Advanced Logic Synthesis for Electronics
www.ALSE-FR.com

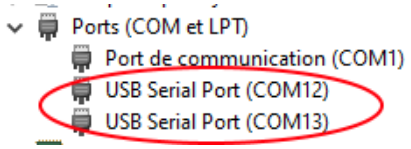
Application Note
Gowin TEC0117 Blinker

Introduction

If you just bought the TEC0117 Littlebee board, you can follow this small Application Note to go through the whole process of setting up the Gowin tools, creating a simple project and testing it on the FPGA board.
The design is a simple 27 bits binary counter (running at **100 MHz**) driving LEDs.

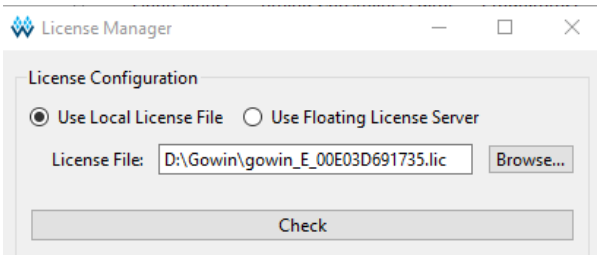
Install "Gowin EDA" (IDE)

- Download the **TEC0117 Schematics** from the Trenz website (to view the pins assignment).
- Make sure you have admin rights on your PC !
- **Register**, then download the latest Gowin Software (for your OS) from the GowinSemi Website.
  (do **not** select the Education version !).
- Unpack the installation zip file and **run the executable**.
  At the end, choose to install the FTDI drivers for Windows7/8/10/11 (not Windows XP).
  A Gowin icon should have been placed on your desktop.
- Verify that the board is recognized under Windows : plug the TEC0117 in a USB port.
  Open Windows Device Manager and check that you have now two new Serial Ports.
  If not, try to (re-)install the FTDI drivers.



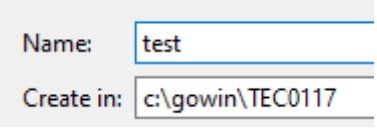
  ☞ If you have other FTDI peripherals (like USB-RS232 adapters), unplug them.

- Check your HostId : `ipconfig /all` then note the Physical Address of your main Ethernet Adapter.
  This will look like this : "00-E0-3D-69-17-35".
- Request a License ("Apply License" page). Use the above information as "PC MAC Address".
  Type of license = **Local**. Software = **GOWIN EDA**.
  After a few minutes you will receive an email with 4 attachments. Ignore the images and copy the license file named gowin**_E_**xxxxxxxx.lic under the Gowin installation directory.
  The other licence file (no _E_) is for older versions.
- Launch Gowin. It will fail in the absence of selected license file, and it will open the License Manager.
  Browse to select your license file (the one with _E_, copied under the install directory), then **Check** that it is correctly recognized.



- Try again to launch Gowin EDA : it should open and be operational. You're good to go !

Create your first design



- Launch **Gowin EDA**. Quick Start : **New Project**.
- Name = **test**, Create In = **c:\gowin\TEC0117**
  Select Device = **GW1NR-LV9QN88**C6 - Finish
- Right-click in the Design pane : **New File** → **Verilog File**. Name = **blink** (Add to current project)
  Enter the following SystemVerilog code (cut & paste) then save it (Ctrl-S) :

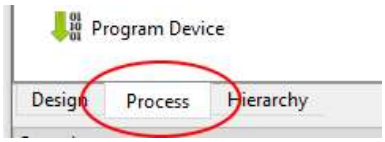
```
module blink (input logic CLKX, output logic [1:2] LED);
logic [26:0] Cntr;
always_ff @(posedge CLKX) Cntr <= Cntr + 1'b1;
assign LED = Cntr[26:25];
endmodule
```

- Right-click in the Design pane : **New File** → **Physical Constraints File**. Name = **blink**
  Enter the following Constraints (cut & paste) then save the file (Ctrl-S) :

```
IO_LOC "CLKX" 63;
IO_PORT "CLKX" PULL_MODE=UP;
IO_LOC  "LED[1]" 86;
IO_LOC  "LED[2]" 85;
IO_PORT "LED[1]" PULL_MODE=UP DRIVE=8;
IO_PORT "LED[2]" PULL_MODE=UP DRIVE=8;
```

- Right-click in the Design pane : **New File** → **Timing constraints File**. Name = **blink**
  Enter the following SDC code (cut & paste) then save the file (Ctrl-S) :

```
create_clock -name CLK -period 10 [get_ports {CLKX}]
set_false_path -to [get_ports {LED*}]
```



- Activate the **Process** pane.
- Right-click on "**Synthesize**" then **Configuration**.
  Top Module = **blink**
  Select Verilog language = "**System Verilog 2017**", then **Apply**.
- Double Click on **Synthesize** : it should compile without error.
  Double Click on **Place & Route** : it should compile without error.
- Make sure the TEC0117 is plugged.
  Double Click on **Program device** : the "cable" (the TEC0117 board) should be detected.
- The programming panel should look like this :

- Click on "**Program/Configure**".
  After a few seconds, the two rightmost LEDs closest to the switch should count (binary).
- You can inspect the various reports, view the schematics (RTL view), open the floorplanner...
- Assign the 8 LEDs : modify `output logic [1:8] LED` and : `LED = Cntr[26:19]`.
  You can use the Floorplanner and the Package view to drag and drop the extra 6 LED ports to their location (84,83,82,81,80,79).
  You could also try with the 12 MHz clock (pin 35) for a slower pace.
- You can program the internal Flash so the board will power up with the blinker.

Update (Feb 2024)

The latest version of Gowin EDA does now support VHDL.

So you can try with this code :

```
-- GoWin Blinker example in VHDL
-- -----------------------------
-- https://alse-fr.com

Library IEEE;
    use IEEE.std_logic_1164.all;
    use IEEE.numeric_std.all;
Entity blink is
  port (CLKX : in  std_logic;
        LED : out std_logic_vector(1 to 8) );
end entity blink;
Architecture RTL of blink is
  signal Cntr : unsigned (28 downto 0);
begin
  Cntr <= Cntr + 1 when rising_edge(CLKX);
  LED  <= std_logic_vector(Cntr(28 downto 21));
end architecture RTL;
```

Conclusion

I hope this small Application Note was useful to quickly become familiar with the Gowin environment.

It's an attractive tool : very compact and easy to install and use, very responsive and fast, with a good support for the modern FPGA Design Methodology (VHDL, SystemVerilog, SDC …).

How extensive the support of VHDL is, remains to be verified. A quick test showed no VHDL2008 setting, and the float_pkg package was not supported. But for a lot of existing VHDL code, this is not a major issue.

With the new Arora V family, Gowin has an attractive offer for quite a range of applications.

However, my experience to locate and buy the chips or Gowin kits in Europe was much more painful than for the other vendors which are available from all the usual suspects (Mouser Digikey RS etc). This is not the case for Gowin and if you are looking for a few parts or for a kit you'll need to spend some time.

Bert Cuzeau – A.L.S.E – FPGA.fr