DE10-Nano for Laboratorio di interazioni fondamentali

Federico Lazzari

February 11, 2025

This document is a guide to the DE10-Nano board. Students are recommend to read Sections 1 and 2. Sections 3 and 4 are for special use cases only.

1 Connect to the board

Connect a PC to the USB port of the NIM module. A new network interface will be added to the PC. The PC will get the IP address `192.168.7.5` and the DE10-Nano OS will be reachable at the address `192.168.7.1`. If the board is connected to a LAN is also possible to reach it via the IP address provided by the LAN DHCP server.

HDMI output is not available with firmware for data acquisition. For debugging porpoise, a UART terminal is available connecting a PC to the USB mini-B port near the ethernet port.

More information are available in the DE10-Nano Getting Started Guide [1].

SSH

Access to the DE10-Nano OS through Secure SHell (SSH) with the user *root*:

```
ssh root@192.168.7.1
```

There is not a password, if it is required leave the field blank.

VNC

On DE10-Nano runs a Virtual Network Computing (VNC) server. Connect to VNC server using your preterite VNC client. No password/encryption is required.

UART

Open a serial connection with baud rate 115200:

```
screen /dev/ttyUSB0 115200
```

1.1 Update system time

The DE10-Nano do not have a real-time clock (RTC), therefore the system clock stops every time the board is not powered. To get meaningful timestamp is recommended to sync the DE10-Nano OS time to the PC system time:

```
ssh root@192.168.7.1 date -s @$(date -u +"%s")
```

2 Acquire data

The FPGA firmware provided two function: it records a timestamp every time a signal to one of the GPIO input ports rises (digital), records the amplitude of an analog signal when a digital trigger signal is provided (analog). Both functions are enabled by default.

2.1 Digital function

The FPGA has a register that count the number of clock cycle since the FPGA reset is deasserted (timestamp). When a signal to one of the 12 GPIO input ports rises, the timestamp is recorded. This operation has a fixed latency of 5 clock cycle (25 ns). Moreover the same signal provided on a second port should be recorded with a different timestamp (±1 clock cycle) due to fluctuation during NIM-TTL conversion and transition between logic states.

The clock freqeuncy is 200 MHz, therefore the counter is increased by one every 5 ns. The counter has 30 bits. Every 2^{30} clock cycles (5.36870912 s) it is reverted to 0. At every counter reset a special word is recorded. This word contains the reset count, starting from 0.

The FPGA might not be able to record signals shorter than the clock period (5 ns) and to distinguish two signals on the same line separated less than a clock period (5 ns). It is recommend to shape signals in order to maintain high logic level ($V_s > 2.4$ V) for more than 5 ns, and return to low logic level ($V_s < 0.4$ V) for at least 5 ns before sending a second signal. Preliminary tests show no errors if the LVTTL signal remains over 2 V at least 4 ns, and if between two consecutive signals the voltage remains under 0.8 V for at least 3 ns (6 ns under 2 V).

The FPGA accepts an aggregated input rate $\mathcal{O}(10$ kHz). Hitting the input rate limits leads to data loss and warning messages during data acquisition.

The digital FPGA function is always enabled during data acquisition. Running the DAQ software with the option `-d`, only this function is enabled (i.e. it disables the analog function).

2.2 Analog function

The board con acquire analog signals with amplitude in the range $[0, 4.095]$ V with 1 mV of resolution. When a signal on `GPIO_0 IN 11` rises (trigger signal), the Analog to Digital Converter (ADC) of the board start the conversion of signal at `ADC 0` port. Then the digitalized value is recorded. For each trigger signal 8 conversion are performed in sequence.

The acquisition require a fixed amount of time:

- The timestamp of the trigger signal is recorded with 25 ns of delay.
- After additional $10 - 30$ ns the ADC start the conversion.
- The digitalized value is recorded 2135 ns after the start of the conversion.
- The next conversion begin after 65 ns (i.e. 2200 ns after the previous one).

The FPGA accepts a trigger rate $\mathcal{O}(1$ kHz).

The analog function can be disabled running the DAQ software with the option `-d`.

2.3 Commands

The software for data acquisition is already available in the root home folder of the DE10-Nano OS (`/home/root`). For any need it is also available, together to a example program to decode the output file, in the git repository https://baltig.infn.it/lazzari/de10-nano-software.

Connect LVTTL signals (max 3.3 V, more details in Appendix A) to the GPIO ports of the NIM module. `GPIO_0 IN` and `ADC` ports are terminated at 50 Ohm.

Connect to the DE10-Nano OS and start a screen section:

```
ssh root@192.168.7.1
screen
```

The screen section is not mandatory, however without it the DAQ software will be terminated if the PC is disconnected (see Appendix B).

Start DAQ software:

```
./FIFOread
```

The DAQ software writes a file with name `FIFOread_YYYYMMDD-HHMMSS.txt` in the actual folder, where `YYYYMMDD-HHMMSS` is the start time of data acquisition. During data acquisition it does not print timestamps on screen. It is possible to check data acquisition in a second screen windows following the output file:

```
tail -f `ls -rt FIFOread_* | tail -n 1`
```

however it is updated in block of 4096 bytes (∼ 342 lines). The DAQ software compiled with VERBOSE option is available under the name FIFOread_dbg. This special flavor prints timestamps on screen in addition to the usual output file. Accepted input rate might be lower. A preliminary test did not report significant reductions, however it is recommend to do not use this version for long unmonitored data acquisition.

Stop DAQ software pressing ctrl+c.

Copy the output file to PC:

```
scp root@192.168.7.1:FIFOread_YYYYMMDD-HHMMSS.txt ./
```

During data acquisition is possible to detach the screen session pressing ctrl+a then d. To retrieve a detached session use the command:

```
screen -r
```

2.4 Output file format

The output file can be used as it is or converted in a more human readable file using the macros available in the git repository https://baltig.infn.it/lazzari/de10-nano-software (not still documented). It has a line for each clock cycle with at least one rising input line. Each line has two words: the first one contains the list of rising input lines, the second one contains the clock counter value. Additional lines, with the same structure are added at every clock counter reset. In these cases the second word contains the reset counter value. The words formats are shown in Figure 1 and Figure 2.

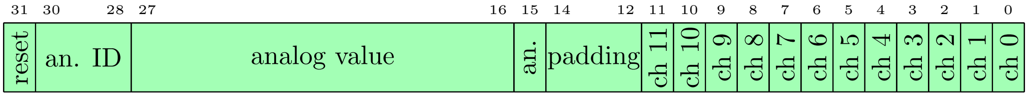

Figure 1: Channel word format.

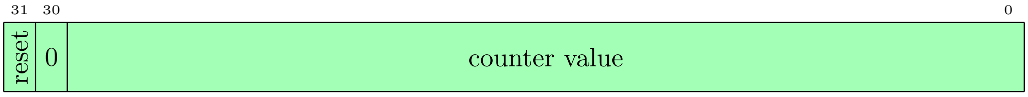

Figure 2: Counter word format.

If the reset flag of the two word are not asserted in the same line, an error occurred during data acquisition.

The channel word carries also information about the analog readings. Bit 15 specifies if an analog reading is stored in the word. Bits $16-27$ contain the digitalized value of the analog signal. Bit $28-30$ report which one of the 8 reading triggered by the same signal is stored in this word.

3 Compile the firmware

The firmware source code is available at https://baltig.infn.it/lazzari/de10-nano-hardware. To compile it make sure that the software stack on the PC is configured as reported in the repository readme.

Clone the git project and run the first compilation:

```
git clone https://baltig.infn.it/lazzari/de10-nano-hardware.git
cd de10-nano-hardware
make de10-nano-base.all
```

Subsequent compilations can be performed directly from Quartus GUI opening the Quartus project file de10-nano-base/de10-nano-base.qpf.

To produce the Raw Binary File (rbf), execute:

```
quartus_cpf -c -o bitstream_compression=on \
  de10-nano-base/output_files/de10-nano-base.sof \
  de10-nano-base/output_files/de10-nano-base.rbf
```

3.1 Change FPGA firmware

FPGA firmware is stored in SD card FAT partition, with name `de10-nano.rbf`. It is loaded right before OS boot. To change the firmware, overwrite the old firmware with the new one and reboot the DE10-Nano OS:

```
scp de10-nano-base/output_files/de10-nano-base.rbf \
  root@192.168.7.1:/media/FAT/de10-nano.rbf
```

4 Prepare a board

A preconfigured DE10-Nano OS system is available in the Google Drive folder of the course (*dispositivi/DE10-Nano/*). It was obtained from the *DE10_Nano_Xfce* factory including some addition:

- **nano** simple text editor
- **screen** terminal multiplexer
- **de10nano_In14_Out0_A0.rbf** default DAQ firware
- **read** program for reading FPGA registers
- **write** program for writing FPGA registers
- **FIFOread** program for data acquisition
- **system updates** to 13/03/2024

Download and extract the OS image. Copy the image on the MircoSD card:

```
dd bs=4M if=de10-nano_lab_OS.img \
  of=/dev/disk/by-id/MY_SD_CARD \
  conv=fsync oflag=direct status=progress
```

Expand OS partition covering the whole free space of the drive with your favorite software. Insert the MircoSD card in the board socket.

Configure the board like for running *DE10_Nano_Xfce* factory image. The FPGA Configuration Mode Switch (MSEL) must be set to 01010 (MSEL[4:0] = 01010) as shown in Figure 3. More information are available in the DE10-Nano Getting Started Guide [1].

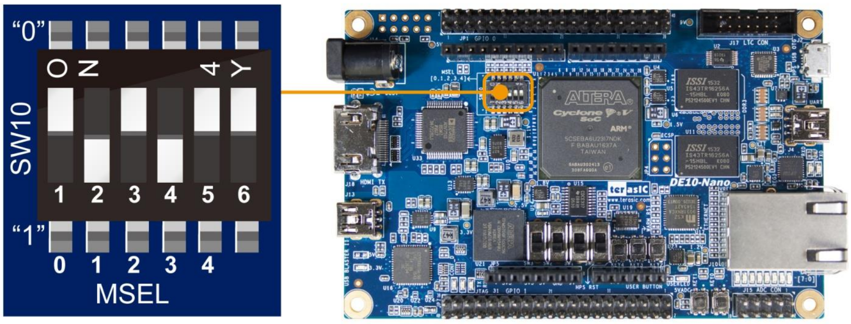

Figure 3: FPGA Configuration Mode Switch set in FPPx32 Mode.

4.1 Prepare the MircoSD from factory image

Download the *Linux Xfce Desktop* image from DE10-Nano resources webpage [2]. Extract and copy the image on the MircoSD card:

```
dd bs=4M if=de10-DE10_Nano_Xfce.img \
  of=/dev/disk/by-id/MY_SD_CARD \
  conv=fsync oflag=direct status=progress
```

Insert the MircoSD card in the board socket.

With the board connected to internet update the system and install the additional software:

```
opkg update
opkg upgrade
opkg install nano
opkg install screen
```

Copy the DAQ firmware and software.

A LVTTL voltage levels

Low Voltage Transistor-Transistor Logic (LVTTL) is an interface standard for digital integrated circuits [3]. The nominal power supply voltage V_{CC} is 3.3 V, with tolerance of ± 0.3 V for industrial grade device (normal range) and ± 0.15 V for commercial grade device (narrow range).

At low logic level, LVTTL drivers must provide a voltage lower than $V_{OL} = 0.4$ V. At high logic level, it must be greater than $V_{OH} = 2.4$ V.

LVTTL receivers must accept as low logic level signals with voltage in the range $[-0.3, 0.8]$ V, and signals with voltage in the range $[2.0, V_{CC} + 0.3]$ V as high logic level.

LVTTL drivers are compatible with TTL receivers, however TTL drivers are usually incompatible with LVTTL receivers. Figure 4 shows the voltage levels for TTL and LVTTL signals.

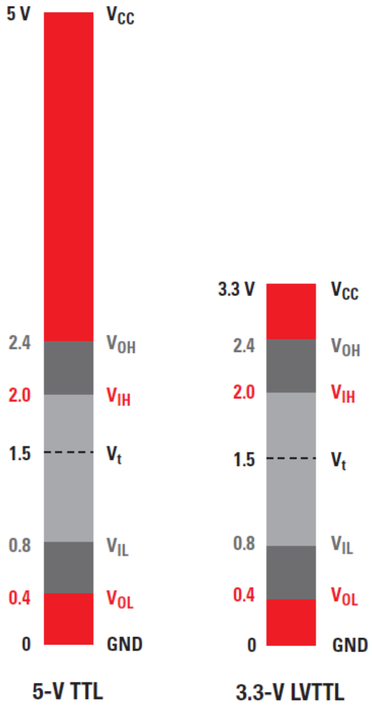

Figure 4: Voltage levels of TTL and LVTTL signals.

B Screen

GNU Screen is a terminal multiplexer [4]. It allows to open any number of windows (virtual terminals) inside a single screen session. Processes running in screen will continue to run when their window is not visible even if the user is disconnected. Keyboard inputs are usually sent to the program running in the current window, the `ctrl+a` keystroke initiate a command to the window manager. The most commond command are:

- `ctrl+a ?` Get a list of commands.
- `ctrl+a` \ Quit screen.
- `ctrl+a d` Detach screen session.

- `ctrl+a c` Create a new window (with shell).
- `ctrl+a 0` Switch to window 0 (by number).
- `ctrl+a Ctrl+a` Toggle between the current and previous windows.
- `ctrl+a "` List all windows.
- `ctrl+a S` Split current region horizontally into two regions.
- `ctrl+a |` Split current region vertically into two regions.
- `ctrl+a tab` Switch the input focus to the next region.
- `ctrl+a X` Close the current region.
- `ctrl+a Q` Close all regions but the current one.

References

[1] Terasic, "Getting Started Guide," 2020-10-30, https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=1046&FID=753abb03f323f8f2135130881425ee7b.

[2] ——, "DE10-Nano resources webpage," https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=1046&PartNo=4#contents.

[3] JEDEC, "Interface Standard for Nominal 3 V/3.3 V Supply Digital Integrated Circuits," *JESD8C.01*, September 2007, https://www.jedec.org/standards-documents/docs/jesd-8c01.

[4] F. S. Foundation, "GNU Screen," https://www.gnu.org/software/screen/.