Quick Installation Guide

Firmware:

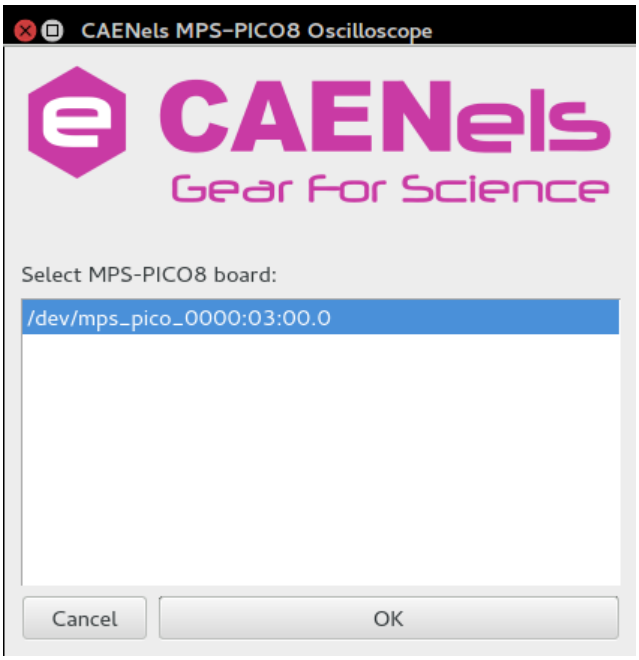
Install the firmware (“MPS_virtex_top.rle.hpm”) on the DAMC-FMC25 following the instructions in the README.md.
Power-cycle the DAMC-FMC25 at the end of the procedure.

Driver (mps-pico8-driver):

Follow the instructions in the README.md to build, install and load the driver. Once the driver is installed and the AMC board running with the correct firmware, it’s suggested to perform a reboot of the PC-AMC board where the driver has been installed.

Software (mps-pico8-oscilloscope):

Follow the instructions in the README.md.
The Oscilloscope is a PyQt5 application, it’s necessary to install the right dependencies.
When you run the application you should see this:



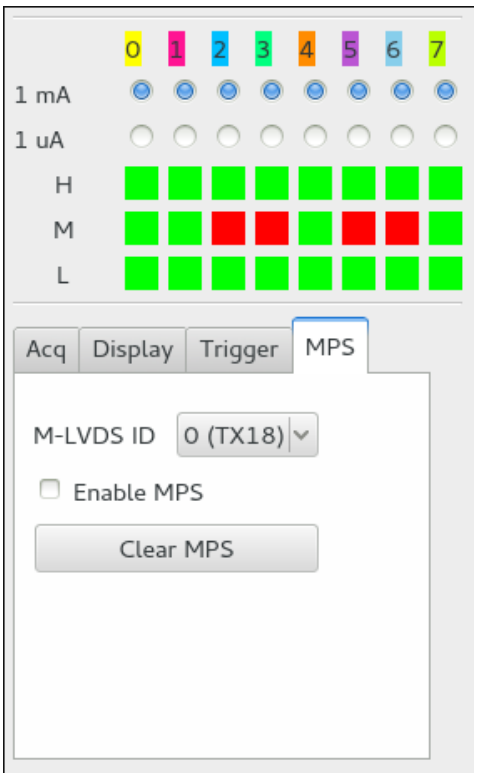
Select the card and press OK. The card number, “03” in this case, it’s related to the card position in the crate.
If the firmware and driver work correctly the oscilloscope should plot the current reading and FFT of the selected channel, as follows:

This is the main window of the oscilloscope. It is used to monitor all the main MPS quantities.

MPS Configuration Menu:

From this menu it is possible to configure all the thresholds and the number of samples on which the moving average is performed.
As an example, an MPS is generated when the sum of all the acquired samples in the moving window is grater than N_SAMP multiplied by the correspondent THRESHOLD:

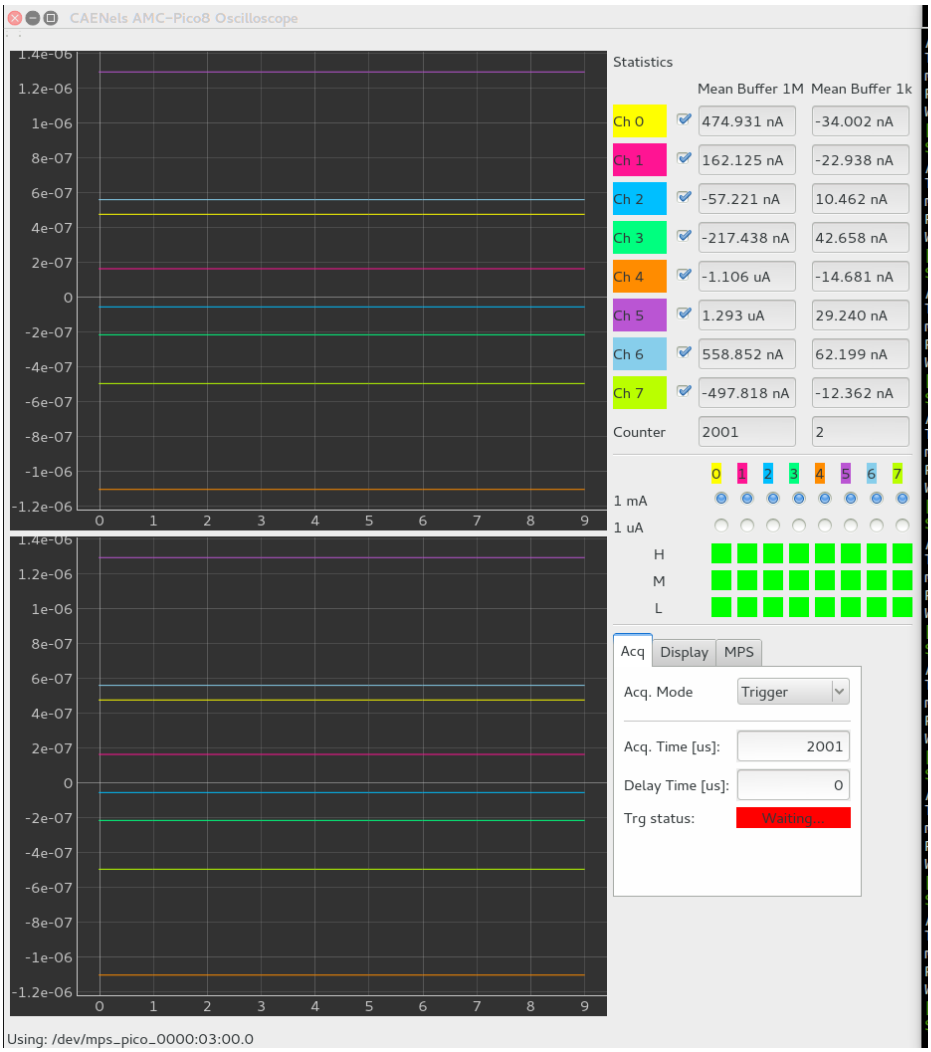
MPS tab (on the main window):



In the MPS tab it is possible to:

- Select the M-LVDS line on which the MPS will be asserted
- Enable/Disable the MPS generation (by disabling the correspondent M-LVDS line)
- Clear the MPS, by clearing all the accumulators of the moving average modules.

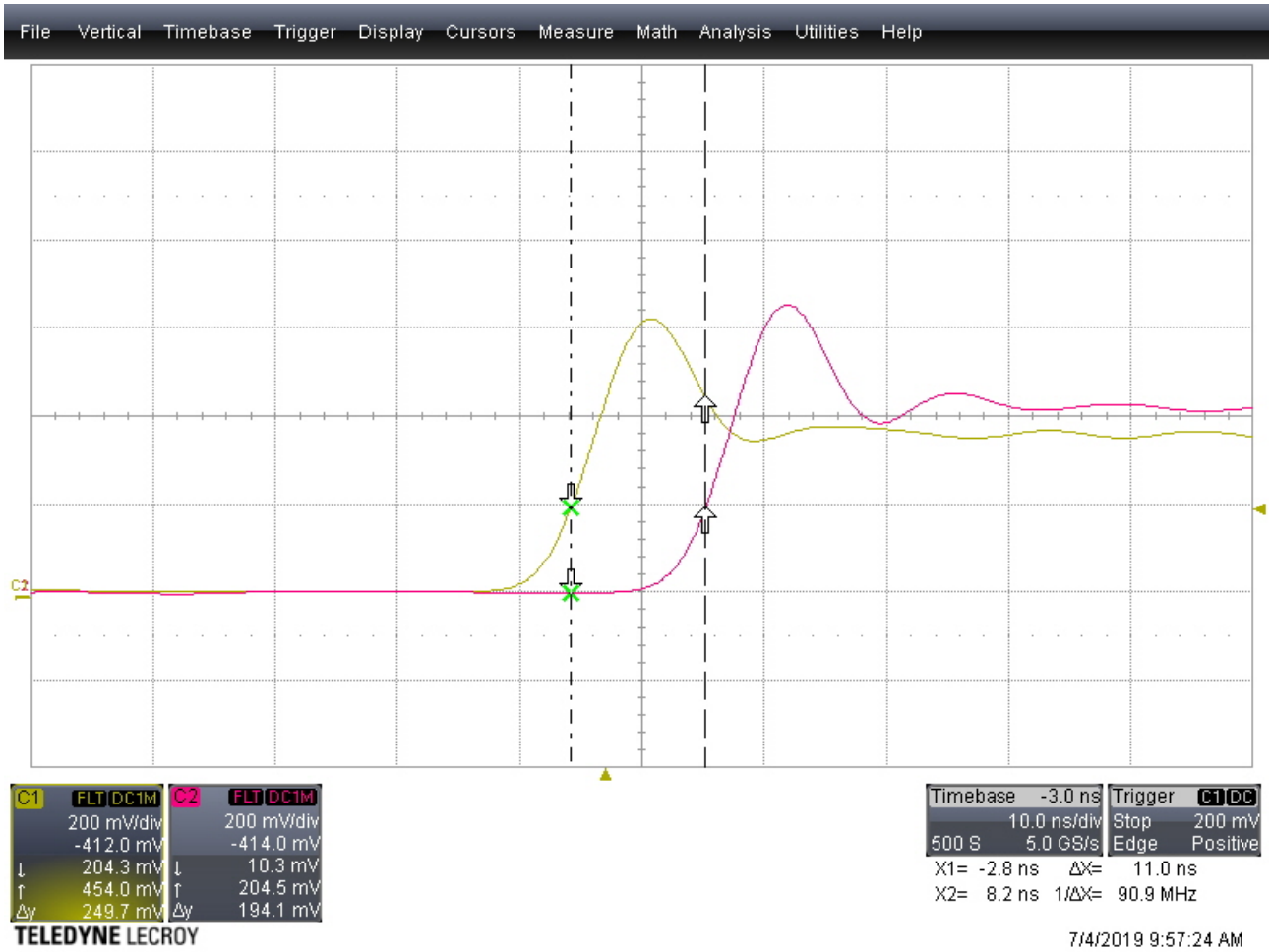
MPS Trigger Mode:



To generate a manual trigger for debugging purposes, just raise the second bit of the MANUAL_CTRL register (address 0x294) as in the picture below. And lower that to zero before sending another trigger event. You'll see the data updating when the trigger signal is detected.

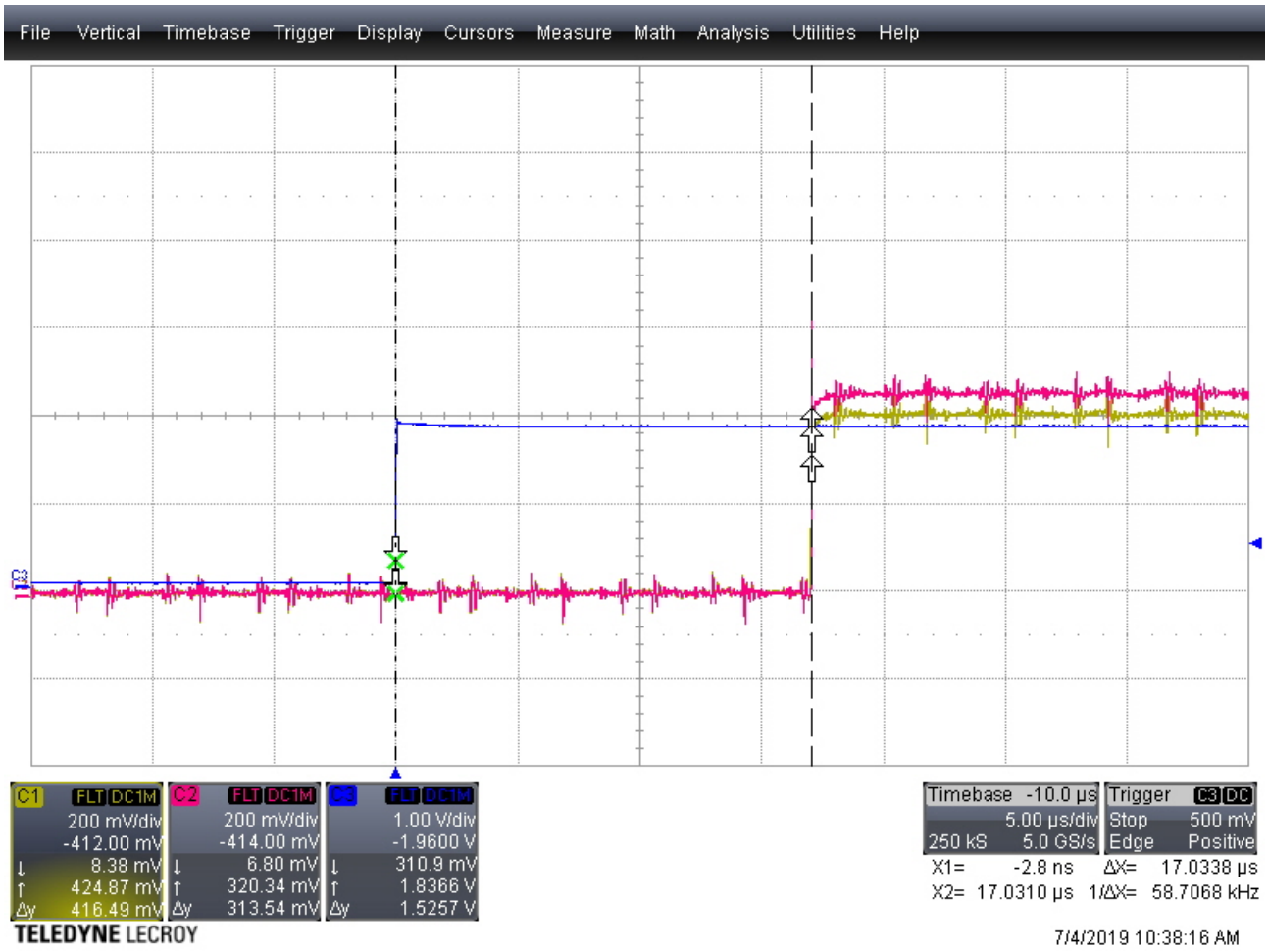
```
Written 0x0; readback 0x0
[14:41:41 - 19-08-02 - caenels@caenels-uTCA] /home/caenels
$ sudo pcimem /sys/bus/pci/devices/0000:03:00.0/resource0  0x294 W 0x2
/sys/bus/pci/devices/0000:03:00.0/resource0 opened.
Target offset is 0x294, page size is 4096
mmap(0, 4096, 0x3, 0x1, 3, 0x294)
PCI Memory mapped to address 0x7f714394b000.
Written 0x2; readback 0x2
[14:42:03 - 19-08-02 - caenels@caenels-uTCA] /home/caenels
$ sudo pcimem /sys/bus/pci/devices/0000:03:00.0/resource0  0x294 W 0x0
/sys/bus/pci/devices/0000:03:00.0/resource0 opened.
Target offset is 0x294, page size is 4096
mmap(0, 4096, 0x3, 0x1, 3, 0x294)
PCI Memory mapped to address 0x7f83db1cc000.
Written 0x0; readback 0x0
[14:42:05 - 19-08-02 - caenels@caenels-uTCA] /home/caenels
$ sudo pcimem /sys/bus/pci/devices/0000:03:00.0/resource0  0x294 W 0x2
/sys/bus/pci/devices/0000:03:00.0/resource0 opened.
Target offset is 0x294, page size is 4096
```

MPS M-LVDS latency:



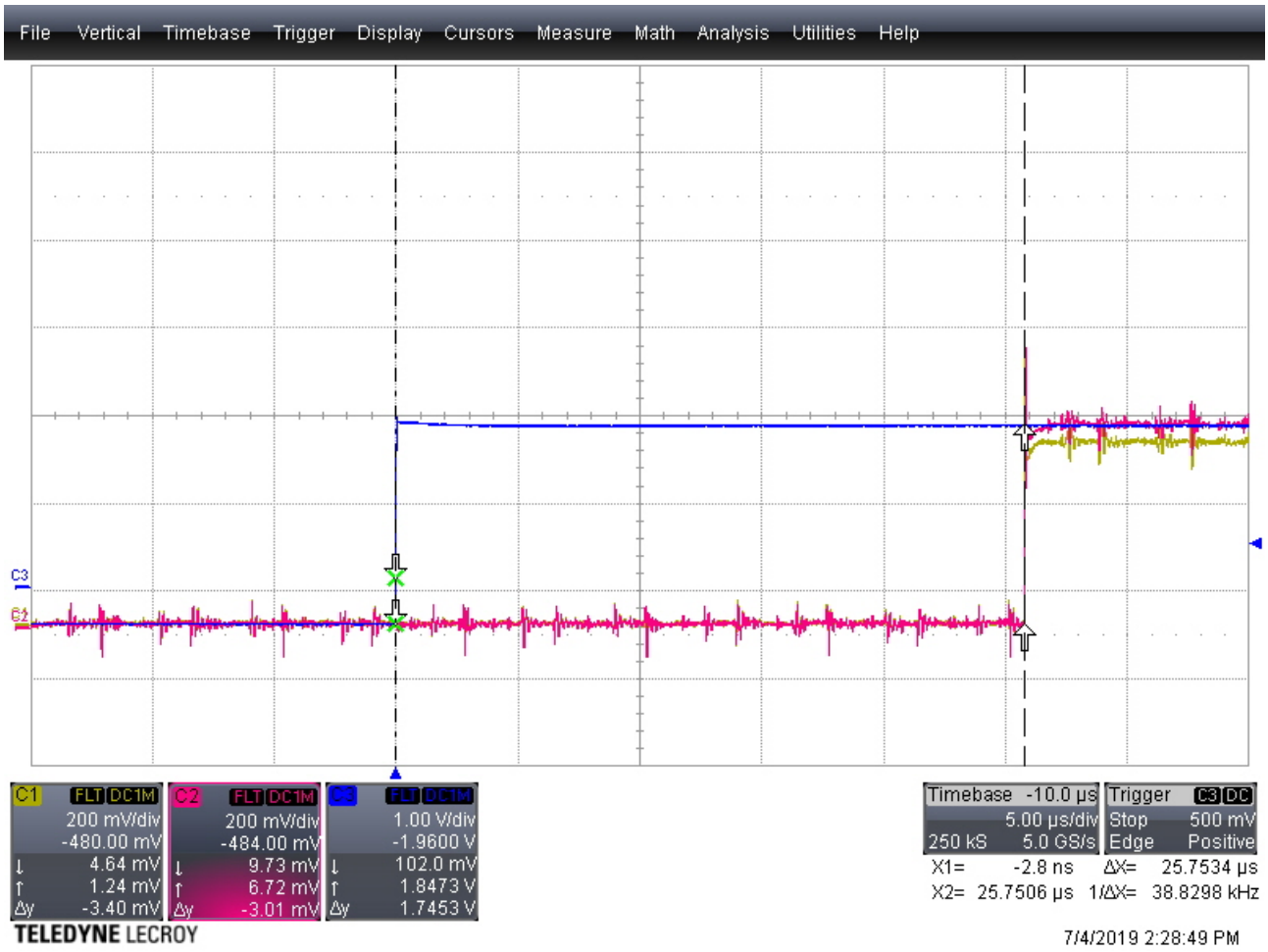
This is the latency between one board (AMC-PICO8-MPS) and a fake TPCTRL (in this case a DAMC-FMC25). As you can see, the backplane connectivity is very fast, 11 ns to send the signal from the AMC-PICO8 to the TPCTRL.

MPS detection and generation with N_SAMP = 1:



This is the delay from an input signal (blue curve) to the MPS generation. We configured the AMC-PICO8-MPS in order to generate the MPS signal just with **ONE** sample over threshold. The latency is 17 us = 1 us (sampling) + 16 us (FMC-PICO-1M4 analog filter).
The cut off frequency of the standard FMC is around 10 kHz.

MPS detection and generation with N_SAMP = 10:



This is the delay from an input signal (blue curve) to the MPS generation. We configured the AMC-PICO8-MPS in order to generate the MPS signal **N_SAMP = 10** samples over threshold. The latency is around 26 us = 10 us (sampling 10 values) + 16 us (FMC- PICO-1M4 analog filter).