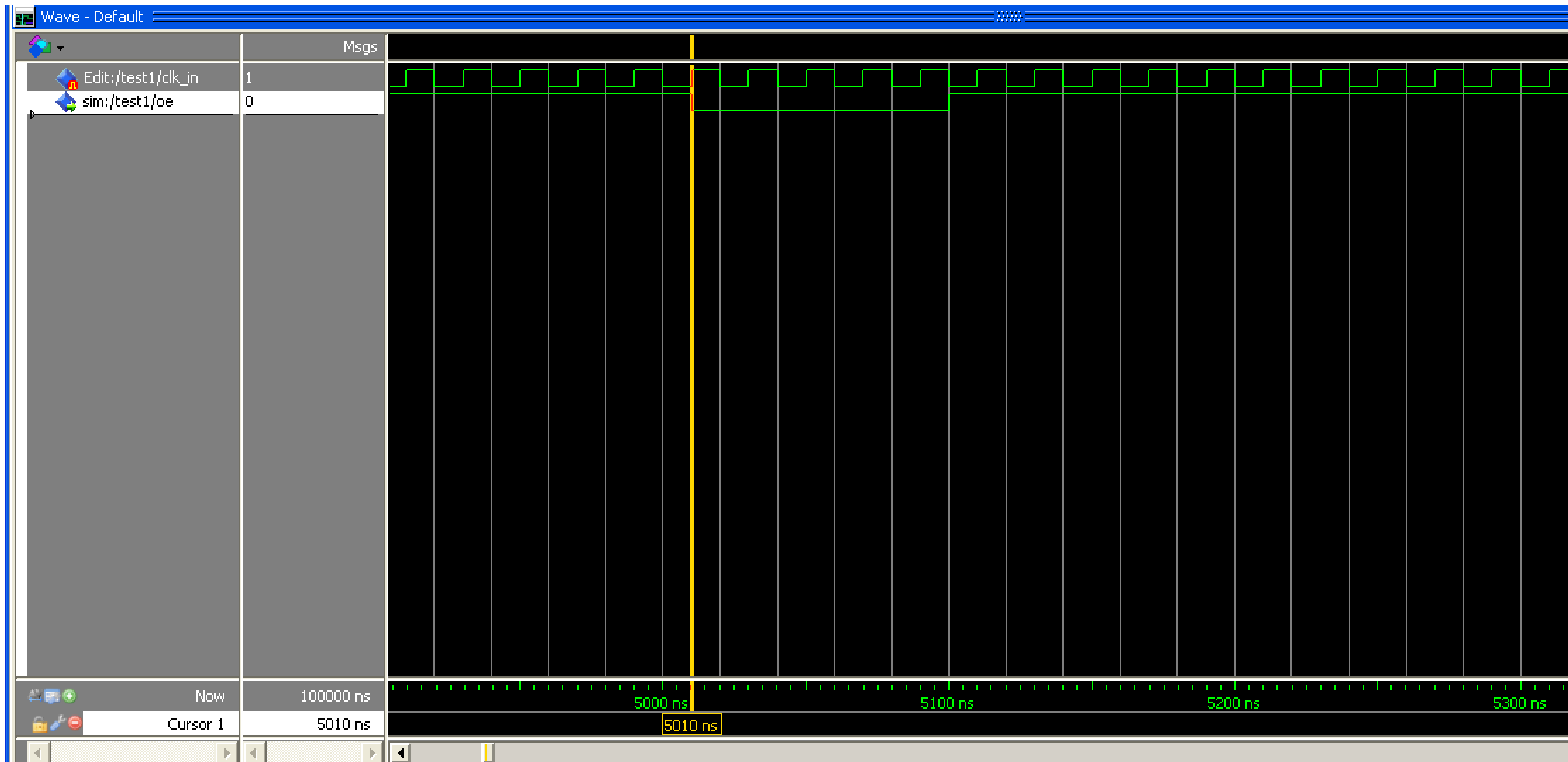
Activer l'afficheur pendant 10ms toutes les 0,5s.



Premier test de la carte Adafruit



Activer l'afficheur pendant 10ms toutes les 0,5s.



Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

- Les pins **R1**, **G1** and **B1** délivrent les données de couleurs.
 - LAT (latch) signale la fin de la ligne des données,
 - CLK (clock) rythme le chargement des données.
 - OE (output enable) permet d'activer l'affichage ou verrouiller l'affichage.
-
- Génération d'un signal de période 500ms
 - Durée niveau bas : 10 ms (activation)
 - Durée niveau haut : 490 ms (verrouillage)

Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use ieee.numeric_std.all;
4
5  entity test0 is
6  port (
7      clk_in    : in  std_logic;
8      -- LED Panel IO
9      clk_out   : out std_logic;
10     rgb1      : out std_logic_vector(2 downto 0);
11     lat       : out std_logic;
12     oe        : out std_logic
13 );
14 end test0;
15
16 architecture Behavioral of test0 is
17     signal s_rgb1 : std_logic_vector(2 downto 0) := (others => '0');
18     signal s_lat  : std_logic := '0';
19     signal s_oe   : std_logic := '1';
20     signal clk_div2 : std_logic := '0';
21     signal clk_div4 : std_logic := '0';
22     signal trigger : integer := 0;    -- pour déclencher la fin du chargement
23
```

Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

```
27  [-] begin
28  [-]   process (clk_div4)
29      variable cpt : integer := 0;
30      variable cpt1 : std_logic_vector (31 downto 0) := (others => '0');
31  begin
32      -- assignation
33      rgb1 <= s_rgb1;
34      lat <= s_lat;
35      oe <= s_oe;
36      -- chargement registre
37  [+   if (unsigned(cpt1) < 32) then
44  [+   else
67      cpt1 := std_logic_vector( unsigned(cpt1) + 1 );
68  end process;
69  -
70  -- gestion clk_out
71  [+   process (clk in)
80      -- Diviseur de frequence
81  [+   process (clk in)
92  end Behavioral;
```

Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

```
28 process (clk_div4)
29     variable cpt : integer := 0;
30     variable cpt1 : std_logic_vector (31 downto 0) := (others => '0');
31 begin
32     -- assignation
33     rgb1 <= s_rgb1;
34     lat <= s_lat;
35     oe <= s_oe;
36     -- chargement registre
37     if (unsigned(cpt1) < 32) then
38         if (cpt1(0) = '1') then
39             s_rgb1(2) <= '1';
40         else
41             s_rgb1(2) <= '0';
42         end if;
43         trigger <= 0;
44     else
67         cpt1 := std_logic_vector( unsigned(cpt1) + 1 );
68     end process;
```


Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

```
44  else
45      -- lat s'exécute une seule fois par cycle et lorsque le registre est plein
46      if (unsigned(cpt1) = 32) then
47          s_lat <= '1';
48      elsif (unsigned(cpt1) >= 33) then
49          s_lat <= '0';
50          -- oe pour Activer l'afficheur pendant 10ms toutes les demi-secondes
51          if (cpt < 10) then -- cpt < 1000000
52              s_oe <= '0';
53          elsif (cpt < 500) then -- cpt < 50000000
54              s_oe <= '1';
55          else
56              cpt := 0;
57              cpt1 := (others => '0');
58          end if;
59          cpt := cpt + 1;
60          end if;
61
62          -- trigger
63          if (unsigned(cpt1) > 32) then
64              trigger <= 1;
65          end if;
66      end if;
67      cpt1 := std_logic_vector( unsigned(cpt1) + 1 );
68  end process;
```

Deuxième test de la carte Adafruit

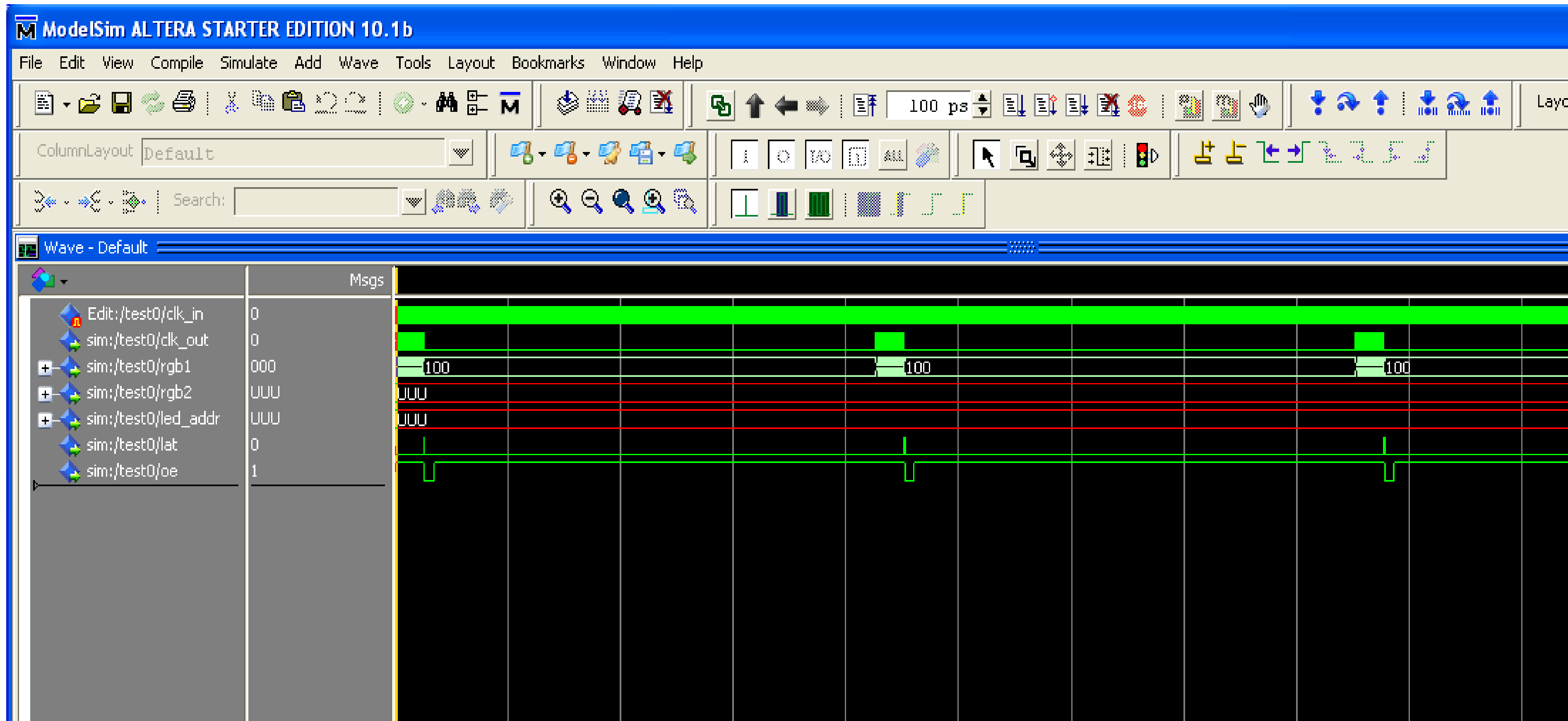


Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.

```
69 |
70 |
71 |  -- gestion clk_out
72 | | process (clk_in)
73 |  | begin
74 | |     if (trigger = 0) then
75 | |         clk_out <= clk_div2;
76 | |     else
77 | |         clk_out <= '0';
78 | |     end if;
79 | | end process;
80 | |
81 |  -- Diviseur de frequence
82 | | process (clk_in)
83 | |     variable cpt3 : integer := 0;
84 | | begin
85 | |     if (cpt3 mod 2 = 0) then
86 | |         clk_div2 <= not clk_div2;
87 | |     end if;
88 | |     if (cpt3 mod 4 = 0) then
89 | |         clk_div4 <= not clk_div4;
90 | |     end if;
91 | |     cpt3 := cpt3 + 1;
92 | | end process;
    | end Behavioral;
```

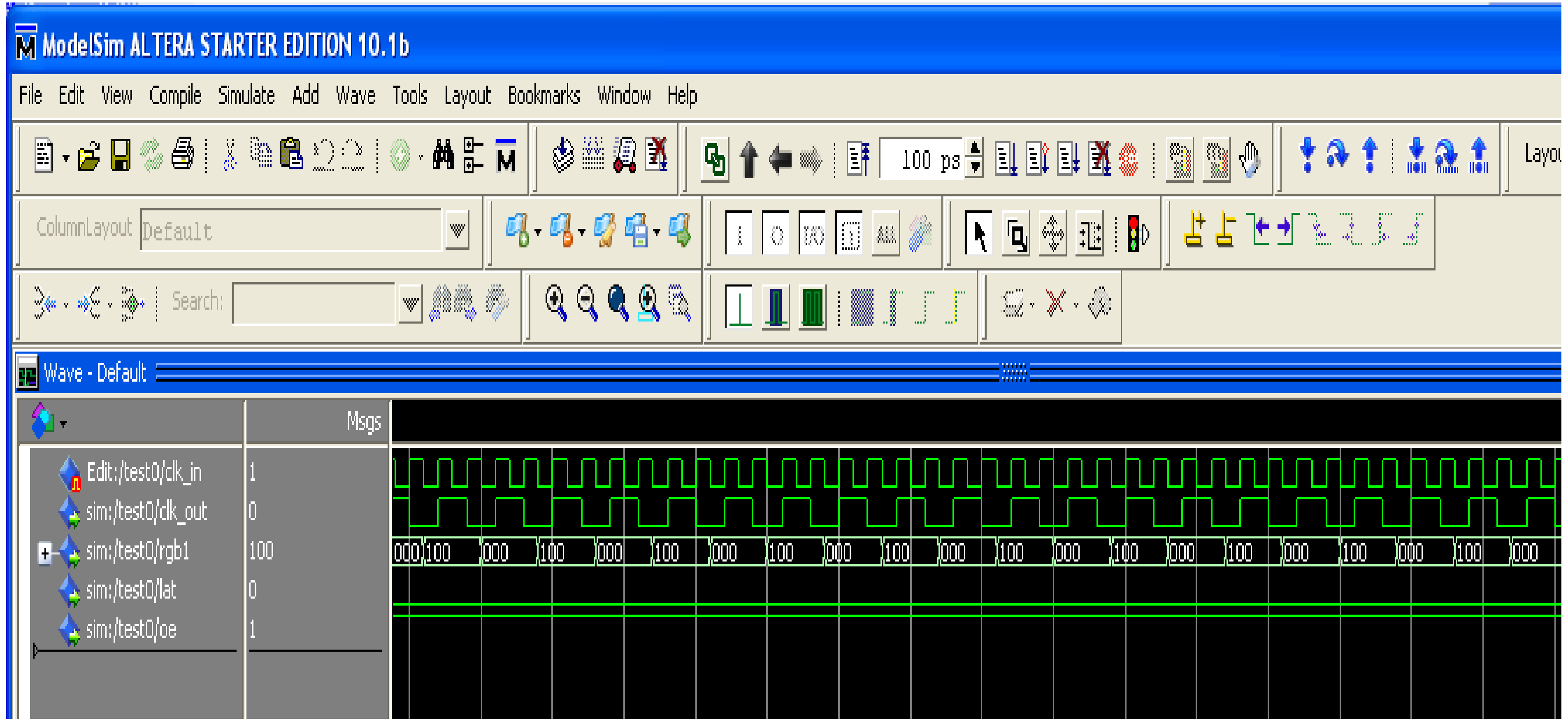
Deuxième test de la carte Adafruit

Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.



Deuxième test de la carte Adafruit

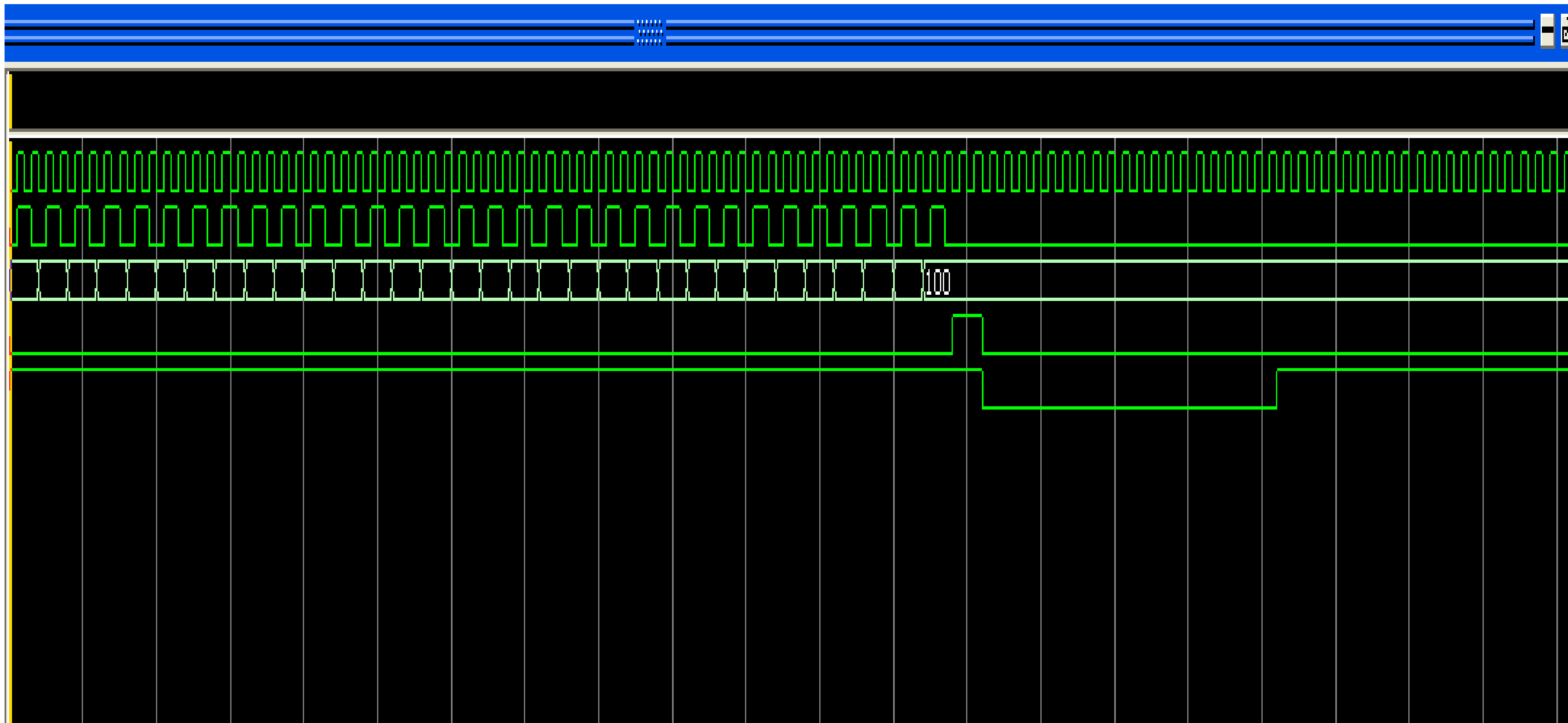
Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.



Deuxième test de la carte Adafruit



Activer l'afficheur 1 LED sur 2 pendant 10ms toutes les 0,5s.



- Présentation de la matrice à LED
- Contrôle par FPGA de la matrice à LED
- Simulation des signaux de contrôle
- Test sur la carte Adafruit