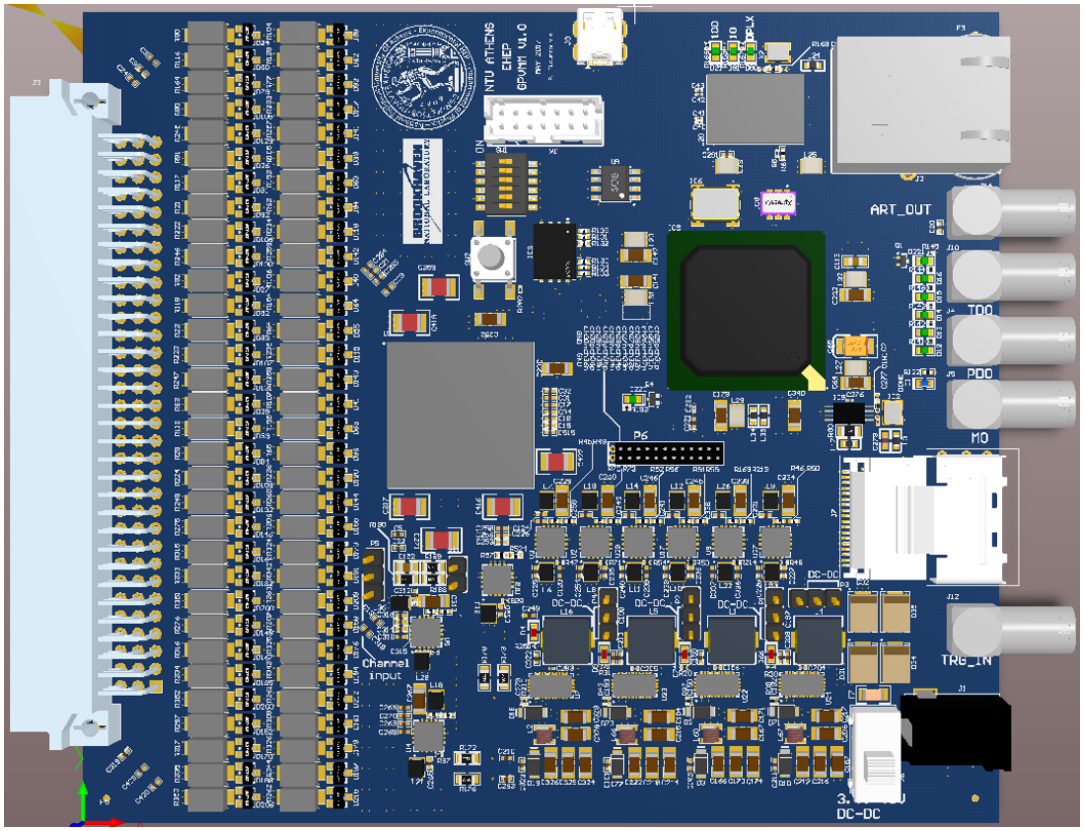
User Manual

GPVMM3 Board
Version 1.0



T. Alexopoulos[1], D. J. Antrim[3], C. Bakalis[1,2], P. Gkountoumis[1,2], G. Iakovidis[2], A. Koulouris[1,2], D. Matakias[1,2], P. Moschovakos[1,2], V. Polychronakos[2], P. Tzanis[1]

[1] *National Technical University of Athens*
[2] *Brookhaven National Laboratory*
[3] *University of California, Irvine*

v1.3, December 5, 2017

Contents

Chapter 1

GPVMM Board

The GPVMM v1.0 board has been designed by the National Technical University of Athens and Brookhaven National Laboratory for general purpose readout use. It Includes a VMM3 ASIC with 64 channels, a Xilinx® FPGA Artix part-ID (*xc7a100t_0*) and a connector (P.N 9-1393644-1) for direct connection with the detector. The various information of the board are depicted in figure 1.1. The board is power supplied with $+3.3\,\mathrm{V}$ or $+3.4\,\mathrm{V}$ to $+42\,\mathrm{V}$, selected by power switch and jumpers on the board. For the selectable power P1, P2, P3 and P4 must be set accordingly with the help of overlay test for the selection of DC-DC, as is shown in figure 1.2.

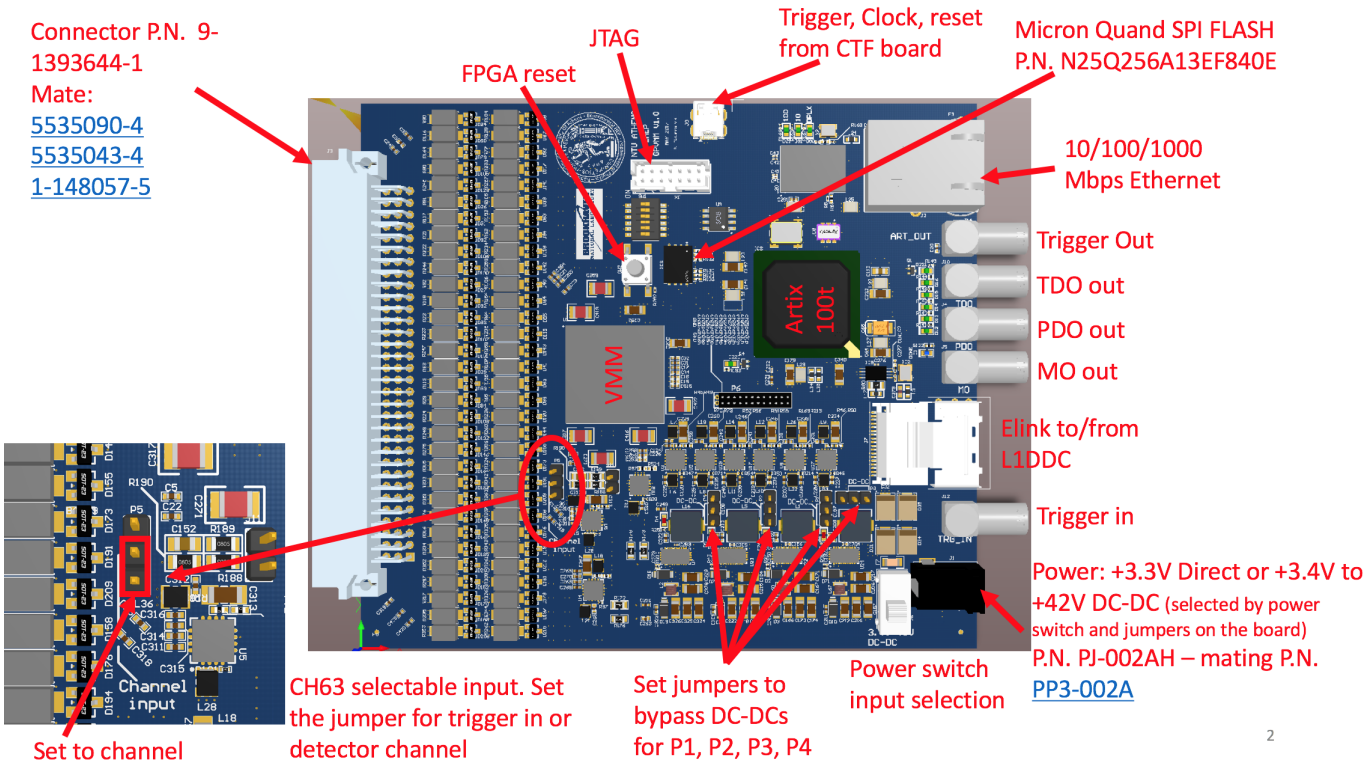


Figure 1.1: GPVMM v1.0 board

The board gives the user the ability to set the jumper for trigger in or detector channel, re-program board's firmware via JTAG and get trigger, clock and a reset signal from CTF module connected to the board via a uHDMI connector. The presence and interconnection of the GPVMM board with the CTF module is mandatory, if the firmware that has been loaded to the FPGA is *gpvmm_withCTF.bin*[1].If the loaded firmware is *gpvmm_noCTF.bin*, then the GPVMM can be operated standalone. In any case, the LED labeled as *D15* should turn on, which means that the FPGA's PLL has been locked. The communication between PC and GPVMM is implemented through

[1]The firmware's .bin files can be downloaded from `https://gitlab.cern.ch/NSWelectronics/vmm_boards_firmware` and the branch that contains the .bin files for the GPVMM is *VMM3_MASTER _GPtemp*.

10/100/1000 Mbps Ethernet and there is an optional connection to or from L1DDC board through E-link via the miniSAS connector. The schematic pinout of uHDMI and miniSAS is shown in figure 1.4 . The miniSAS cable is connected to 1.2 V banks and should be attached to CTF board and CML drivers (max 1.2 V). Also, miniSAS can be corverted to AC coupled with the replacement of 0805 resistors with capacitors.
However, the board includes ESD protection using diodes for protection for all the 64 channels input from positive and negative pulses, based in figure 1.2 schematic. The status of FPGA and firmware can be checked by the DONE signal of the FPGA blue led on the board.

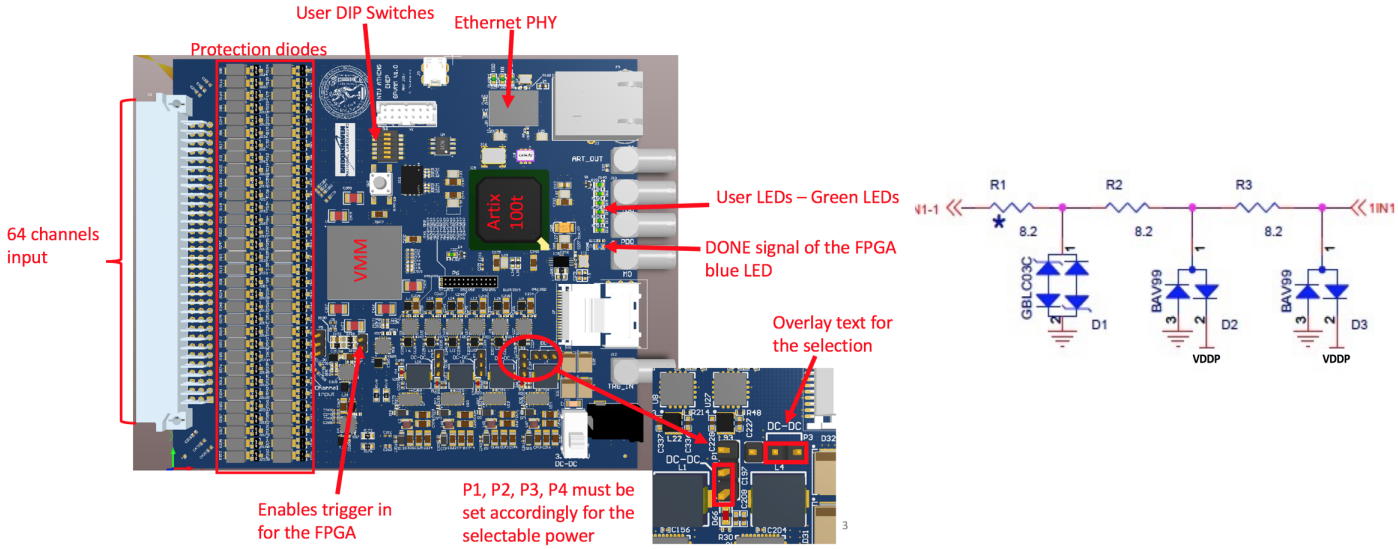


Figure 1.2: ESD Protection of GPVMM board

Also, GPVMM boards includes a 26-pin Header for direct output of 12 VMM channels and the header pin mapping is shown in figure 1.3. The input channel mapping of the board's connector is also imprinted in figure 1.3.

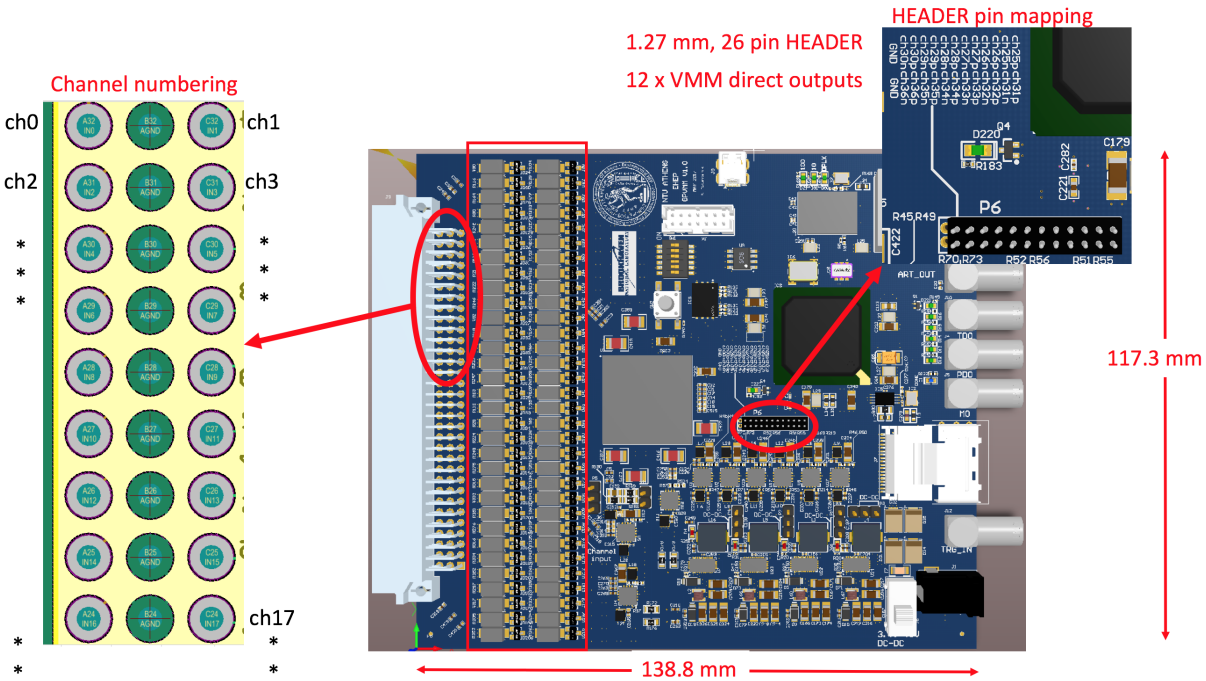


Figure 1.3: Input channel and header output mapping of GPVMM board

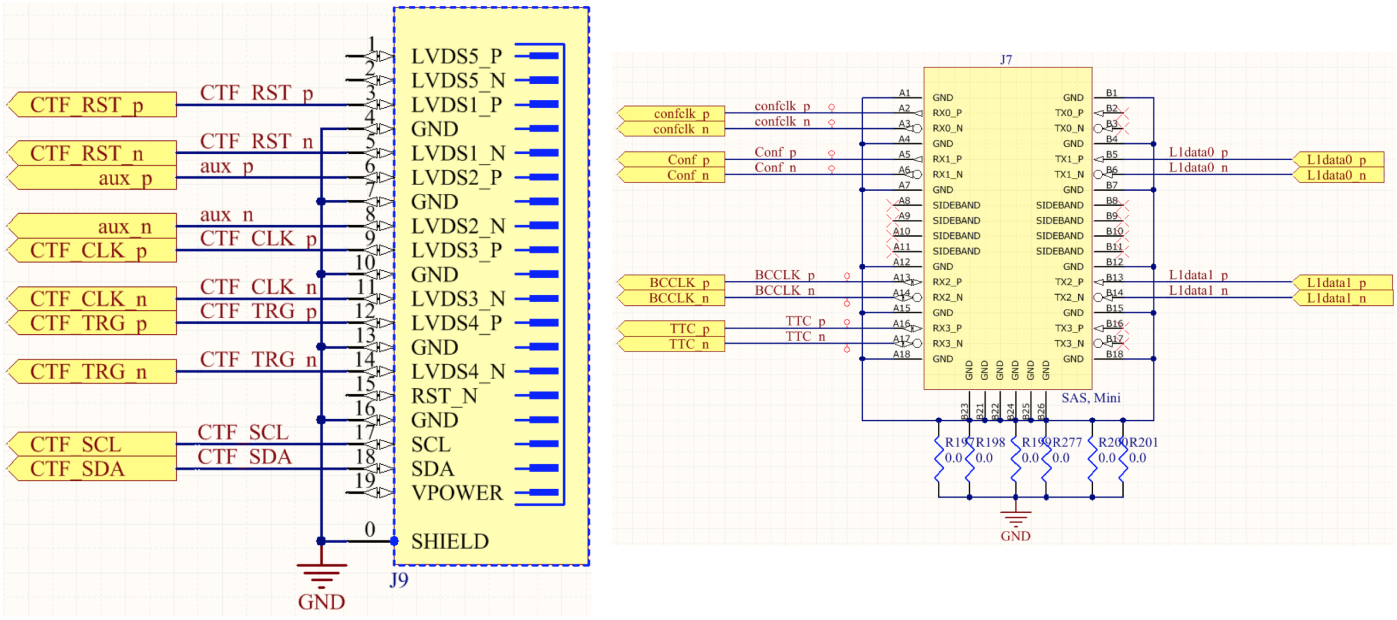


Figure 1.4: The schematic pinout of uHDMI and miniSAS

Chapter 2

Firmware

This chapter includes information about re-programming the FPGA and some aspects of the VMM Readout/Configuration Firmware of the GPVMM board. Be advised that there are two versions of the firmware, with or without the support of CTF module. The firmware's .bin files can be downloaded from `https://gitlab.cern.ch/NSWelectronics/vmm_boards_firmware` and the branch that contains the .bin files for the GPVMM is *VMM3_MASTER _GPtemp*.

2.1 Configuring the GPVMM Board

In order to program the GPVMM board, i.e. pass the firmware to its FPGA, one has first to open up Vivado® and go to the *Hardware Manager*, as can be seen from figure 2.1:

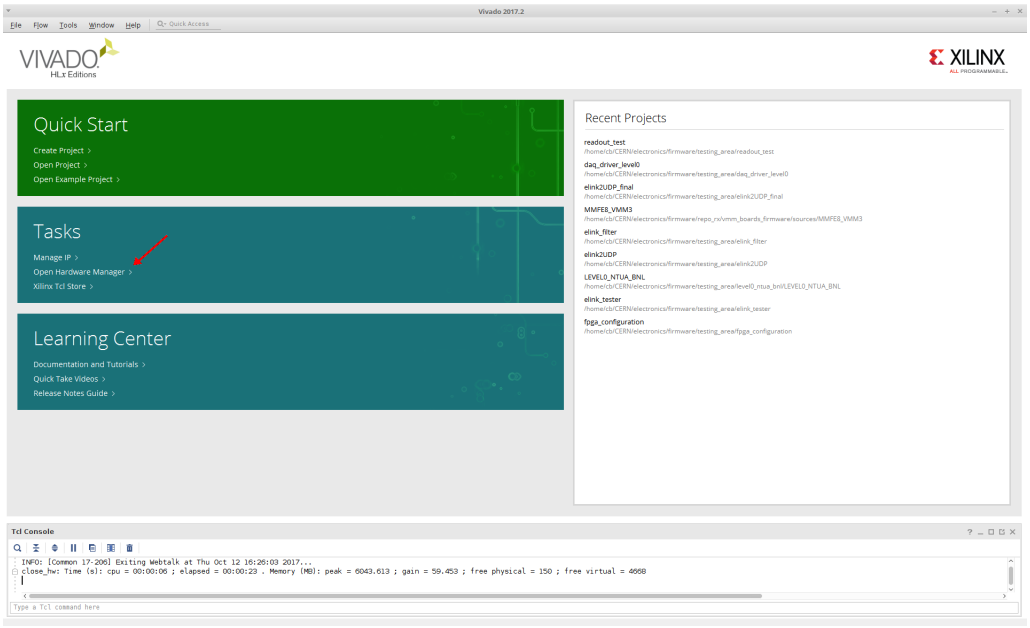


Figure 2.1: Select Hardware Manager

The next step would be to *Open Target*, i.e. initialize the communication with the FPGA chip, via JTAG. It is now assumed that the host PC is connected to the board via the programming cable, and that the board is powered-on. Click on *Auto-Connect*.
Vivado® will now try to communicate with the FPGA. If all goes well, then you will see the serial number of the programming cable (*xilinx_tcf/Digilent/2102499* on figure 2.3) and the FPGA part-ID (*xc7a100t_0*) on figure 2.3). On figure 2.3, after right-clicking on the device, the user clicks on *Program Device...* which will directly program the FPGA of the board. Note that after loading up the firmware this way, if the board is power-cycled, this firmware will be lost from the FPGA.

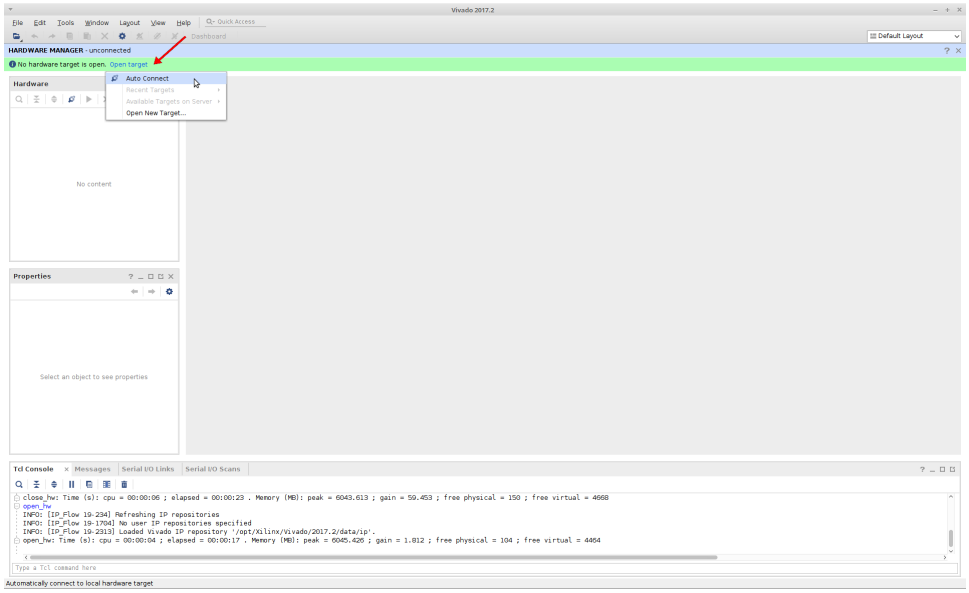


Figure 2.2: Open Target

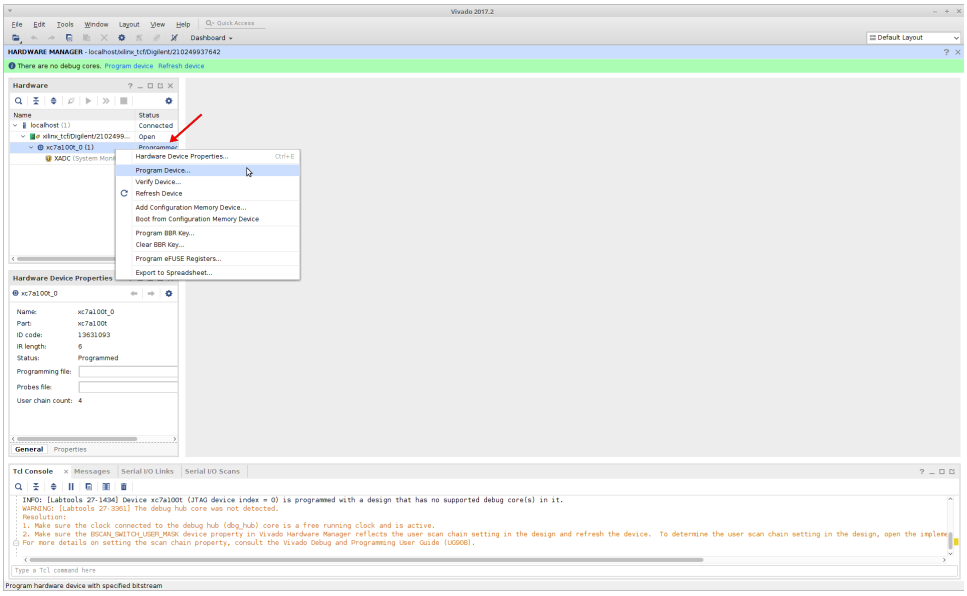


Figure 2.3: Programming the FPGA directly

Then the user should click on the browser button of the pop-up window, and select the appropriate .bin or .bit file. By clicking 'OK', the firmware will be passed to the FPGA; this takes some time to complete, and when the progress bar maxes out, the firmware is in the FPGA, and the user may disconnect the programming cable safely. If the user wishes to load the .bin file which contains the firmware to the flash memory instead, so that the FPGA loads it up after every power-cycle, then the process is a bit different. One has to right click on the FPGA part-ID again, and click on *Add Configuration Memory Device* (see figure 2.5).

After this, the user should select the appropriate memory that is on the GPVMM board via the menu. The device-id is *mt25ql256-spi-x1_x2_x4*, which in other versions of Vivado® is indicated as *n25q256-3.3v-spi-x1_x2_x4*. You will find this part easy if you select the appropriate options in the indicated drop-down menus at figure 2.6. Remember that there is a 3.3 V memory and a 1.8 V one in the list. You must choose the 3.3 V memory.

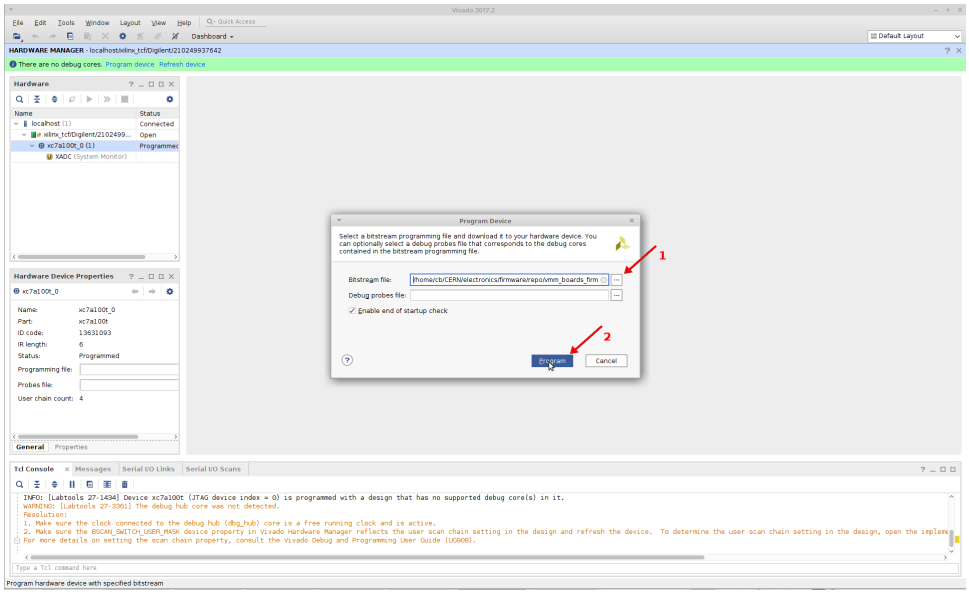


Figure 2.4: Selecting the bitfile

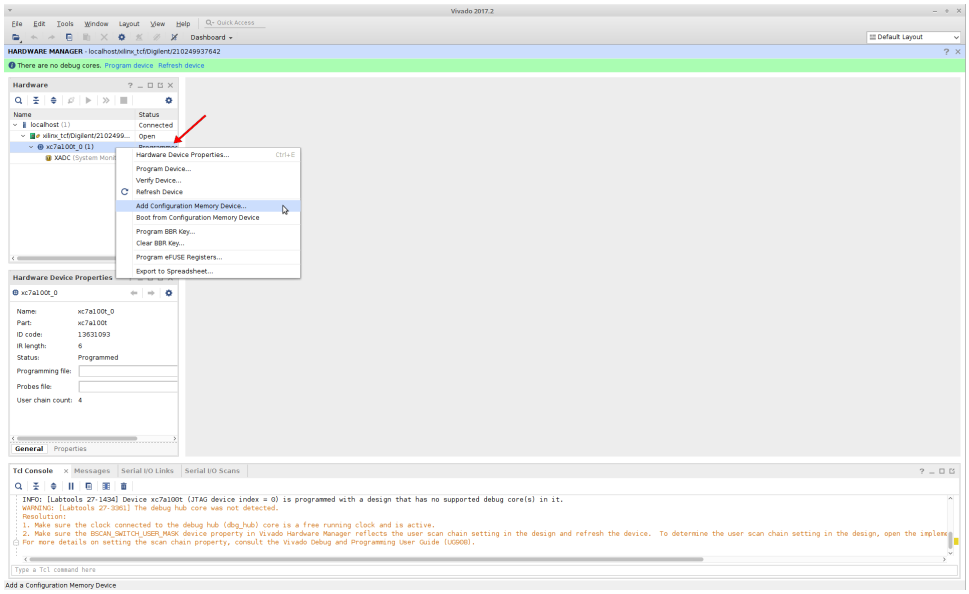


Figure 2.5: First step towards programming the flash memory

If prompted, select to program the configuration memory device now. The next dialog box will let you browse and select the .bin file you want to load into the flash memory. By clicking 'OK', the firmware will start to get written to the memory. This takes more time than the usual programming. When the process has finalized, the user has to disconnect the programming cable and power-cycle the board, in order for the FPGA to accept the firmware from the board's memory. Be advised that we sometimes have noticed an inability of the FPGA to get configured by the flash memory after a power cycling, if the programming cable is attached to the JTAG connector.

2.2 The VMM Readout/Configuration Firmware

The firmware allows the user to communicate with the VMM ASIC on the GPVMM board, via the VERSO software. The firmware is compatible with the master branch of the software (v.4.0.0 is one of the stable releases that have been tested).

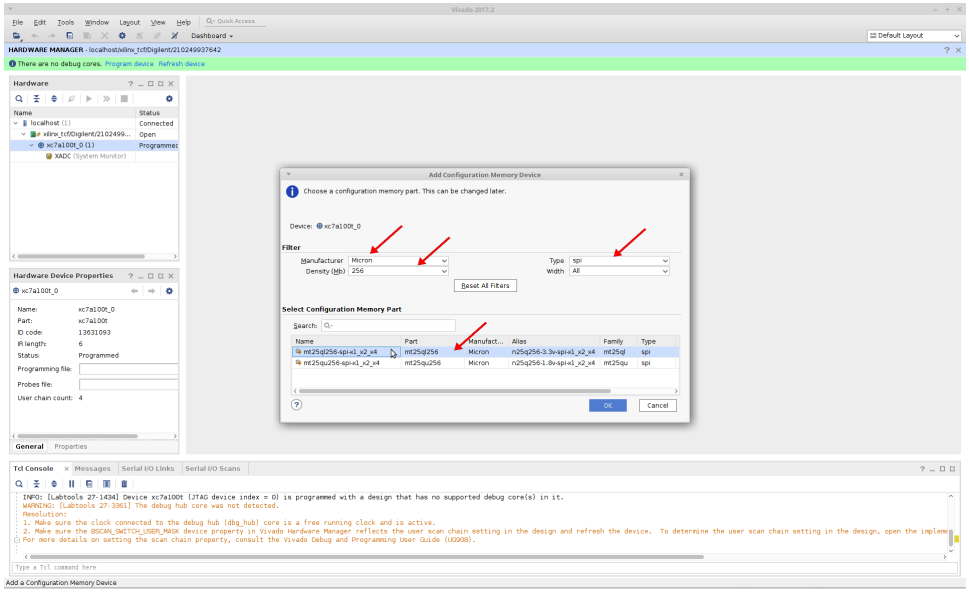


Figure 2.6: Selecting the correct flash memory

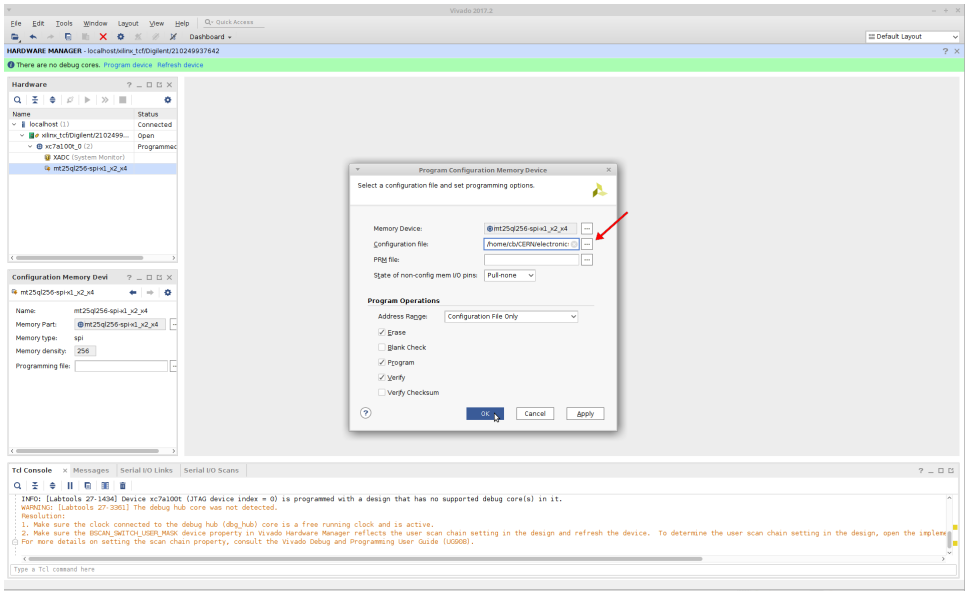


Figure 2.7: Programming the flash memory

CKBC/CKTP, xADC

The firmware is able to send the CKBC signal to the VMM, which can be configured to be either 40 Mhz, 20 Mhz or 10 Mhz. It can also send test pulses to the VMM, by asserting the CKTP signal to the VMM's input. The CKTP pulse is a periodic pulse, which can be configured by the user in terms of period, high-time, and skewing with respect to the CKBC. If the user has enabled the appropriate registers in order to activate the test pulsing circuit of each channel in the VMM, then the CKTP will induce charge at that channel, thus creating an event.

The firmware also instantiates the xADC module, which can be used for calibration of the VMM.

Dynamic IP Changing

The firmware is able to change its IP on-the-fly with the aid of the software, and keep that address over power-cycles. This is achieved by storing the IP address designated by the user in an on-board memory. Upon firmware start-up, the logic reads out the information from that memory, thus retaining the address. By changing the IP

address of the board, the user is able to use several boards in the same network, thus allowing for data taking with several boards/VMMs. Be advised that the only IP addresses accepted by the firmware can be of the form of: $192.168.0.x$, $x = 0 - 255$. The default address of the board is $192.168.0.2$.

VMM Configuration

Naturally, the user may configure the VMM through the software and FPGA firmware. The user makes the appropriate selections at the GUI, and by pressing 'Configure', the software sends the configuration bits via UDP to the FPGA, to be picked up by the firmware. The firmware then configures the VMM3 via the SPI protocol. After completing the configuration of the VMM, the FPGA sends a configuration reply packet to the software. The software does not process this reply, but it can be monitored by the user with an appropriate network monitoring tool (e.g. *Wireshark*). A configured VMM can yield data to the user. The firmware then must readout the VMM, and forward the data to the software for decoding.

2.3 Reading-Out the VMM

In principle, two readout modes are supported: *Continuous*, and *Level0*. Each readout mode is implemented by its respective firmware bitfile.

Continuous Mode

The continuous readout mode, which was used in the VMM2, and is still present as a readout option for VMM3, is the most simple readout method of the VMM. Once an event gets buffered into the VMM's FIFOs, it is readily available to be readout by the FPGA. Upon the reception of a trigger signal by the board, either internal (associated with the CKTP), or external (e.g. from scintillators), the FPGA will start querying the VMM for data by asserting the CKTK/CKDT signals. Once the VMM's buffers are empty, the event readout has finalized, and the firmware will send the data via UDP to the VERSO software for decoding. The FPGA will start the readout process, after a fixed latency with respect to the trigger. This latency can be input by the user through the GUI.

Level0 Mode

The level0 readout mode, which will be used in the final experiment of ATLAS, is a bit more complicated to implement, as it needs careful configuration by the user. The level0 readout allows the user to cherry-pick data that have a specific BCID, thus allowing for selecting only data of interest. The FPGA asserts the L0 signal to the VMM's input, and the VMM responds with data packets accordingly. The L0 signal is asserted by the FPGA to the VMM, with a latency with respect to the trigger received by the FPGA (again, either generated internally by the CKTP, or externally), equal to the value set by the GUI. If the VMM's and the FPGA's configurations are correct, then data will be read out. If the configuration is incorrect, then the VMM will send out empty packets (only headers).

The VMM has two registers which should be enabled *(sL0ena, sL0enaV)*. In order for the user to read out data via level0 and CKTP assertion, then he/she must do the following: First calculate the difference between the *l0offset* VMM register and the *rollover* VMM register. Usually the rollover is set to 4095. The difference between these values should yield a number that must be multiplied by the CKBC's period (usually $25\,\text{ns}$), thus giving a time value. (e.g. $4095 - 4060 = 35 \times 25\,\text{ns} = 875\,\text{ns}$). Upon receiving a L0 signal from the FPGA, the VMM will "look back" for data at that specific time (in the example given it will go back $875\,\text{ns}$) . So, in order for the user to get data with CKTP, the latency at the GUI (which in the level0 readout mode is the latency between the assertion of the L0 signal with respect to the trigger assertion), should be equal to that difference (for this example, a latency of $143 \times 6.25 = 893.75\,\text{ns}$ should be correct). So the FPGA would readout the data associated with that CKTP.

If the user wants to get data with level0 mode and external trigger, then after calculating the difference between the two aforementioned registers, he/she should take into account the intrinsic latency between the actual trigger assertion with respect to its reception by the board, plus the latency at the GUI. If the sum of these two latencies

are about equal with the time value yielded by the *l0offset* and *rollover* difference then the user should see monitor data coming out of the VMM and FPGA.

Be advised that these are *general guidelines* to get data with level0 readout mode. If those two time values are equal or about equal to each other, and the VMM is refusing to give data, then this usually means that the user is out of the window and is pointing towards an incorrect ”BCID area” of the VMM. By changing the values of the latency, or by reconfiguring the VMM with different *l0offset* values one should try to ”find” the data in the VMM’s buffers. Also, when ”looking” for data like this, it is considered good practice to set the *window* VMM register to 7, in order to maximize the window opening width.

Other Guidelines and Tips

The user should also bear in mind that when the FPGA is in ACQ ON state, it can only go to ACQ OFF state and will ignore any other configurations. When the user wishes to configure the VMM, then he/she should do so when the FPGA is in ACQ OFF state. When the user wishes to set different values to the FPGA registers (i.e. change the latency, change the CKTP characteristics etc.), he/she should do so when the FPGA is in ACQ OFF state.

The user is advised to ping the board before attempting to operate the software. The default IP is 192.168.0.2.

A useful tool to monitor the transactions between the FPGA and the PC that hosts the software is *Wireshark*. The packets that are exchanged between the software and firmware can be seen at this application’s GUI, thus giving an overview of the state of the setup (e.g., if the FPGA sends packets of only a few bytes while in ACQ ON, then the VMM does not send out any data).

Chapter 3

Software

This chapter describes the instructions for installing and using the official VMM Ethernet Readout Software (Verso). The Central DAQ Sofware for configuring VMM-based front end boards in continuous or L0-trigger mode and a arbitrary of front-end boards, UDP based readout and event building. The VERSO is a graphical interface with monitor and calibration capabilities and developed alongside the baseline NSW electronics NTUA/BNL firmware.

3.1 Installation

The VERSO Software for the GPVMM board hosted on NSWElectronics GitLab under : `https://gitlab.cern.ch/NSWelectronics/vmm_readout_software` .
The recommended release is v4.0.0 which is for VMM2 and VMM3 readout. To obtain this release do:

```
git clone -b v4.0.0 https://gitlab.cern.ch/NSWelectronics/vmm_readout_software.git
```

The software requirements are:

- Qt 5.7
- ROOT 5.34
- Boost 1.60
- C++11 (gcc>=4.7)

There are a few steps that need to be taken in order for you to obtain, install, and get the DAQ software running. You can follow the insructions giver under the GitLab url mentioned above.
! After the installation of these softwares it is important for the VERSO to set the enviromental parameters of Qt, Boost and ROOT packages under the bashrc of your terminal:

```
alias qmake=<path-to-Qt-directory>/Qt/5.7/clang_64/bin/qmake
export DYLD_LIBRARY_PATH=<path-to-boost-directory>/boost/lib:$DYLD_LIBRARY_PATH
export ROOTSYS=<path-to-ROOT-directory>
export PATH=$PATH:$ROOTSYS/bin
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
export DYLD_LIBRARY_PATH=$ROOTSYS/lib:$DYLD_LIBRARY_PATH
```

3.2 Overview

After the succesful installation of the VERSO software, user is ready to start the GUI by running the executable verso file under the VERSO installation path. On Linux machines this executable, by default, will be located in vmm_readout_software/build/ . On typical MAC machines this executable tends to be located in vmm_readout_software/build/vmmdcs.app/Contents/MacOS/ . Whichever the case may be, the DAQ can be opened up by double-clicking on it from a graphical file-browser or by executing it from within a terminal:

```
./<path-to-vmmdcs-executable>/verso
```

The VERSO interface will start up and on figure 3.1 is what it looks. The different areas of VERSO software will be explained below.

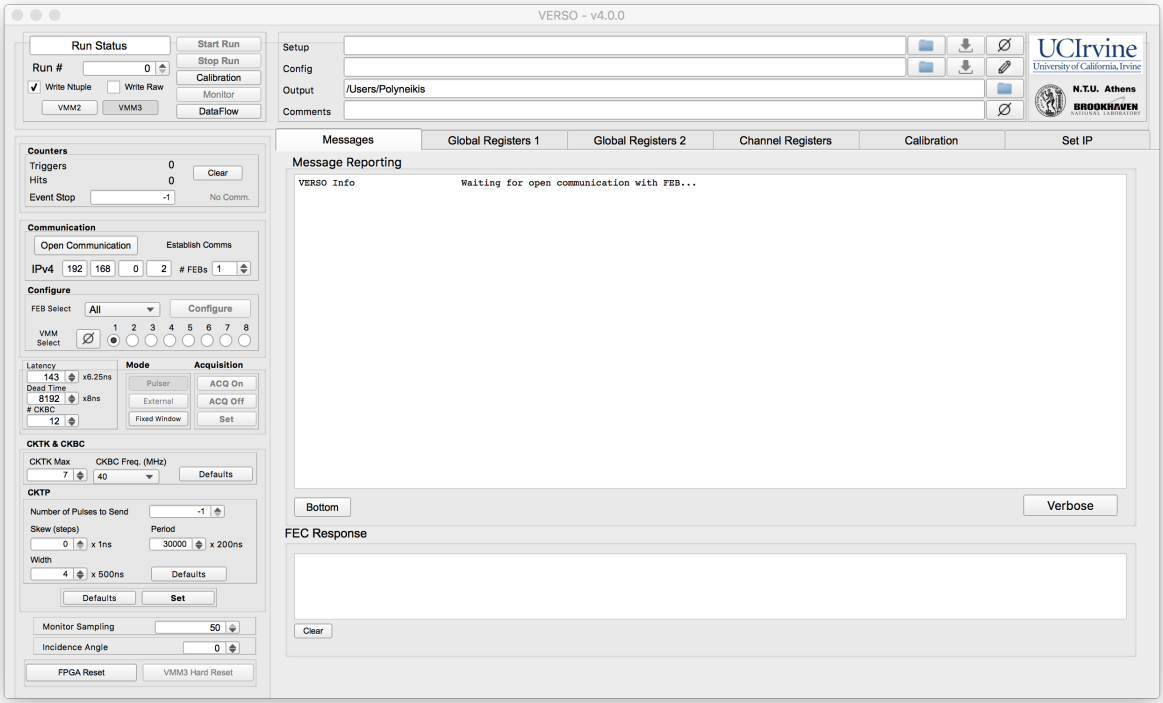
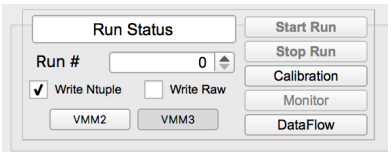


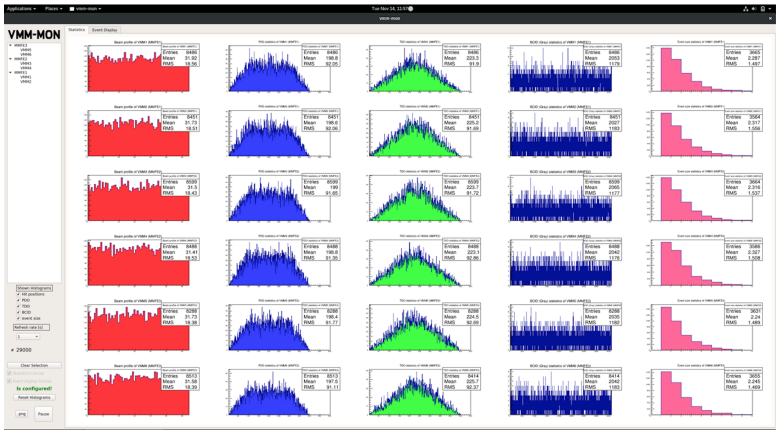
Figure 3.1: Startup of VERSO interface

Run Control



Displays the number and status of the data acquisition run which is in progress and gives the ability to the user to dump event-indexed ROOT n-tuple and/or raw data file. The user can also start & stop the readout/run and set VVM2, VMM3, L0 readout and config. By pressing calibration, user set the run type to calibration and will begin configured calibration loop when run starts with the callibration settings that user have configured to the Calibration Tab settings. By pressing DataFlow, user opens up VERSO dataflow monitor and by pressing Monitor, user enable sending event samples to VMM monitor (`https://gitlab.cern.ch/aikoulou/vmm-mon`).

Monitor



The vmm-mon application is a single window app to monitor the data that VERSO records. The main panel shows the plots, depending on the settings in the left panel.

- The list of boards and VMMs is shown, and the user can select any item in the tree to show the corresponding plots in the main panel.
- Settings: refresh interval, type of plots to show (PDO, TDO etc),
- Clear selection in the tree, in order to make the main panel show only VMM plots.
- Reset histograms (resets all the histograms)
- Pause/Resume
- .png, makes a png snapshot of the main panels and saves to the directory of the vmm-mon.

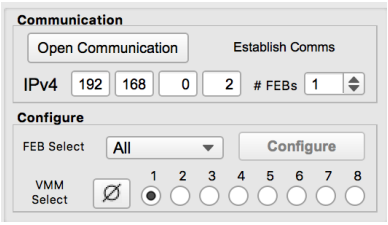
Top-level Configuration Parameters

In Config field, user can load, set and write a configuration xml file with the VMM settings that has already have done to the VERSO interface. The procedure to write a new config file is to set the interface parameters, press the ”pen” button and specify the directory that user wants to save the config file. In this way, user has the ability to load the VMM configuration by pressing the folder icon, load the config file and press the ”download” button to set the configuration parameters to the VERSO interface. In Output field, user can specify the directory location to dump the output files and in Comment field, user can comment/text to store as a field inside of ROOT n-tuple. Also, VERSO gives and optional Setup field, where user can load the DAQ setup configuration such as detector-to-elx channel mapping and detector mapping of detectors and electronics that he can find under vmm_readout_software/readout_configuration directory.

Counters

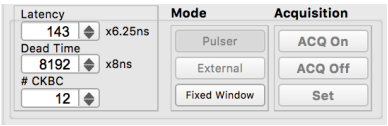
Displays the total number of triggers (events) recorded and the total numberof unique VMM channel hits recorded. In Event Stop field, user can set the number of event to record in the run and by setting -1 there is no limit of the events.

Communication and Configuration



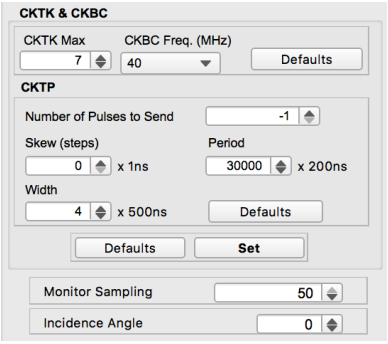
By pressing Open Communication, user setups UDP paths and test (ping) FEB connection. In IPv4 field, user can specify the ip of the VMM(FEB) board and sets the adress of the first FEB and also the number of FEBs to talk to. In Configuration Frame, user can Select the FEB number that wants to send the VMM configuration parameters and also sets to which FEB user will send all other commands. In VMM Select, user specifies the VMM id on selected FEB to configure. By pressing Configure button, user sends VMM configuration to the selected FEBs and VMMs. If user want to use multiple FEBs and take data from all of them, should set the IPv4 of the first FEB (e.g 192.168.0.2) , select the total number of FEBs and send the configuration to All FEBs with 1 VMM per board (such as GPVMM3). For a successful data acquistion of multiple boards, FEBs ip addresses must be in ascending order, e.g 1st FEB 192.168.0.2 2st FEB 192.168.0.3 etc.

Triggers and Acquistion Mode



The user is able to change the trigger latency and dead-time. Also, can set the Acquistion/Trigger Mode by enabling the Pulser (internal) or External VMM trigger mode. A third option given, Fixed Window, where user can send specific number of CKBCs. By pressing Set, user sends trigger settings and acquisition mode command to FPGA of the GPVMM board. By AQN Off and AQN On, user can turn off/on VMM adquistion mode.

FPGA clocks Configuration

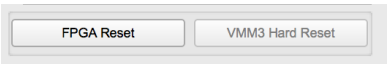


User can set the number of CKTKs and also the frequency of CKBC (40,20 and 10 MHz CKBC possible). The CKTP frame is for the configuration of the CKTP (test pulse), where user sets fixed number of pulser to send (<0 for no limit) and also the CKTP skew w.r.t CKBC (1ns steps for 40ns clock, 6.25 ms steps otherwise). By pressing Set, user sends CKTK, CKBC and CKTP configuration to the FEBs.

Monitor Sampling and Angle

User selects the event sampling rate parameters for sending to vmm-mon application and also the incident angle of detector to be stored in output ROOT n-tuple.

FPGA & VVM Hard Reset



Sends FPGA reset or VMM hard reset command.

Global Register I & II & Channel Registers

The VERSO software gives user the ability to set the Global Configuration Registers of the VMM3. The associated bit names of the registers in Global Registers Tabs are described anatically in official VMM3 docu-

mentation which can be found under NSWElectronics Twiki (`https://twiki.cern.ch/twiki/pub/Atlas/NSWelectronics/vmm.pdf`) . Under the tabs Global Register I & II ,figure 3.4, user can set most common register such as Test Pulse in DAC Counts, Threshold in DAC Counts, Channel Gain, Peak Time, TAL Slope, Channel Polarity, Channel Monitor etc. In Channel Registers, figure 3.3, user can set the registers the 64 channels of VMM3 and also mask/unmasked the channels by the option SM/ST.

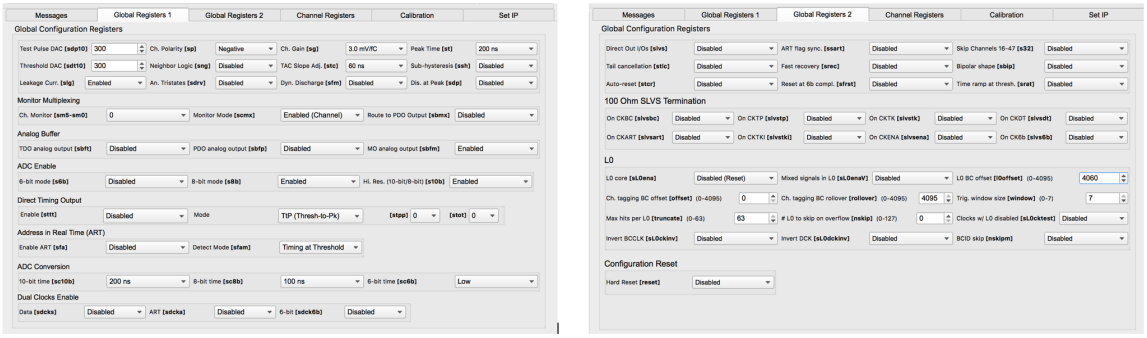


Figure 3.2: Global Registers I & II Tabs

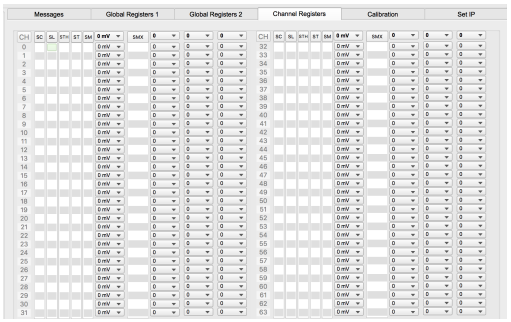
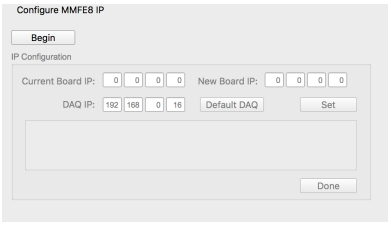


Figure 3.3: Channel Registers Tab

FEB IP Configuration Panel



Under the Set IP tab, user can change the default ip of the FEB card (default: 192.168.0.2) to a new board ip. By pressing Set, user sends the IP configuration to the FPGA. After the change of ip, it is mandatory for the user to re-set the IPv4 field with the new FEB's ip under the Communication Frame of VERSO software.

Calibration

Under Calibration Tab, user can perform calibration routines such as xADC and Non-xADC calibration type. In xADC based sampling calibration type, user can perform calibration routines to sample:

- Analog DAC levels (threshold and pulser)
- Channel-by-channel threshold variations, stepping over the VMM threshold trimmers values
- Channel baseline & noise levels

In non-xADC based sampling calibration type, user can perform calibration routines to perform loops under the shown loop parameters such as Gain, Peak Time, TAC Slope, Test Pulse DAC etc. Further information for the calibration techniques can be found under Daniel Antrim's presentation (`https://indico.cern.ch/event/`

671377/contributions/2746181/attachments/1536796/2407884/2017_10_6_verso_and_calibration_overviewPDF.pdf). Also, a calibration software has also been developed to perform calibration analyses on the output of the calibration runs from VERSO software and can be found under NSWElectronics GitLab (https://gitlab.cern.ch/NSWelectronics/vmm_calibration_software).

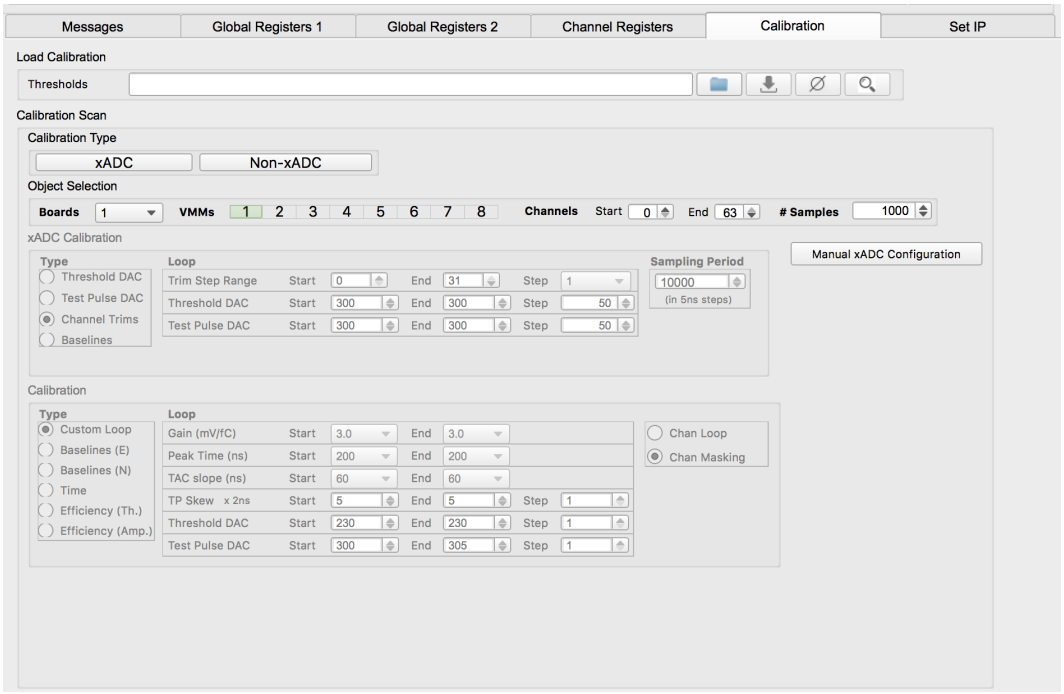


Figure 3.4: Calibration Tab

Chapter 4

Running Procedure

This chapter describes the right steps for succesful data acquisition either with internal trigger (pulser) and external trigger (scintillator).

4.1 Communication

The very first step for the right communication with the board is to set static IPv4 of the desktop's network card with the below network settings:

- IP Address : 192.168.0.16
- Subnet Mask : 255.255.255.0

After the setup of network settings, user plugs the GPVMM board in the desktop pc through a ethernet cable and supplies the board with $+3.3$ V or $+3.4$ V to $+42$ V DC-DC (selected by power swich and jumpers on the board). User can check the communication with the board by pinging the board's IP address (default IP: 192.168.0.2) :

```
ping 192.168.0.2
```

Also, user can find the ip of the board by broadcast ping:

```
ping -b 192.168.0.255
```

If the ping is succesful and the Ethernet LEDs are blinking, user can continue with the experiment setup. If not, user should connect GPVMM board to CTF board via uHDMI to get the reference clock, once the connection is succesful the D15 LED ("PLL/MMCM Lock" signal) will turn on and Ethernet LEDs will start to blink.
Also, it is mandatory to mentioned that there are two different versions of the GPVMM board's firmware, with reference clock via uhdmi with CTF board and another without the reference clock. The default firmware of GPVMM boards needs CTF reference clock and in addition user should follow Chapter 3 to program GPVMM with the right firmware.

4.2 Setup

The experiment setup for data acquisition using GPVMM board is shown in figure 4.1 below. The setup consisted of the user's detector, PC for the communication with GPVMM board through ethernet cable, oscilloscope for the monitor output of the VMM channels, power supply, scintillators for the trigger mode(TTL pulses) and also a fan is appropriate for the cooling of VMM and FPGA chip. Also, the use of a CTF board for reference clock is optional and it depends on the GPVMM's firmware. User in order to use the reference clock from CTF board must not connect GPVMM board directly to CTF via uHDMI, but user must use a compatible HDMI adapter which has designed by NTUA. Additional information for the HDMI adapter board can be found at Appendix F.

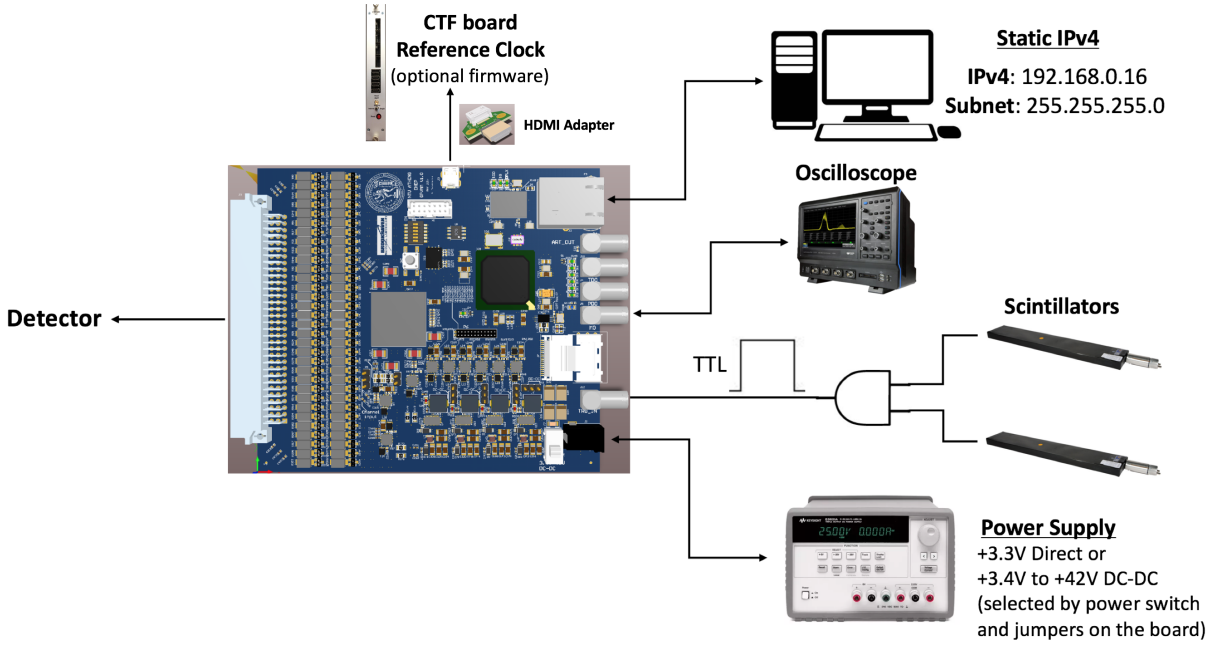


Figure 4.1: Experimental Setup

Internal Trigger (Pulser)

The steps below describe the procedure the user must follow to be able to record data with the Internal Trigger Mode (Pulser):

1. Start VERSO software
2. Set board's IP in IPv4 field
3. Press Open Communication button
4. Set Global Register's 1 & 2 parameters
5. Send Internal Pulse[ST] or Mask[SM] channels through Channel Registers Tabs
6. Select Pulser Mode In Mode Frame
7. Press FPGA Reset button
8. Press VMM3 Hard Reset button
9. Press Set button in CKTP Frame
10. Press Set button in Mode Frame
11. Press Configure button to sent the configuration parametes to VMM

External Trigger

The steps below describe the procedure the user must follow to be able to record data with the External Trigger Mode (e.g scintillator):

1. Connect Scintillator to TRG_IN port
2. Start VERSO software
3. Set board's IP in IPv4 field
4. Press Open Communication button
5. Set Global Register's 1 & 2 parameters
6. Mask[SM] channels through Channel Registers Tabs (optional)
7. Select External Mode In Mode Frame
8. Set trigger parameters (latency, lat.extra, dead time etc.) in Trigger Frame
9. Press FPGA Reset button
10. Press VMM3 Hard Reset button
11. Press Set button in CKTP Frame
12. Press Set button in Mode Frame
13. Press Configure button to sent the configuration parametes to VMM

Acquisition and Monitor Quideline

By successfully setting the above steps, user is ready to start the acquisition of data by pressing ACQ On in Acquistion Frame as well as record them in a ROOT n-tuple by pressing Start Run. If the procedure is succesful, the counter of Triggers and Hits will start to count the number of triggers and hits that user is recording. Anytime, user can stop the recording procedure by pressing Stop Run and analyse the ROOT n-tuple file which is placed under the directory that user have specify in Output field and also stop the acquisition by pressing ACQ OFF in Acquistion Frame. For every change that will be served by the user in the VERSO parameters and registers, it is mandatory to repeat steps 9-13 for the successful sending of registers to the VMM. Also, it is important to mentioned that user can record noisy channels of GPVMM board by not setting ST to channels in Channels Registers tab.
The user is recommended to use a oscilloscope connected to GPVMM3 board through a LEMO cable in MO ouput, to monitor the VMM channel output pulse , example of internal pulser shown in figure , which have selected to monitor by the Ch. Monitor [sm5-sm0] selection in Global Registers 1 tab in VERSO software. Otherwise, a useful tool to monitor the transactions between the GPVMM board and the PC that hosts the software is *Wireshark*. The packets that are exchanged between the software and firmware can be seen at this application's GUI, thus giving an overview of the state of the setup (e.g., if the FPGA sends packets of only a few bytes while in ACQ ON, then the VMM does not send out any data).

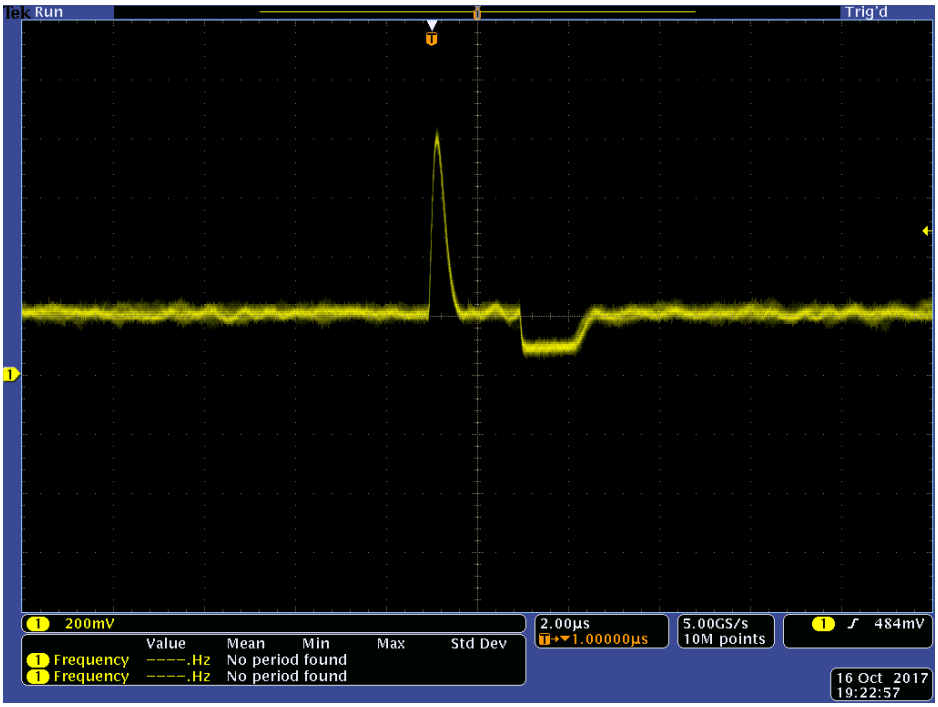


Figure 4.2: Monitor Output of VMM channel through Internal Trigger Mode (Pulser)

Appendix A

VMM3 Registers

The below table includes the Global Configuration Registers of the VMM:

Global bits (defaults are 0)	Description
sp	input charge polarity ([0] negative, [1] positive)
sdp	disable-at-peak
sbmx	routes analog monitor to PDO output
sbft [0 1], sbfp [0 1], sbfm [0 1]	analog output buffers, [1] enable (TDO, PDO, MO)
slg	leakage current disable ([0] enabled)
sm5-sm0, scmx	monitor multiplexing. • Common monitor: scmx, sm5-sm0 [0 000001 to 000100], pulser DAC (after pulser switch), threshold DAC, band-gap reference, temperature sensor) • channel monitor: scmx, sm5-sm0 [1 000000 to 111111], channels 0 to 63
sfa [0 1], sfam [0 1]	ART enable (sfa [1]) and mode (sfam [0] timing at threshold, [1] timing at peak)
st1,st0 [00 01 10 11]	peaktime (200, 100, 50, 25 ns)
sfm [0 1]	enables dynamic discharge for AC coupling ([1] enable)
sg2,sg1,sg0 [000:111]	gain (0.5, 1, 3, 4.5, 6, 9, 12, 16 mV/fC)
sng	neighbor (channel and chip) triggering enable
stot [0 1]	timing outputs control 1 (s6b must be disabled) • stpp,stot[00,01,10,11]: TtP,ToT,PtP,PtT • TtP: threshold-to-peak • ToT: time-over-threshold • PtP: pulse-at-peak (10ns) (not available with s10b) • PtT: peak-to-threshold (not available with s10b)
sttt [0 1]	enables direct-output logic (both timing and s6b)
ssh [0 1]	enables sub-hysteresis discrimination
stc1,stc0 [00 01 10 11]	TAC slope adjustment (60, 100, 350, 650 ns)
sdt9-sdt0 [0:0 through 1:1]	coarse threshold DAC
sdp9-sdp0 [0:0 through 1:1]	test pulse DAC
sc010b,sc110b	10-bit ADC conversion time
sc08b,sc18b	8-bit ADC conversion time
sc06b, sc16b, sc26b	6-bit ADC conversion time
s8b	8-bit ADC conversion mode
s6b	enables 6-bit ADC (requires sttt enabled)
s10b	enables high resolution ADCs (10/8-bit ADC enable)
sdcks	dual clock edge serialized data enable
sdcka	dual clock edge serialized ART enable
sdck6b	dual clock edge serialized 6-bit enable
sdrv	tristates analog outputs with token, used in analog mode
stpp [0 1]	timing outputs control 2
slvs	enables direct output IOs
stcr	enables auto-reset (at the end of the ramp, if no stop occurs)
ssart	enables ART flag synchronization (trail to next trail)

Figure A.1: Global Configuration Registers of the VMM

Global bits (defaults are 0)	Description
s32	skips channels 16-47 and makes 15 and 48 neighbors
stlc	enables mild tail cancellation (when enabled, overrides sbip)
srec	enables fast recovery from high charge
sbip	enables bipolar shape
srat	enables timing ramp at threshold
sfrst	enables fast reset at 6-b completion
slvsbc	enable slvs 100 Ω termination on ckbc
slvstp	enable slvs 100 Ω termination on cktp
slvstk	enable slvs 100 Ω termination on cktk
slvsdt	enable slvs 100 Ω termination on ckdt
slvsart	enable slvs 100 Ω termination on ckart
slvstki	enable slvs 100 Ω termination on cktki
slvsena	enable slvs 100 Ω termination on ckena
slvs6b	enable slvs 100 Ω termination on ck6b
sL0enaV	disable mixed signal functions when L0 enabled
reset reset	Hard reset when both high
sL0ena	enable L0 core / reset core & gate clk if 0
l0offset_i0:11	L0 BC offset
offset_i0:11	Channel tagging BC offset
rollover_i0:11	Channel tagging BC rollover
window_i0:2	Size of trigger window
truncate_i0:5	Max hits per L0
nskip_i0:6	Number of L0 triggers to skip on overflow
sL0cktest	enable clocks when L0 core disabled (test)
sL0ckinv	invert BCCLK
sL0dckinv	invert DCK
nskipm_i	magic number on BCID - 0xFE8

Figure A.2: Global Configuration Registers of the VMM (continued)

The below table includes the Channel Configuration Registers of the VMM:

Channel bits (defaults are 0)	Description
sc [0 1]	large sensor capacitance mode ([0] <∼200 pF , [1] >∼200 pF)
sl [0 1]	leakage current disable [0=enabled]
st [0 1]	300 fF test capacitor [1=enabled]
sth [0 1]	multiplies test capacitor by 10
sm [0 1]	mask enable [1=enabled]
sd0-sd4 [0:0 through 1:1]	trim threshold DAC, 1 mV step ([0:0] trim 0 V , [1:1] trim -29 mV)
smx [0 1]	channel monitor mode ([0] analog output, [1] trimmed threshold))
sz010b, sz110b, sz210b, sz310b, sz410b	10-bit ADC zero
sz08b, sz18b, sz28b, sz38b	8-bit ADC zero
sz06b, sz16b, sz26b	6-bit ADC zero

Figure A.3: Channel Configuration Registers of the VMM

Appendix B

Useful Links

1. VERSO Software, NSWelectronics GitLab, https://gitlab.cern.ch/NSWelectronics/vmm_readout_software
2. VMM Calibration Software, NSWelectronics GitLab, https://gitlab.cern.ch/NSWelectronics/vmm_calibration_software
3. VERSO VMM Monitor, NSWelectronics GitLab, https://gitlab.cern.ch/aikoulou/vmm-mon
4. ATLAS NSW Electronics Specifications, *VMM3 Documentation v3.3*, https://twiki.cern.ch/twiki/pub/Atlas/NSWelectronics/vmm.pdf, CERN, Jun 2017
5. C. Bakalis, NSW Electronics Weekly Meeting, *VMM Configuration/Readout Firmware*, https://indico.cern.ch/event/671377/contributions/2746180/attachments/1536788/2408121/fw_bakalis_nsw_elx_171006.pdf,CERN ,Oct 2017
6. D. J. Antrim, NSW Electronics Weekly Meeting, *VERSO DAQ & VMM Calibration*, https://indico.cern.ch/event/671377/contributions/2746181/attachments/1536796/2407884/2017_10_6_verso_and_calibration_overviewPDF.pdf, CERN, Oct 2017

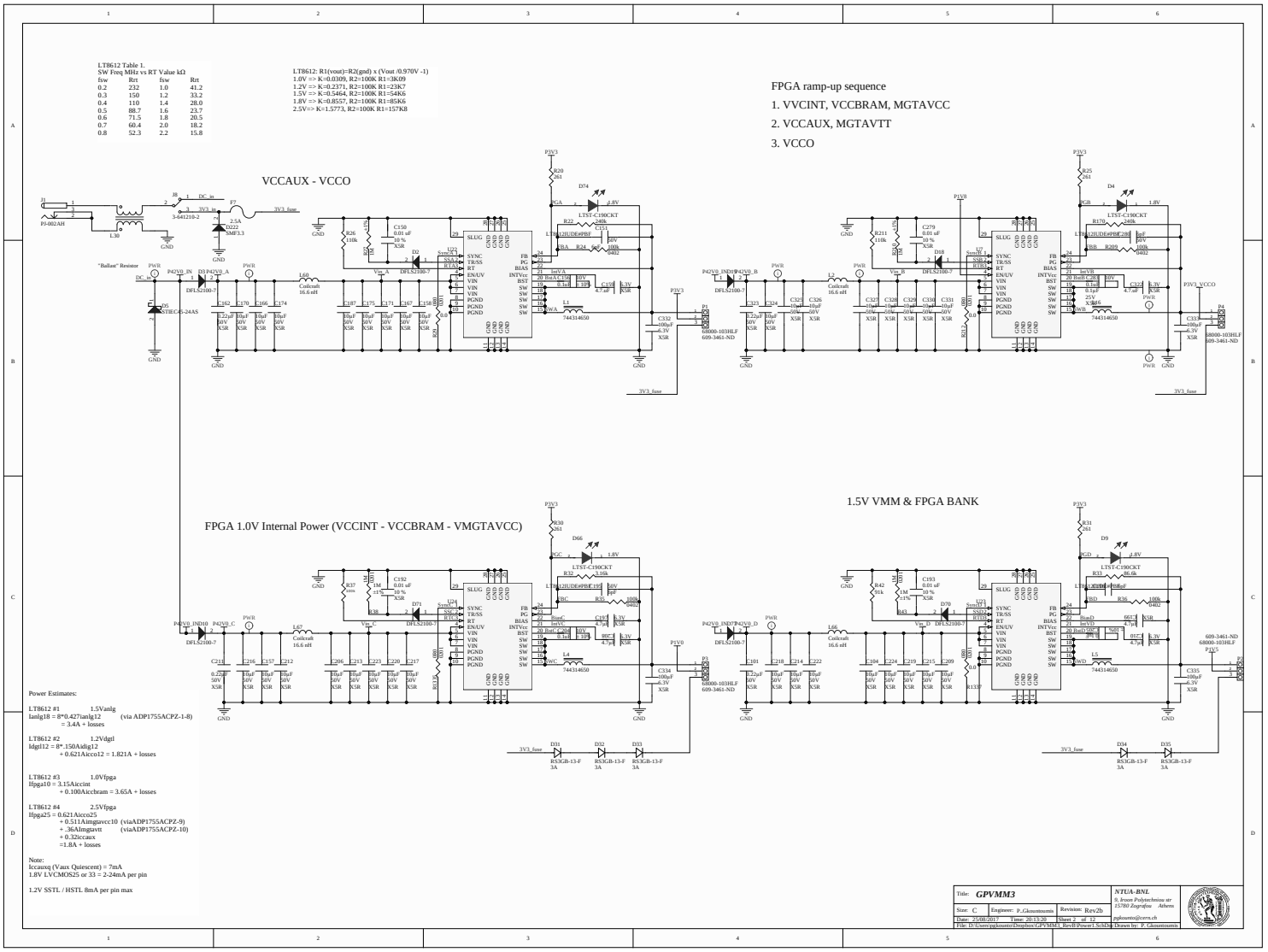
Appendix C

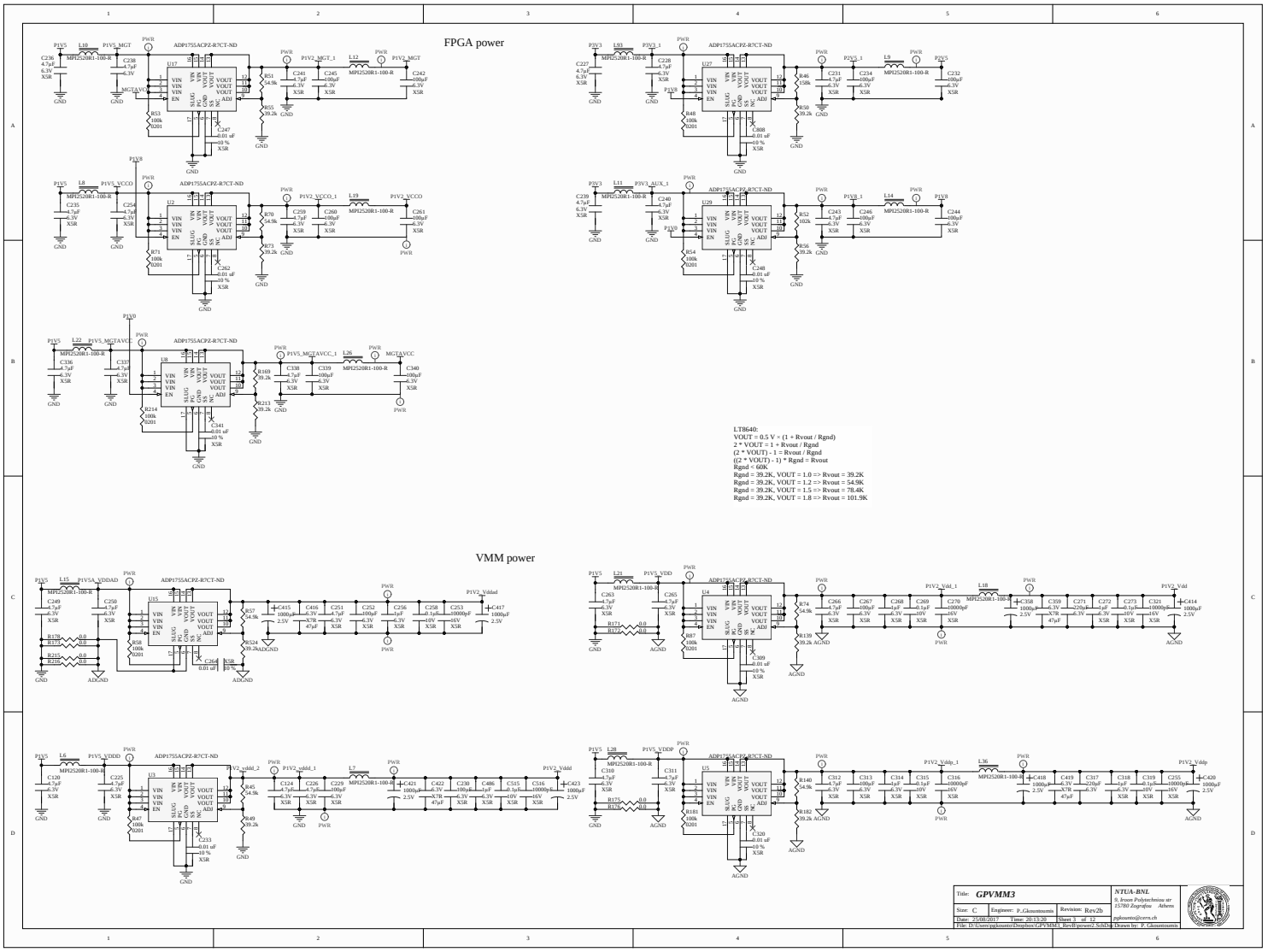
Communication

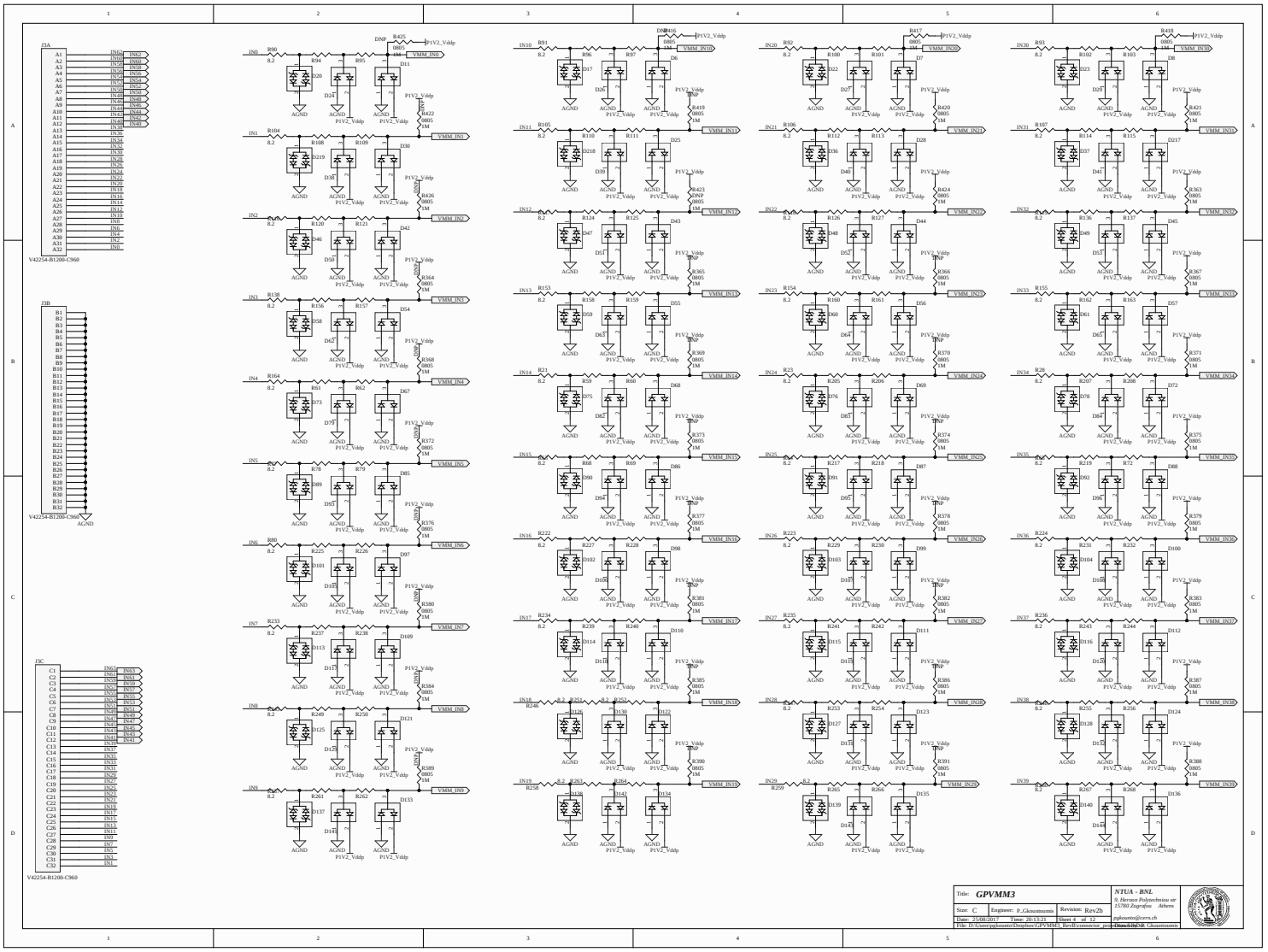
Board : Panagiotis Gkountoumis <panagiotis.gkountoumis@cern.ch>
Sofware : Daniel Joseph Antrim <daniel.joseph.antrim@cern.ch>,
Polyneikis Tzanis <polyneikis.tzanis@cern.ch>
Firmware : Christos Bakalis <christos.bakalis@cern.ch>,
Paris Moschovakos <paris.moschovakos@cern.ch>

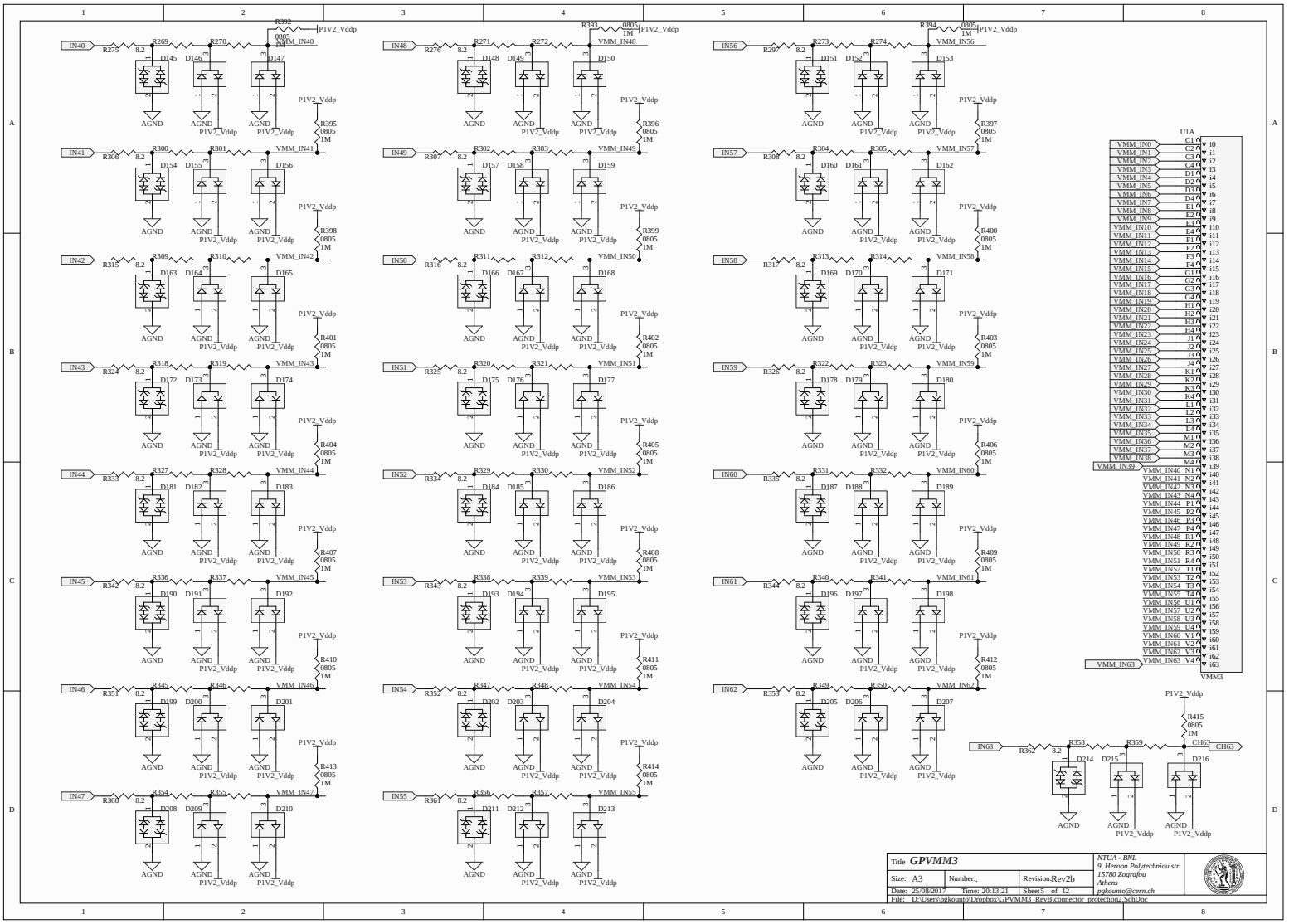
Appendix D

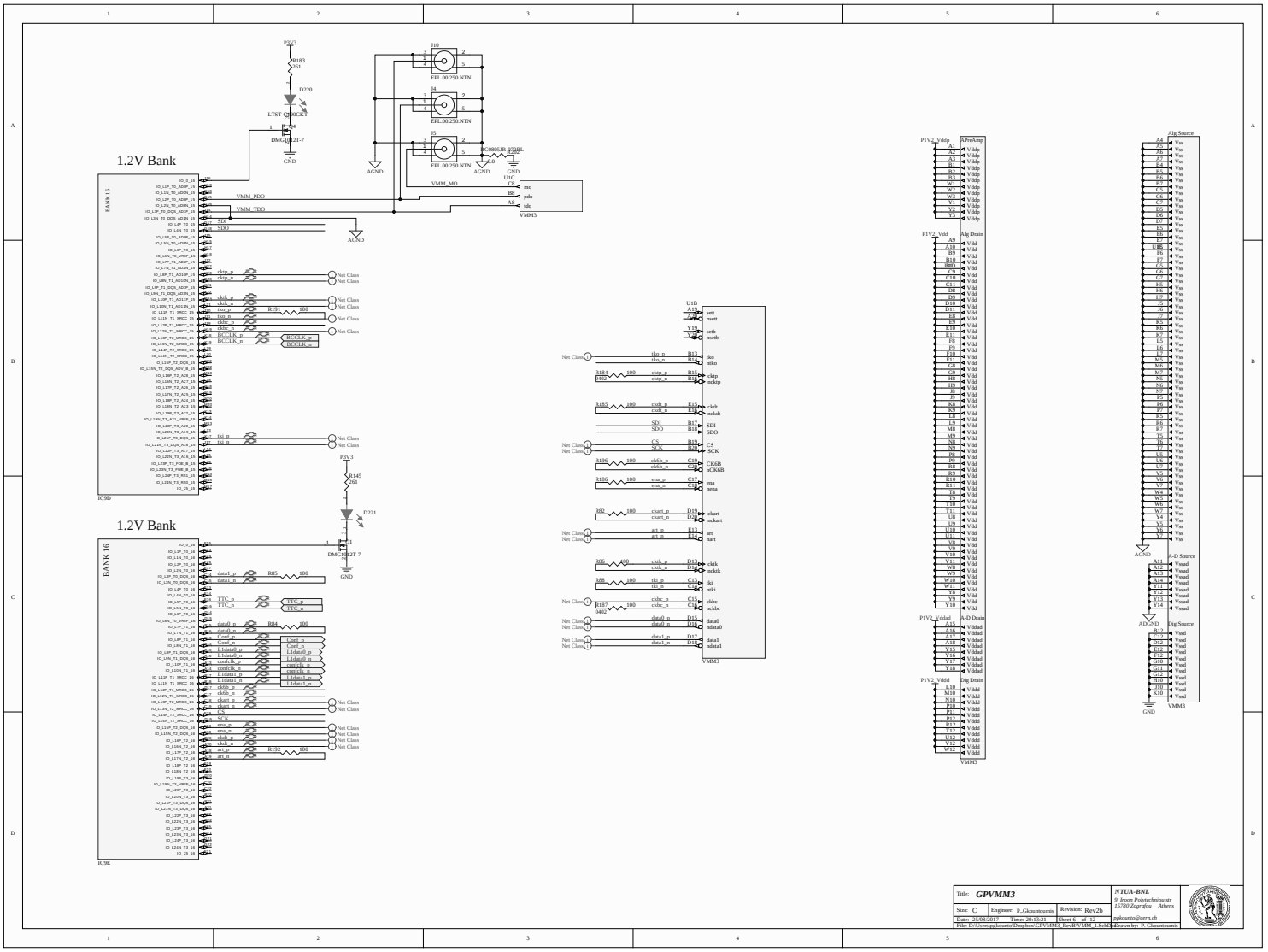
GPVMM3 Schematics

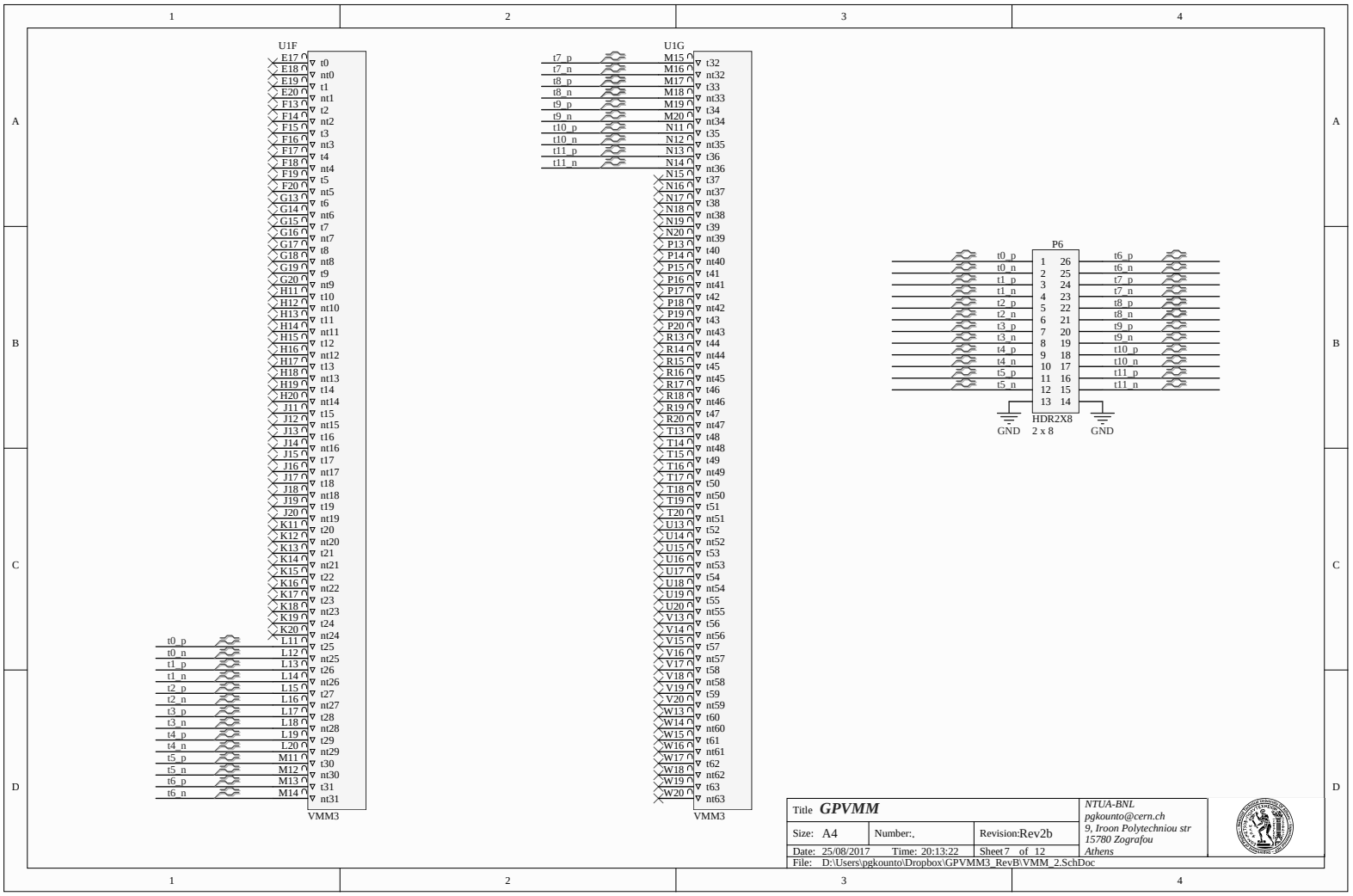

LT8612 Table 1.
SW Freq MHz vs RT Value kΩ
fsw Rrt fsw Rrt
0.2 232 1.0 41.2
0.3 150 1.2 33.2
0.4 110 1.4 28.0
0.5 88.7 1.6 23.7
0.6 71.5 1.8 20.5
0.7 60.4 2.0 18.2
0.8 52.3 2.2 15.8
LT8612: R1(vout)=R2(gnd) x (Vout /0.970V -1)
1.0V => K=0.0309, R2=100K R1=3K09
1.2V => K=0.2371, R2=100K R1=23K7
1.5V => K=0.5464, R2=100K R1=54K6
1.8V => K=0.8557, R2=100K R1=85K6
2.5V=> K=1.5773, R2=100K R1=157K8
FPGA ramp-up sequence
1. VVCINT, VCCBRAM, MGTAVCC
2. VCCAUX, MGTAVTT
3. VCCO
VCCAUX - VCCO
"Ballast" Resistor
FPGA 1.0V Internal Power (VCCINT - VCCBRAM - VMGTAVCC)
1.5V VMM & FPGA BANK
Power Estimates:
LT8612 #1 1.5Vanlg
Ianlg18 = 8*0.427Ianlg12 (via ADP1755ACPZ-1-8)
+ 3.4A + losses
LT8612 #2 1.2Vdgtl
Idgtl12 = 8*.150Aidlg12
+ 0.621Aicco12 = 1.821A + losses
LT8612 #3 1.0Vfpga
Ifpga10 = 3.15Aiccint
+ 0.100Aiccbram = 3.65A + losses
LT8612 #4 2.5Vfpga
Ifpga25 = 0.621Aicco25
+ 0.511Aimgtavcc10 (viaADP1755ACPZ-9)
+ .36Aimgtavtt (viaADP1755ACPZ-10)
+ 0.32iccaux
=1.8A + losses
Note:
Iccauxq (Vaux Quiescent) = 7mA
1.8V LVCMOS25 or 33 = 2-24mA per pin
1.2V SSTL / HSTL 8mA per pin max
Title: GPVMM3
Size: C
Engineer: P_Gkountoumis
Revision: Rev2b
Date: 25/08/2017 Time: 20:13:20
Sheet 2 of 12
File: D:\Users\pgkounto\Dropbox\GPVMM3_Rev0\Power1.SchDoc
Drawn by: P_Gkountoumis
NTUA-BNL
9, Iroon Polytechniou str
15780 Zografou Athens
pgkounto@cern.ch

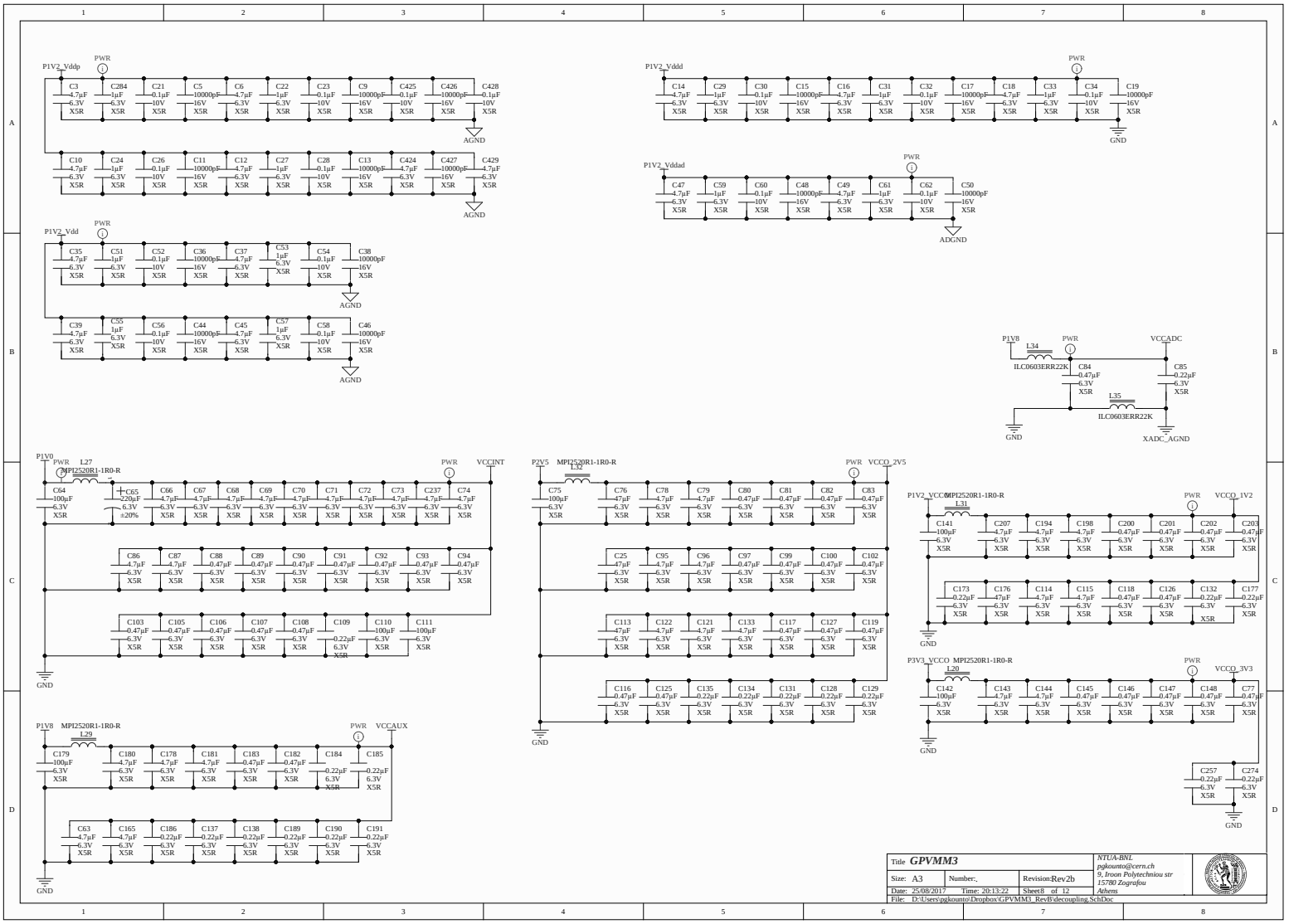

FPGA power
VMM power
LT8640:
VOUT = 0.5 V × (1 + Rvout / Rgnd)
2 * VOUT = 1 + Rvout / Rgnd
(2 * VOUT) - 1 = Rvout / Rgnd
((2 * VOUT) - 1) * Rgnd = Rvout
Rgnd < 60K
Rgnd = 39.2K, VOUT = 1.0 => Rvout = 39.2K
Rgnd = 39.2K, VOUT = 1.2 => Rvout = 54.9K
Rgnd = 39.2K, VOUT = 1.5 => Rvout = 78.4K
Rgnd = 39.2K, VOUT = 1.8 => Rvout = 101.9K
Title: GPVMM3
Size: C
Engineer: P_Gkountoumis
Revision: Rev2b
Date: 25/08/2017
Time: 20:13:20
Sheet 3 of 12
File: D:\Users\pgkoumto\Dropbox\GPVMM3_Rev\Rev2\3.SchDoc
Drawn by: P_Gkountoumis
NTUA-BNL
9, Iroon Polytechniou str
15780 Zografou Athens
pgkoumto@cern.ch

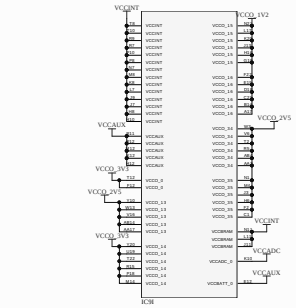

J3A
V42254-B1200-C960
J3B
V42254-B1200-C960
J3C
V42254-B1200-C960
P1V2_Vddp
AGND
VMM_IN0
VMM_IN1
VMM_IN2
VMM_IN3
VMM_IN4
VMM_IN5
VMM_IN6
VMM_IN7
VMM_IN8
VMM_IN9
VMM_IN10
VMM_IN11
VMM_IN12
VMM_IN13
VMM_IN14
VMM_IN15
VMM_IN16
VMM_IN17
VMM_IN18
VMM_IN19
VMM_IN20
VMM_IN21
VMM_IN22
VMM_IN23
VMM_IN24
VMM_IN25
VMM_IN26
VMM_IN27
VMM_IN28
VMM_IN29
VMM_IN30
VMM_IN31
VMM_IN32
VMM_IN33
VMM_IN34
VMM_IN35
VMM_IN36
VMM_IN37
VMM_IN38
VMM_IN39
Title: GPVMM3
Size: C
Engineer: P_Gkountoumis
Revision: Rev2b
Date: 25/08/2017
Time: 20:13:21
Sheet 4 of 12
File: D:\Users\pgkountz\Dropbox\GPVMM3_Rev0\connector_pcb.SchDoc
NTUA - BNL
9, Heroon Polytechniou str
15780 Zografou Athens
pgkountz@cern.ch
Drawn By: PGK

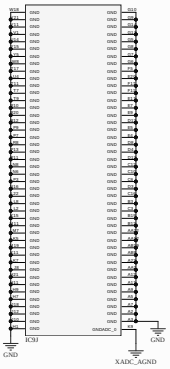

IN40
IN41
IN42
IN43
IN44
IN45
IN46
IN47
IN48
IN49
IN50
IN51
IN52
IN53
IN54
IN55
IN56
IN57
IN58
IN59
IN60
IN61
IN62
IN63
P1V2_Vddp
AGND
U1A
VMM3
Title GPVMM3
Size: A3
Number:
Revision:Rev2b
Date: 25/08/2017
Time: 20:13:21
Sheet5 of 12
File: D:\Users\pgkounto\Dropbox\GPVMM3_RevB\connector_protection2.SchDoc
NTUA - BNL
9, Heroon Polytechniou str
15780 Zografou
Athens
pgkounto@cern.ch

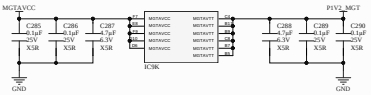

1.2V Bank
1.2V Bank
VMM_PDO
VMM_TDO
VMM_MO
SDO
SCK
CS
SDI
SDO
BCCLK_p
BCCLK_n
VMM3
IC9D
IC9E
GPVMM3
NTUA-BNL


U1F
VMM3
U1G
VMM3
P6
HDR2X8
2 x 8
GND
GND
Title GPVMM
Size: A4
Number:.
Revision:Rev2b
Date: 25/08/2017 Time: 20:13:22
Sheet 7 of 12
File: D:\Users\pgkounto\Dropbox\GPVMM3_RevB\VMM_2.SchDoc
NTUA-BNL
pgkounto@cern.ch
9, Iroon Polytechniou str
15780 Zografou
Athens

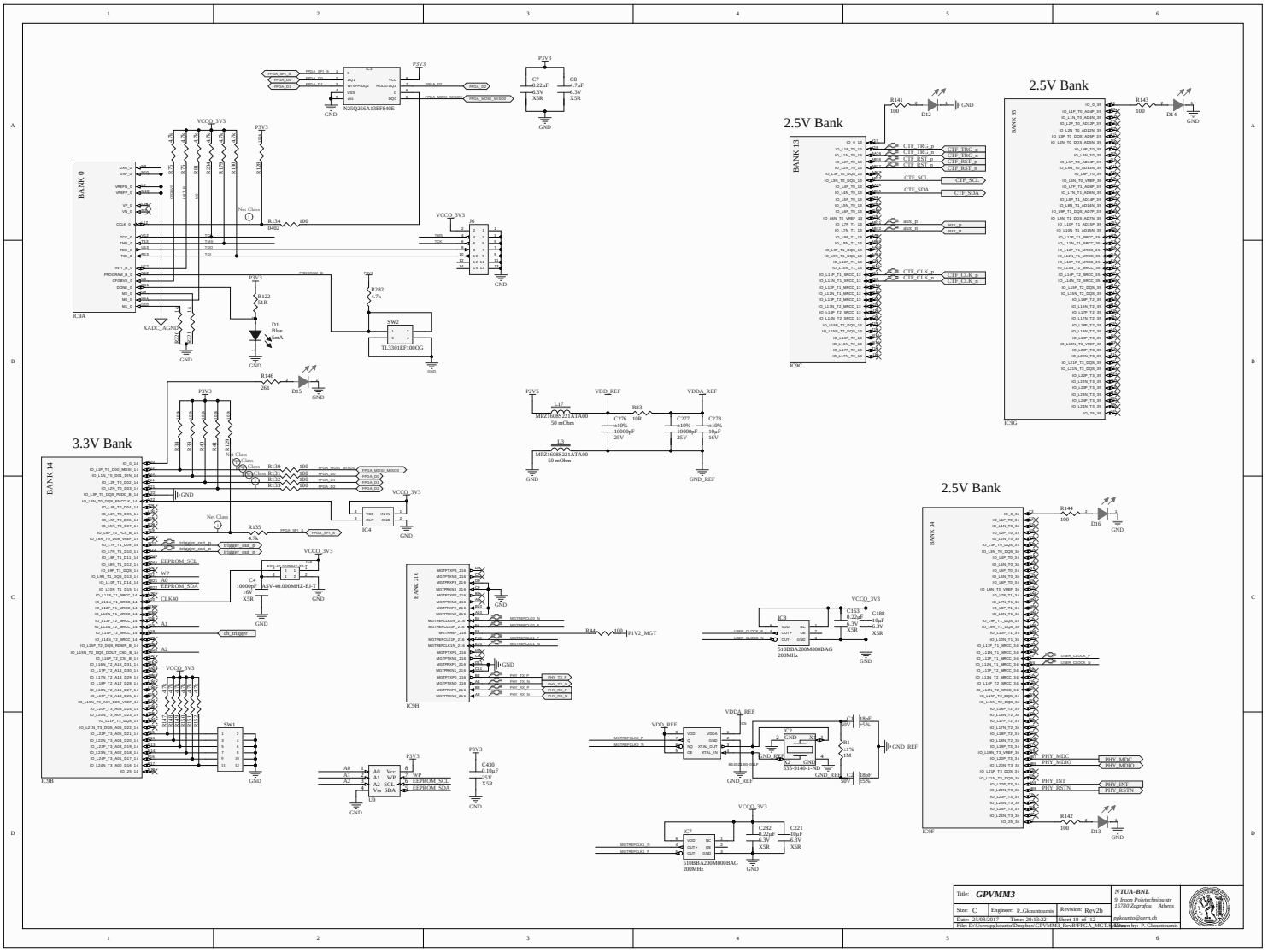

P1V2_Vddp
P1V2_Vdd
P1V2_Vddd
P1V2_Vddad
P1V8
VCCADC
ILC0603ERR22K
XADC_AGND
P1V0
VCCINT
P2V5
VCCO_2V5
VCCO_1V2
P3V3_VCCO
VCCO_3V3
P1V8
VCCAUX
MPI2520R1-1R0-R
GND
AGND
ADGND
PWR
Title GPVMM3
Size: A3
Number:
Revision:Rev2b
Date: 25/08/2017
Time: 20:13:22
Sheet8 of 12
File: D:\Users\pgkounto\Dropbox\GPVMM3_RevB\decoupling.SchDoc
NTUA-BNL
pgkounto@cern.ch
9, Iroon Polytechniou str
15780 Zografou
Athens

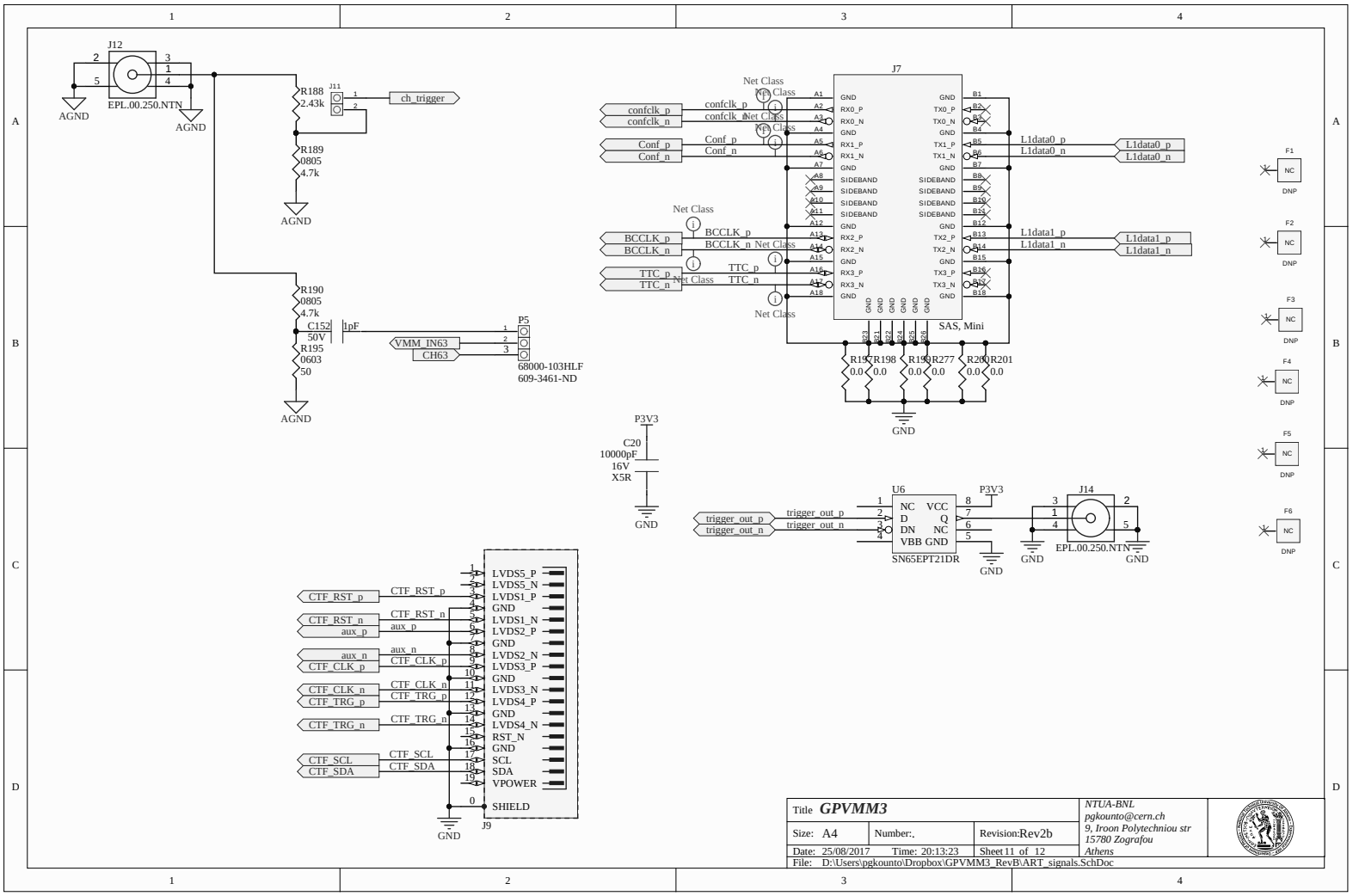




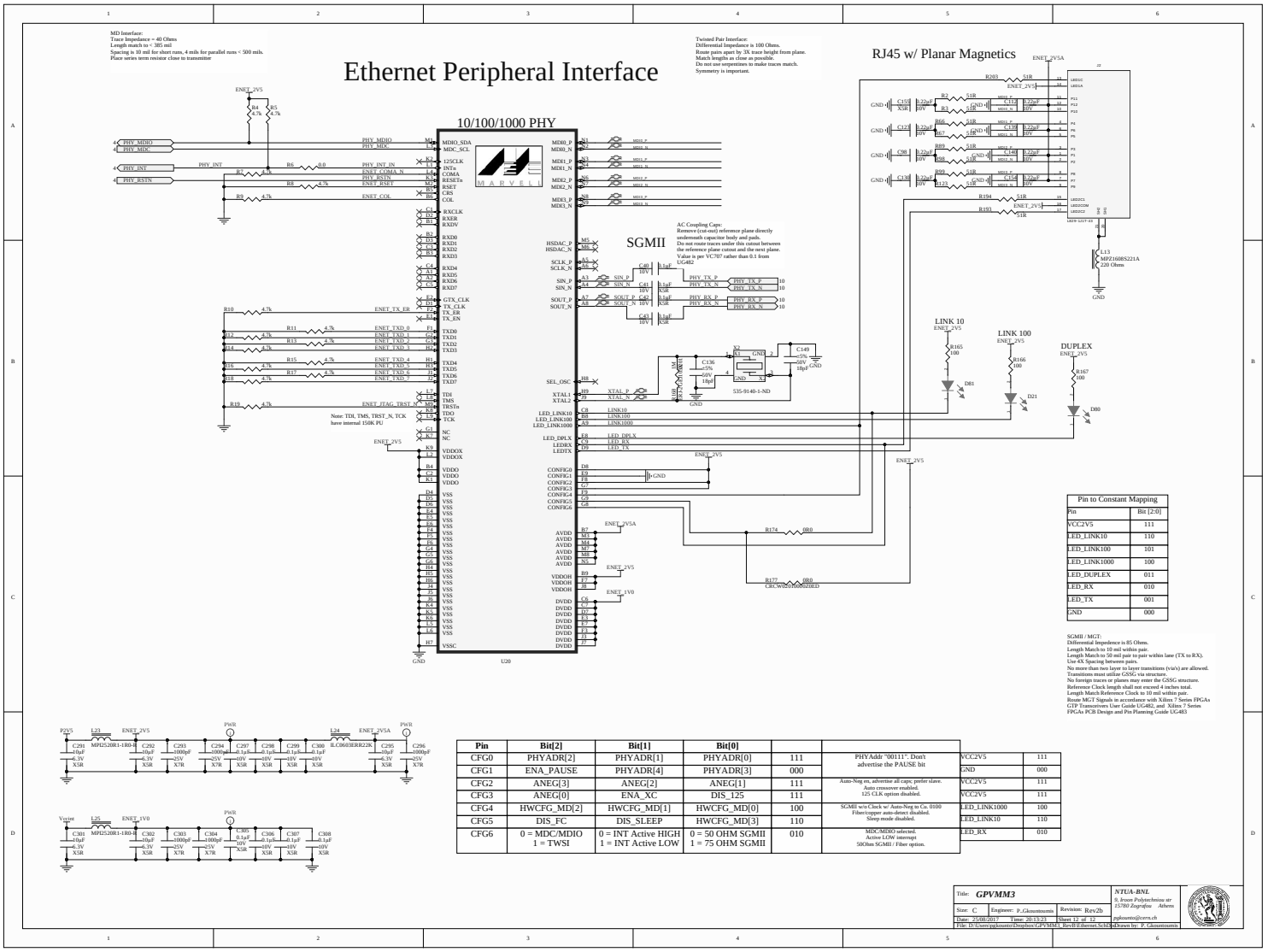


Title: *GPVMM3*			*NTUA-BNL*
Size: C	Engineer: P_Gkountoumis	Revision: Rev2b	*9, Iroon Polytechniou str*
Date: 25/08/2017	Time: 20:13:22	Sheet 9 of 12	*15780 Zografou Athens*
File: D:\Users\pgkount\Dropbox\GPVMM3_Rev0\FPGA_newest_2.SchDoc		Drawn by: P_Gkountoumis	*pgkounto@cern.ch*


2.5V Bank
2.5V Bank
2.5V Bank
3.3V Bank
BANK 0
BANK 13
BANK 85
BANK 14
BANK 34
BANK 216
IC3A
IC3C
IC3G
IC3B
IC3F
IC3H
N25Q256A13EF840E
XADC_AGND
TL3301EF100QG
LFE2M50SE-5FN484C
ASV-40.000MHZ-EJ-T
24LC64
510BBA200M000BAG
200MHz
SI5-9140-I-GND
PHY_MDC
PHY_MDIO
PHY_INT
PHY_RSTN
CTF_TRG_p
CTF_TRG_n
CTF_RST_p
CTF_RST_n
CTF_SCL
CTF_SDA
CTF_CLK_p
CTF_CLK_n
trigger_out_p
trigger_out_n
rb_trigger
EEPROM_SCL
EEPROM_SDA
CLK40
VCCO_3V3
P3V3
VDD_REF
VDDA_REF
GND_REF
GND
Title: GPVMM3
Size: C
Engineer: P_Gkountoumis
Revision: Rev2b
Date: 25/08/2017
Time: 20:13:22
Sheet 10 of 12
File: D:\Users\pgkountou\Dropbox\GPVMM3_Rev\FPGA_MGT_3.SchDoc
Drawn by: P_Gkountoumis
NTUA-BNL
9, Iroon Polytechniou str
15780 Zografou Athens
pgkountou@cern.ch



Title	***GPVMM3***		*NTUA-BNL*
Size: A4	Number:.	Revision:Rev2b	*pgkounto@cern.ch*
Date: 25/08/2017	Time: 20:13:23	Sheet 11 of 12	*9, Iroon Polytechniou str*
File: D:\Users\pgkounto\Dropbox\GPVMM3_RevB\ART_signals.SchDoc			*15780 Zografou*
			Athens

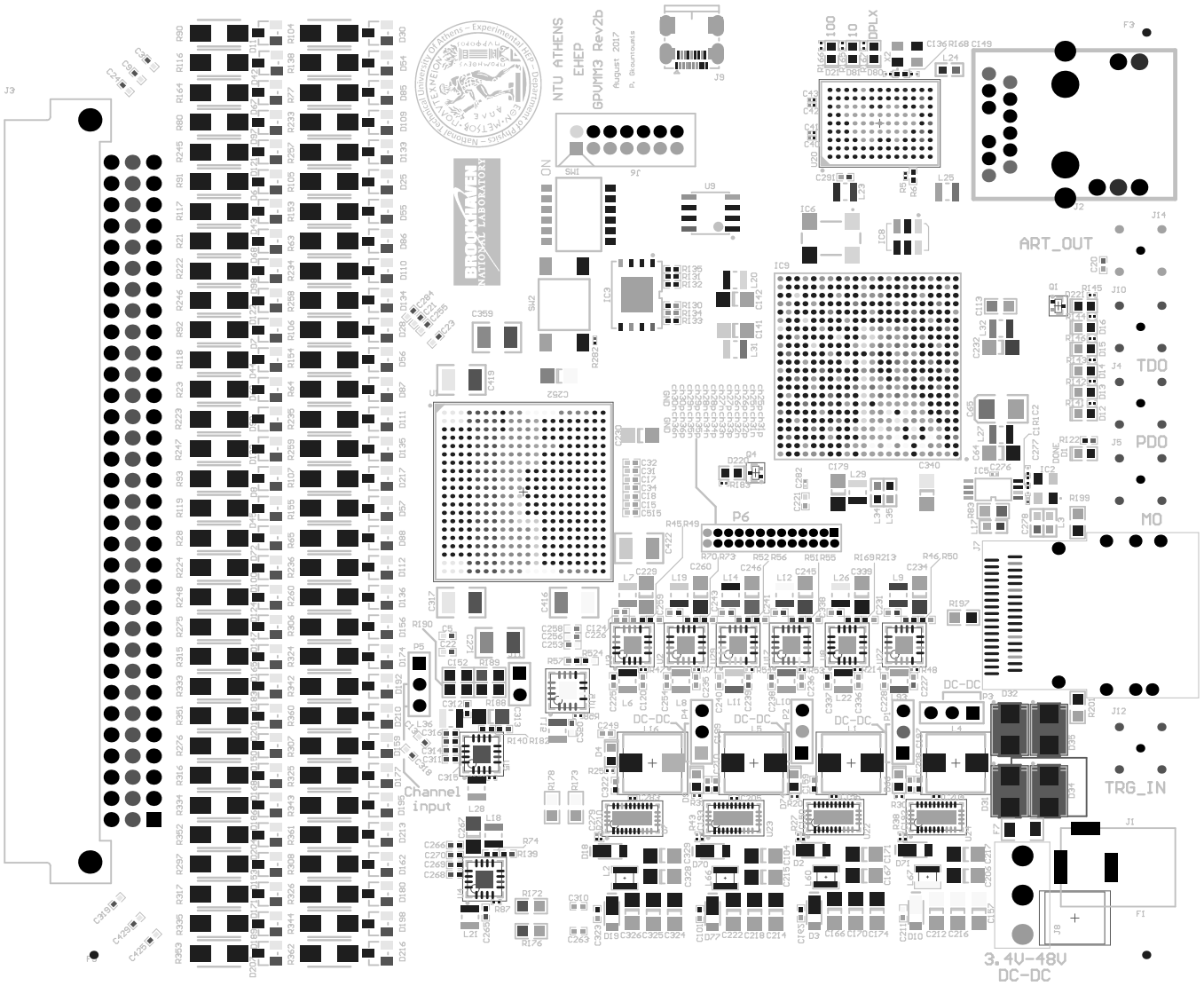


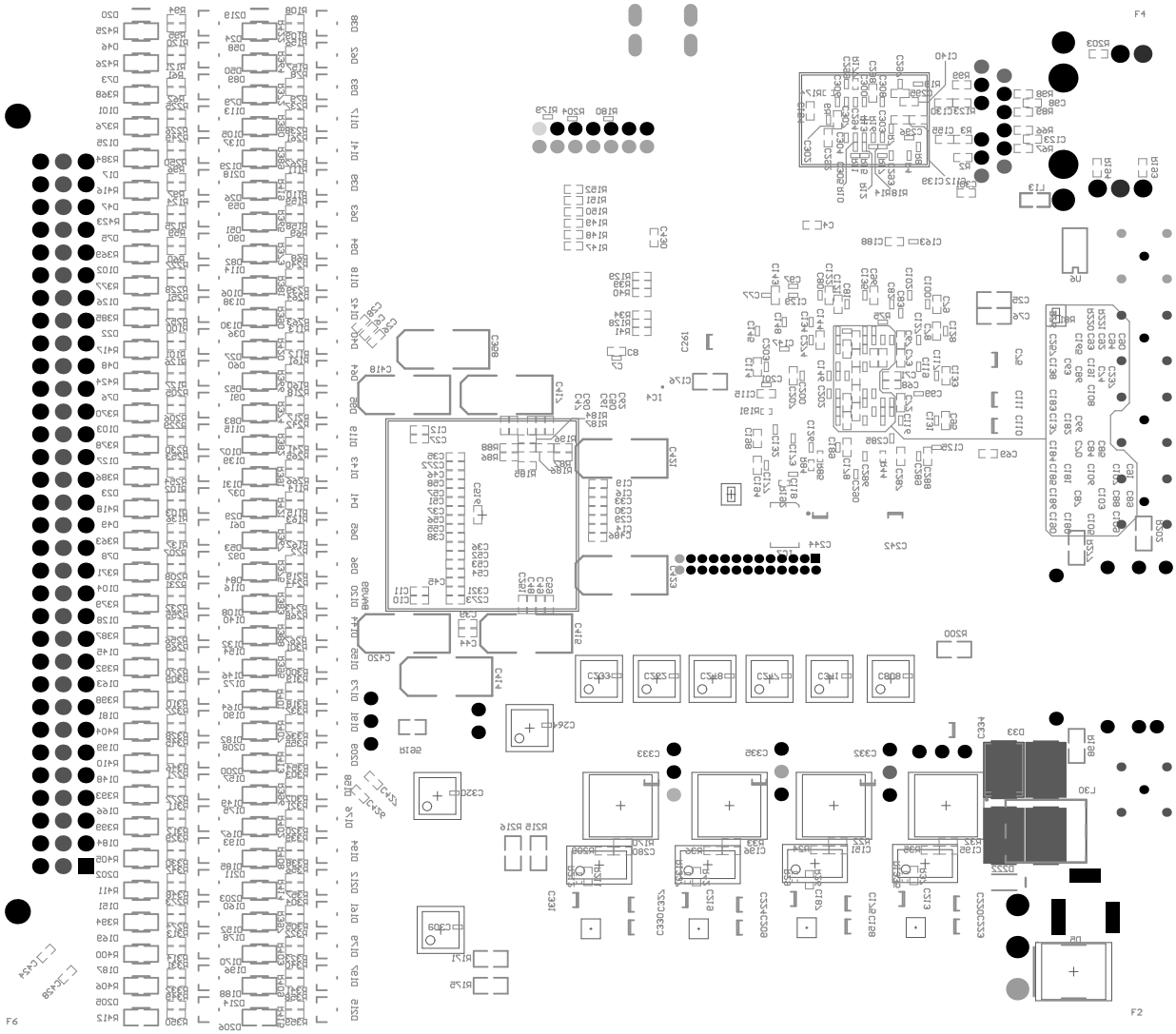
Pin to Constant Mapping	
Pin	Bit [2:0]
VCC2V5	111
LED_LINK10	110
LED_LINK100	101
LED_LINK1000	100
LED_DUPLEX	011
LED_RX	010
LED_TX	001
GND	000

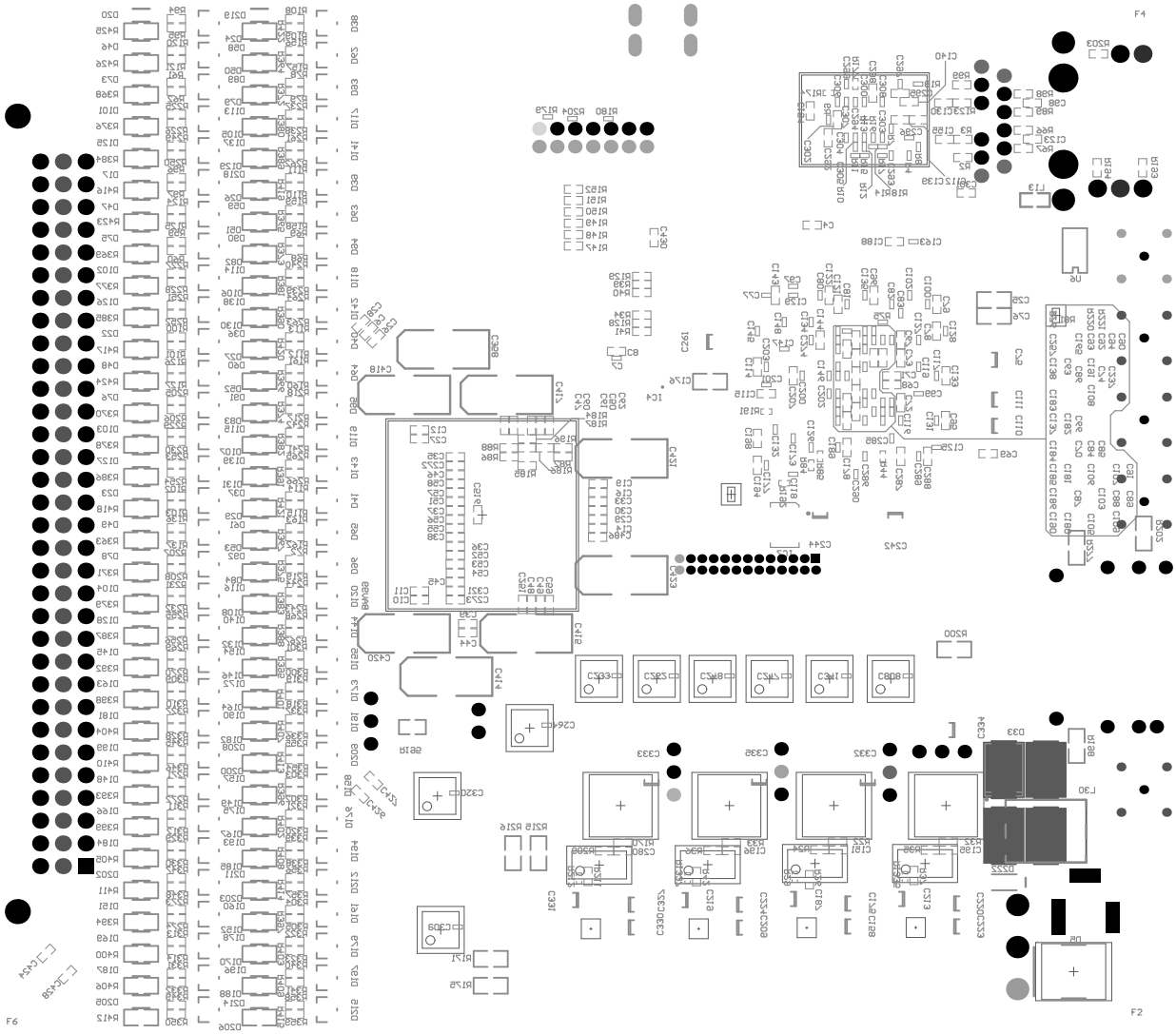
Pin	Bit[2]	Bit[1]	Bit[0]		
CFG0	PHYADR[2]	PHYADR[1]	PHYADR[0]	111	PHYAddr "00111". Don't advertise the PAUSE bit
CFG1	ENA_PAUSE	PHYADR[4]	PHYADR[3]	000	
CFG2	ANEG[3]	ANEG[2]	ANEG[1]	111	Auto-Neg on, advertise all caps; prefer slave. Auto crossover enabled. 125 CLK option disabled.
CFG3	ANEG[0]	ENA_XC	DIS_125	111	
CFG4	HWCFG_MD[2]	HWCFG_MD[1]	HWCFG_MD[0]	100	SGMII w/o Clock w/ Auto-Neg to Cu. 0100 Fiber/copper auto-detect disabled. Sleep mode disabled.
CFG5	DIS_FC	DIS_SLEEP	HWCFG_MD[3]	110	
CFG6	0 = MDC/MDIO 1 = TWSI	0 = INT Active HIGH 1 = INT Active LOW	0 = 50 OHM SGMII 1 = 75 OHM SGMII	010	MDC/MDIO selected. Active LOW interrupt 50Ohm SGMII / Fiber option.

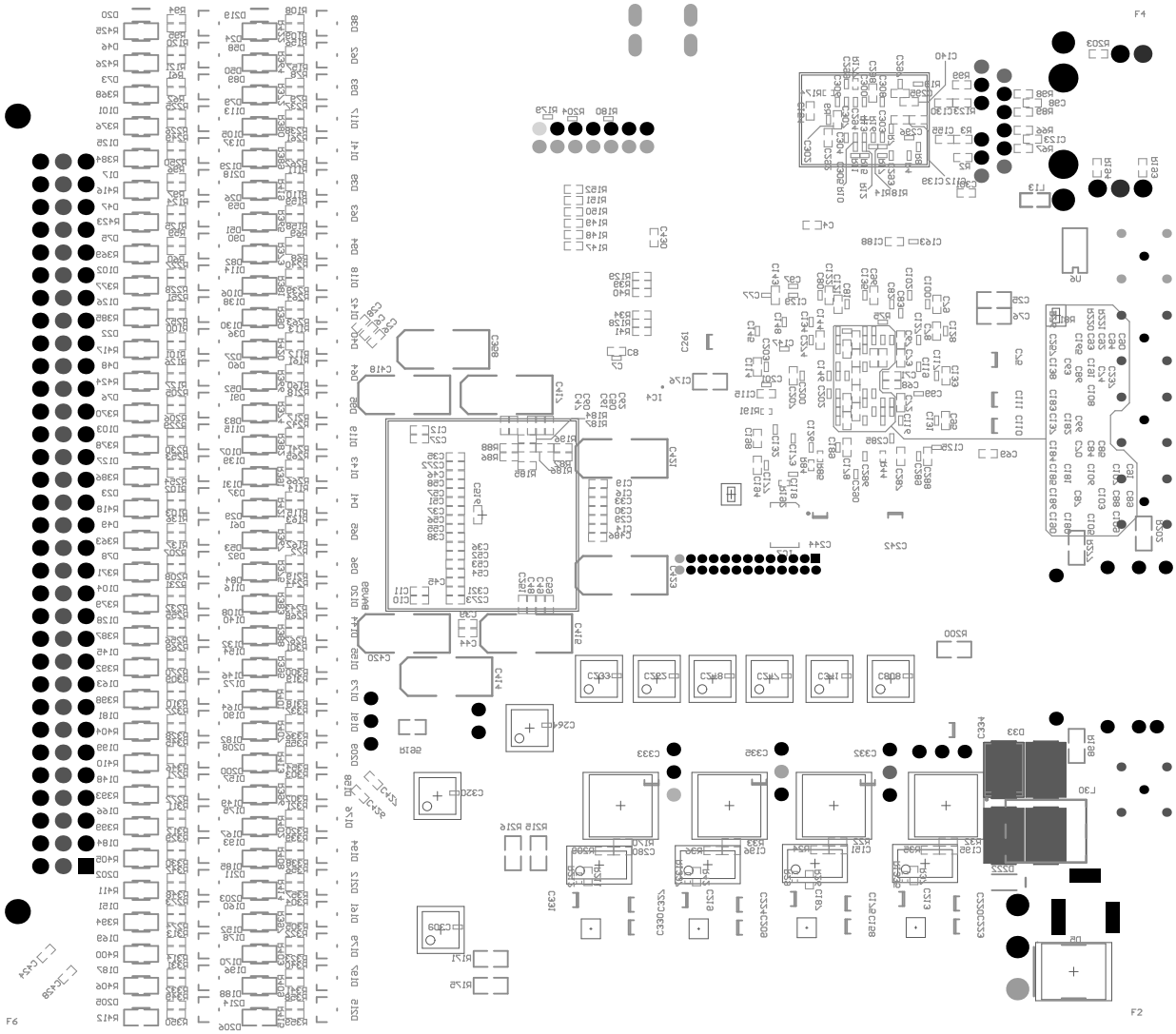
VCC2V5	111
GND	000
VCC2V5	111
VCC2V5	111
LED_LINK1000	100
LED_LINK10	110
LED_RX	010

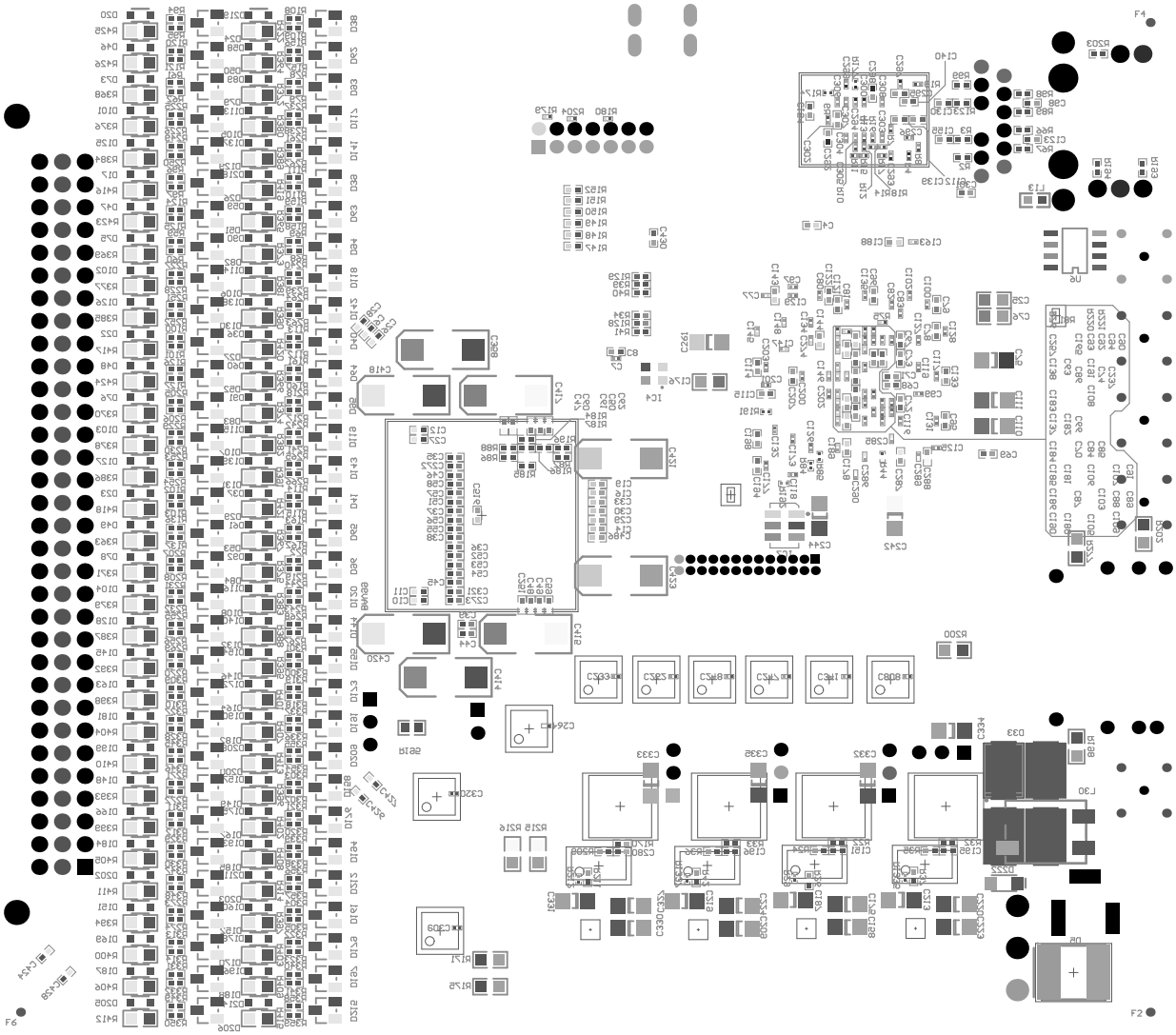
Title:	GPVMM3
Size: C	Engineer: P_Gkountoumis
Revision:	Rev2b
Date: 25/08/2017	Time: 20:13:22
Sheet 12 of 12	
File: D:\Users\pgkountoumis\Dropbox\GPVMM3_Rev\Ethernet.SchDoc	Drawn by: P_Gkountoumis
NTUA-BNL	9, Iroon Polytechniou str 15780 Zografou Athens pgkountoumis@cern.ch


NTU ATHENS
EHEP
GPVMM3 Rev2b
August 2017
P. Gkountoumis
ART_OUT
TDO
PDO
MO
TRG_IN
Channel input
3.4V-48V DC-DC
DC-DC
ON

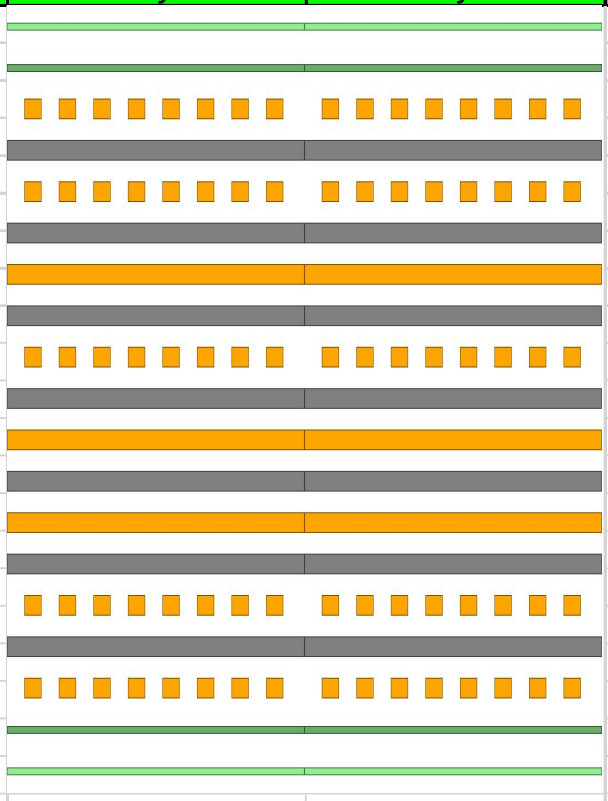








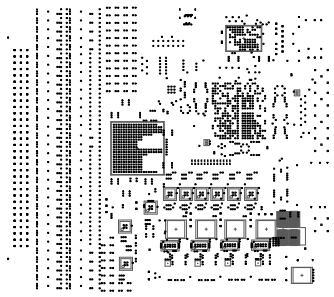
Board Stack Report

Stack Up			Layer Stack			
Layer	Board Layer Stack	Board Layer Stack	Name	Material	Thickness	Constant
1			Top Paste			
2			Top Overlay			
3			Top Solder	Solder Resist	1.00mil	3.5
4			Top Layer	Copper	1.70mil	
5			Dielectric 1		2.64mil	4.25
6			L2_Signal	Copper	0.60mil	
7			Dielectric 2		5.00mil	4.2
8			L3_PWR	Copper	0.60mil	
9			Dielectric 3		10.57mil	4.07
10			L4_Signal	Copper	0.60mil	
11			Dielectric 4		10.00mil	4.2
12			L5_GND	Copper	0.60mil	
13			Dielectric 5		10.57mil	0
14			L6_PWR	Copper	0.60mil	
15			Dielectric 6		5.00mil	4.35
16			L7_Signal	Copper	0.60mil	
17			Dielectric 7		2.64mil	4.25
18			Bottom Layer	Copper	1.70mil	
19			Bottom Solder	Solder Resist	1.00mil	3.5
20			Bottom Overlay			
21			Bottom Paste			
	Height : 55.42mil	Height : 55.42mil				

1. BOARD SHALL BE FABRICATED – PERFORMANCE CLASS II AS PER IPC-6011 AND IPC-6012.
2. VENDOR IDENTIFICATION AND DATE CODE OF MANUFACTURING SHALL BE ETCHED ON THE SOLDER SIDE OF BOARD. THE DATE CODE SHALL BE IN THE FORMAT: "WWYY" WHERE WW=WEEK AND YY=YEAR.
3. FABRICATE USING FILM "FAB/DRILL IDENT" FOR REFERENCE.
4. PERMANENTLY MARK BARE BOARD WITH TEST STAMP USING NON-CONDUCTIVE INK.
5. SILKSCREEN BOTH SIDES USING LIQUID PHOTO IMAGING (LPI), USE NON-CONDUCTIVE INK, NOT ALLOWED ON COMPONENT PADS, MOUNTING HOLES, OR VIAS.(COLOR = WHITE)
6. MATERIAL: PER IPC-4101A/24/26/29/99, COPPER CLAD, HIGH TEMPERATURE FR4 CLASS EPOXY GLASS RATED UL94V-0, 1 OZ COPPER FOR EXTERNAL LAYERS AND 0.5 OZ COPPER FOR INTERNAL LAYERS. MUST SURVIVE ASSEMBLY MAX REFLOW OF 260 DEG C (6 PASSES) – Td RATING: > 340 C; – Z AXIS CTE < 3.5%; – Tg > 170 C (MIN).
7. SOLDER MASK: SMOBC PER IPC-SM-840C, CLASS T, TYPE LPI, 0.0002" MIN TO 0.0008" MAX MEASURED OVER COPPER PLATING, MUST CLEAR ALL LANDS AS INDICATED ON GERBER SOLDER MASK LAYERS, (COLOR = GREEN). PULLBACK / EXPANSION SHOULD BE SET TO 0.05 MM.
8. FINISH: ELECTRO-LESS NICKEL IMMERSION GOLD (ENIG), 2-8 MICRO INCHES GOLD OVER 150-250 MICRO INCHES NICKEL.
9. SOLDERABILITY TEST: CATEGORY 2 OF J-STD-003.
10. ALL TEST POINTS SHALL BE FREE OF SOLDERMASK AND SILKSCREEN.
11. ALL HOLE SIZES ARE AFTER PLATING.
12. VENDOR SHALL USE TEAR-DROP TRANSITIONS FOR TRACES LESS THAN 0.010" ON PLATED THROUGH HOLE COMPONENTS AND SMT CONNECTOR PADS
13. FINISHED BOARDS SHALL NOT HAVE NICKS, SCRATCHES, VOIDS, EXPOSED COPPER, POOR PLATING OR MISDRILLED HOLES.
14. TIE-BARS ON THERMAL PADS SHOULD BE 15 MILS MINIMUM WIDTH.
15. VENDOR MAY ADD COPPER THIEVING AS NEEDED TO IMPROVE MANUFACTURABILITY, THEVING TO BE 0.030" ROUND PADS AT 0.050". THEVING WILL HAVE A MINIMUM OF 0.100" CLEARANCE FROM EXISTING COPPER AND SHOULD NOT BE PLACED UNDER SURFACE MOUNT DEVICES.
16. ALL UNCONNECTED PADS ON INNER SIGNAL LAYERS MUST BE REMOVED, THEVING TO BE 0.030" ROUND PADS AT 0.050" SPACING.
17. ALL FINISHED BOARDS TO BE 100% ELECTRICALLY TESTED
18. UNLESS OTHERWISE INDICATED, ALL LINEAR TOLERANCES SHALL BE XX +/- .010 AND XXX +/- .005

ADDITIONAL NOTES:

20. CONTROL SINGLE ENDED TRACE-GND IMPEDANCE TO 50 OHMS +/- 10% ON ALL SINGLE ENDED SIGNALS: EXTERNAL Top, Bottom = 4 MILS; INTERNAL L2, L4 L7 = 4 MILS.
21. CONTROL DIFFERENTIAL IMPEDANCE TO 100 OHMS +/- 10% ON ALL DIFFERENTIAL SIGNALS: EXTERNAL Top, Bottom = 4 MILS; INTERNAL L3, L4, L7 = 4.0 MILS;
22. VENDOR SHALL PROVIDE TDR TEST COUPON AND IMPEDANCE REPORT.
23. PCB FABRICATION FACILITY SHALL PROVIDE ASSEMBLY WITH WORKING COPIES OF OUTER LAYER GERBERS FOR REVIEW AND EDIT OF SOLDER MASK OPENINGS TO REDUCE POTENTIAL SOLDERING ISSUES
24. FINAL COPIES OF WORKING FABRICATION AND ASSEMBLY DOCUMENTATION SHALL BE PROVIDED TO THE NATIONAL TECHNICAL UNIVERSITY OF ATHENS PHYSICS DEPARTMENT



Layer	Name	Material	Thickness	Constant	Board Layer Stack	Board Layer Stack
1	Top Overlay					
2	Top Solder	Solder Resist	1.00mil	3.5		
3	Top Layer	Copper	1.70mil			
4	Dielectric 1		2.64mil	4.25		
5	Inner 2	Copper	0.60mil			
6	Dielectric 2		5.10mil	4.2		
7	Inner 3	Copper	0.60mil			
8	Dielectric 3		10.57mil	4.07		
9	Inner 4	Copper	0.60mil			
10	Dielectric 4		10.00mil	4.2		
11	Inner 5	Copper	0.60mil			
12	Dielectric 5		10.57mil	0		
13	Inner 6	Copper	0.60mil			
14	Dielectric 6		5.10mil	4.35		
15	Inner 7	Copper	0.60mil			
16	Dielectric 7		2.64mil	4.25		
17	Bottom Layer	Copper	1.70mil			
18	Bottom Solder	Solder Resist	1.00mil	3.5		
19	Bottom Overlay					

Symbol	Count	Hole Size	Plated	Hole Type	Drill Layer Pair	Via/Pad	Pad Shape	Template	Description	Hole Tolerance (+)	Hole Tolerance (-)
✖	1	20.00mil (0.508mm)	PTH	Round	Top Layer - Bottom Layer	Via	Rounded	v74d81e66			
▽	1	39.37mil (1.000mm)	PTH	Rectangle	Top Layer - Bottom Layer	Pad	Rectangle	r150_360x100_300			
⊟	1	118.11mil (3.000mm)	PTH	Rectangle	Top Layer - Bottom Layer	Pad	Rectangle	r150_360x300_100			
⊕	1	137.80mil (3.500mm)	PTH	Rectangle	Top Layer - Bottom Layer	Pad	Rectangle	r150_402x350_100			
⊕	2	61.02mil (1.550mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c199h155			
⊕	2	61.81mil (1.570mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c250h157			
⊟	2	110.24mil (2.800mm)	NPTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c280h280			
✖	2	127.95mil (3.250mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c325h325			
⊕	3	98.06mil (1.800mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c250h180			
C	4	25.59mil (0.650mm)	PTH	Slot	Top Layer - Bottom Layer	Pad	Rounded	r150_259x65_170x100x63_268			
○	5	50.00mil (1.270mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c200h127			
▽	6	35.50mil (0.900mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	c150h80			
◇	14	39.38mil (0.940mm)	PTH	Round	Top Layer - Bottom Layer	Pad	Rounded	r(Mixed)			
E	26	23.62mil (0.600mm)	PTH	Round	Top Layer - Bottom Layer	Pad	r(Mixed)	r(Mixed)			
□	26	30.00mil (0.762mm)	PTH	Round	Top Layer - Bottom Layer	r(Mixed)	Rounded	r(Mixed)			
⊠	28	35.00mil (0.889mm)	PTH	Round	Top Layer - Bottom Layer	r(Mixed)	Rounded	r(Mixed)			
✖	96	39.37mil (1.000mm)	PTH	Round	Top Layer - Bottom Layer	Pad	r(Mixed)	r(Mixed)			
⌂	170	10.00mil (0.254mm)	PTH	Round	Top Layer - Bottom Layer	Via	Rounded	v50h25			
○	2016	8.00mil (0.203mm)	PTH	Round	Top Layer - Bottom Layer	Via	Rounded	r(Mixed)			
	2408 Total										

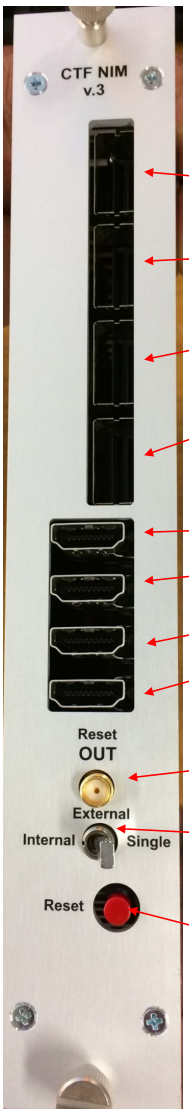
Slot definitions : Routed Path Length = Calculated from tool start centre position to tool end centre position.
Hole Length = Routed Path Length + Tool Size = Slot length as defined in the PCB layout

Appendix E

CTF Board

This appendix includes information about the CTF reference clock board designed by Givi Sekhniaidze.

Front panel



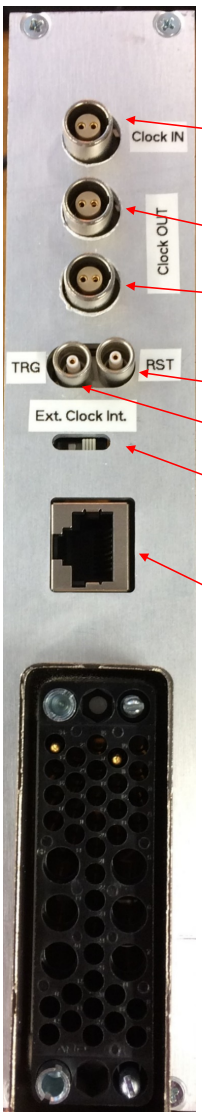
4 mini-SAS connectors
Output signal format:
CML 1.2V

4 HDMI connectors
Output signal format:
LVDS 2.5V

Reset output - CMOS

Trigger mode – Int./Ext./Single
(in this version only external and single triggers are available)

Reset button



Back panel

External clock input
3.3V LVDS

Clock outputs
3.3V LVDS

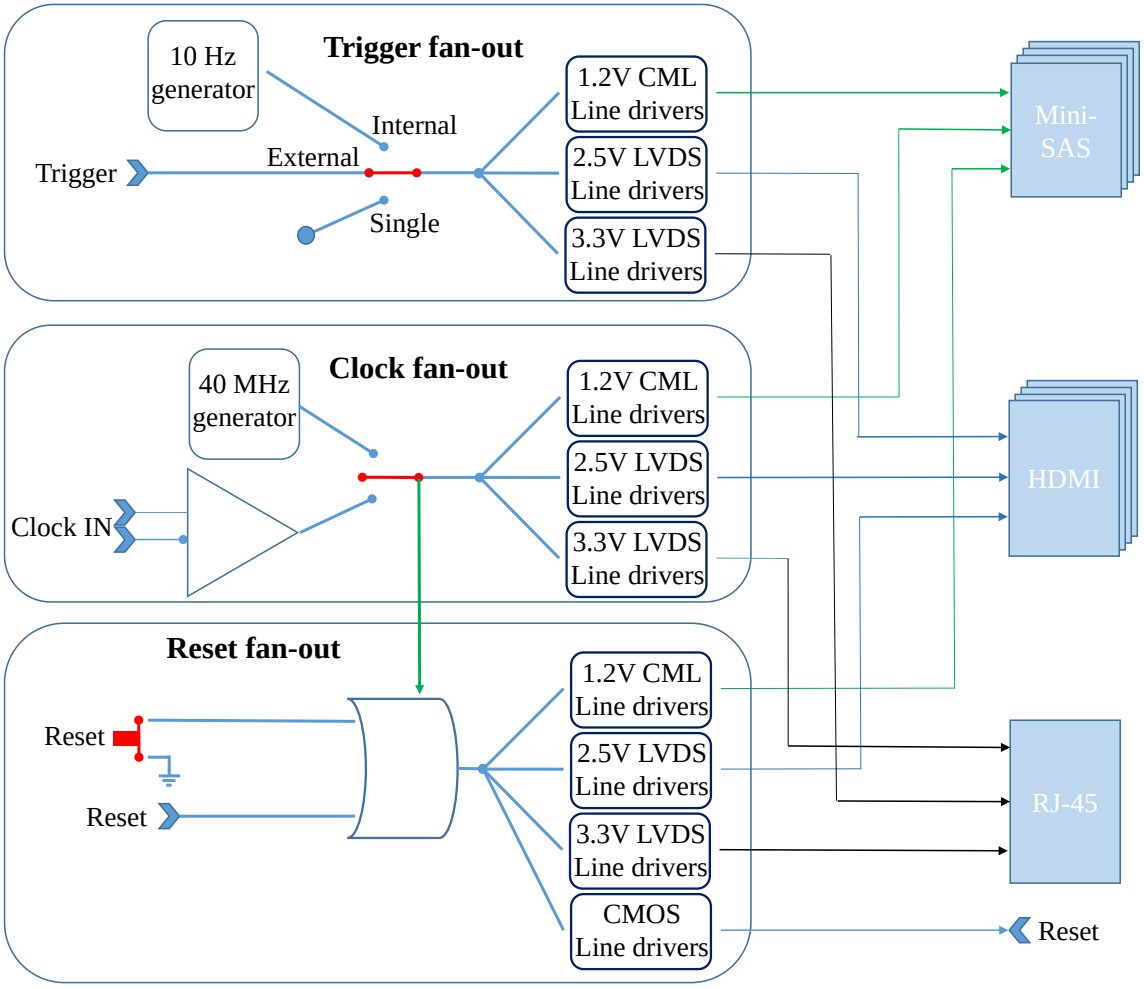
Reset input – NIM

Trigger input – NIM

Clock mode – internal/external

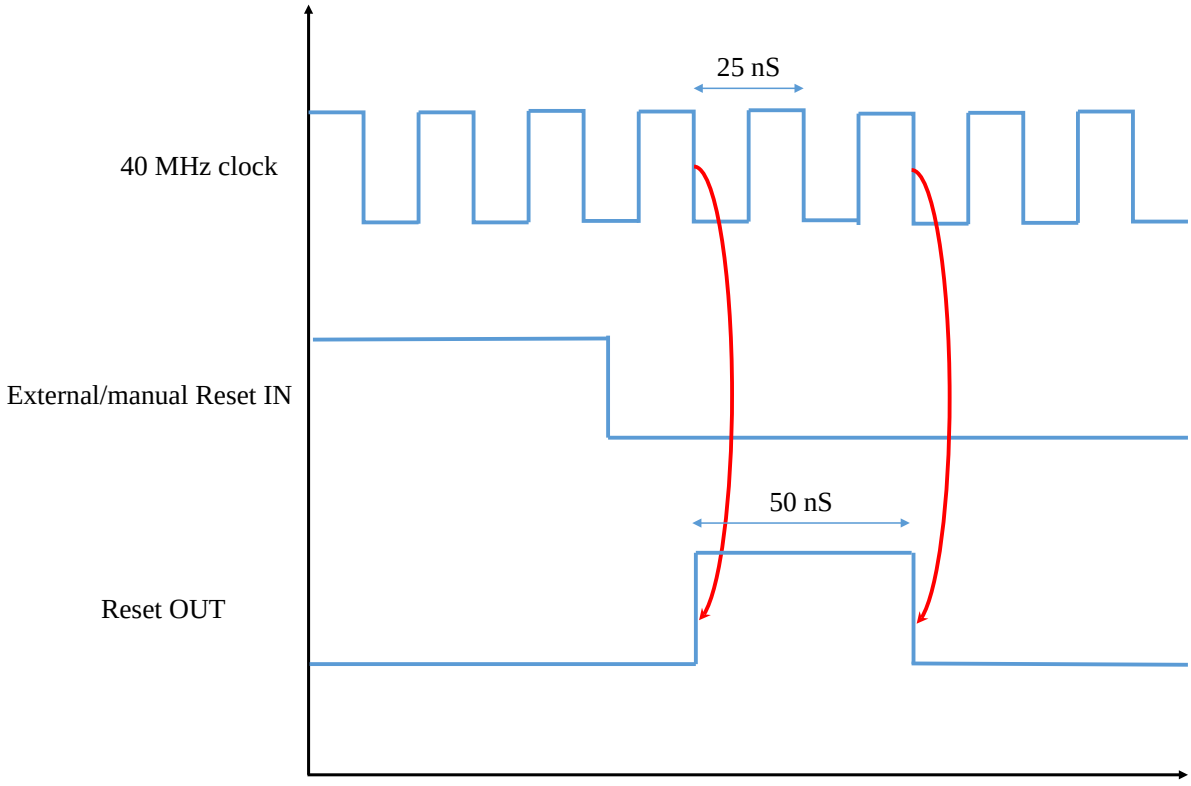
RJ45 connector
Clock, Trigger & Reset output
3.3V LVDS

Block-diagram of the board

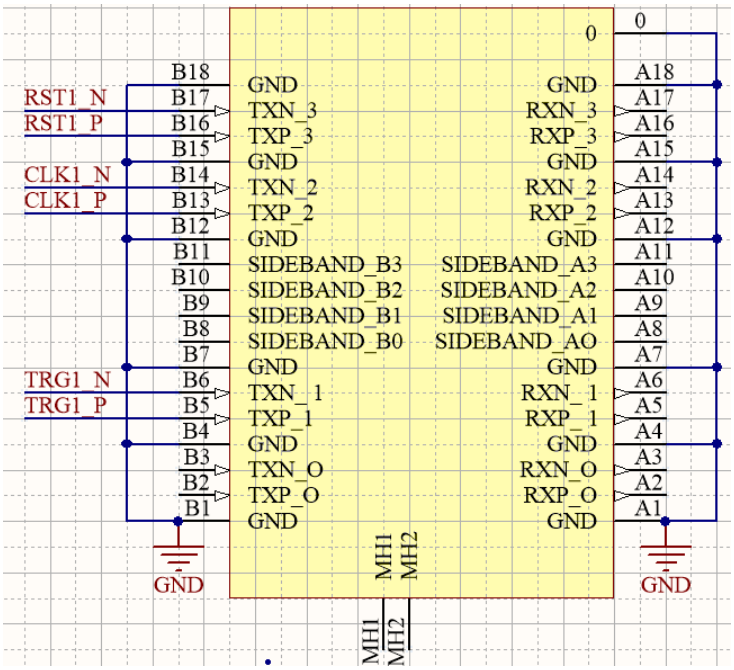


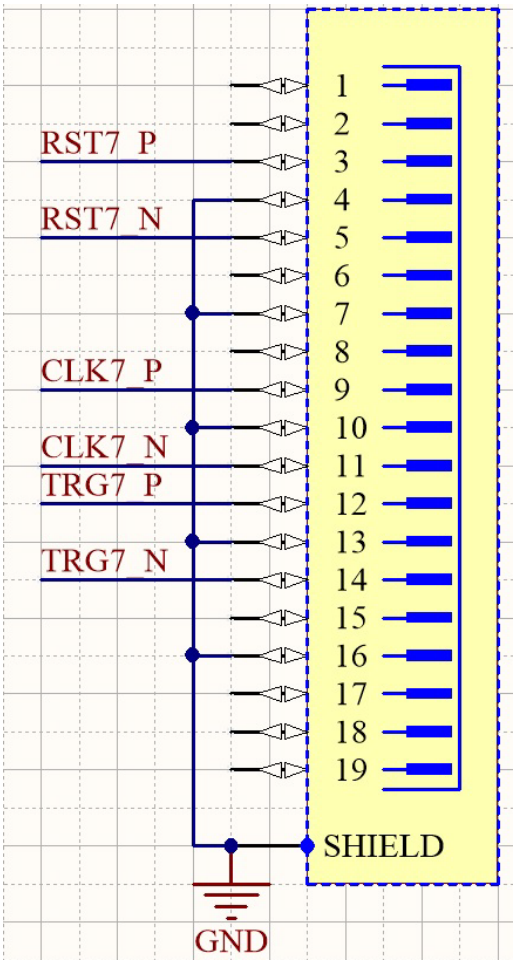
“Reset” signal time diagram

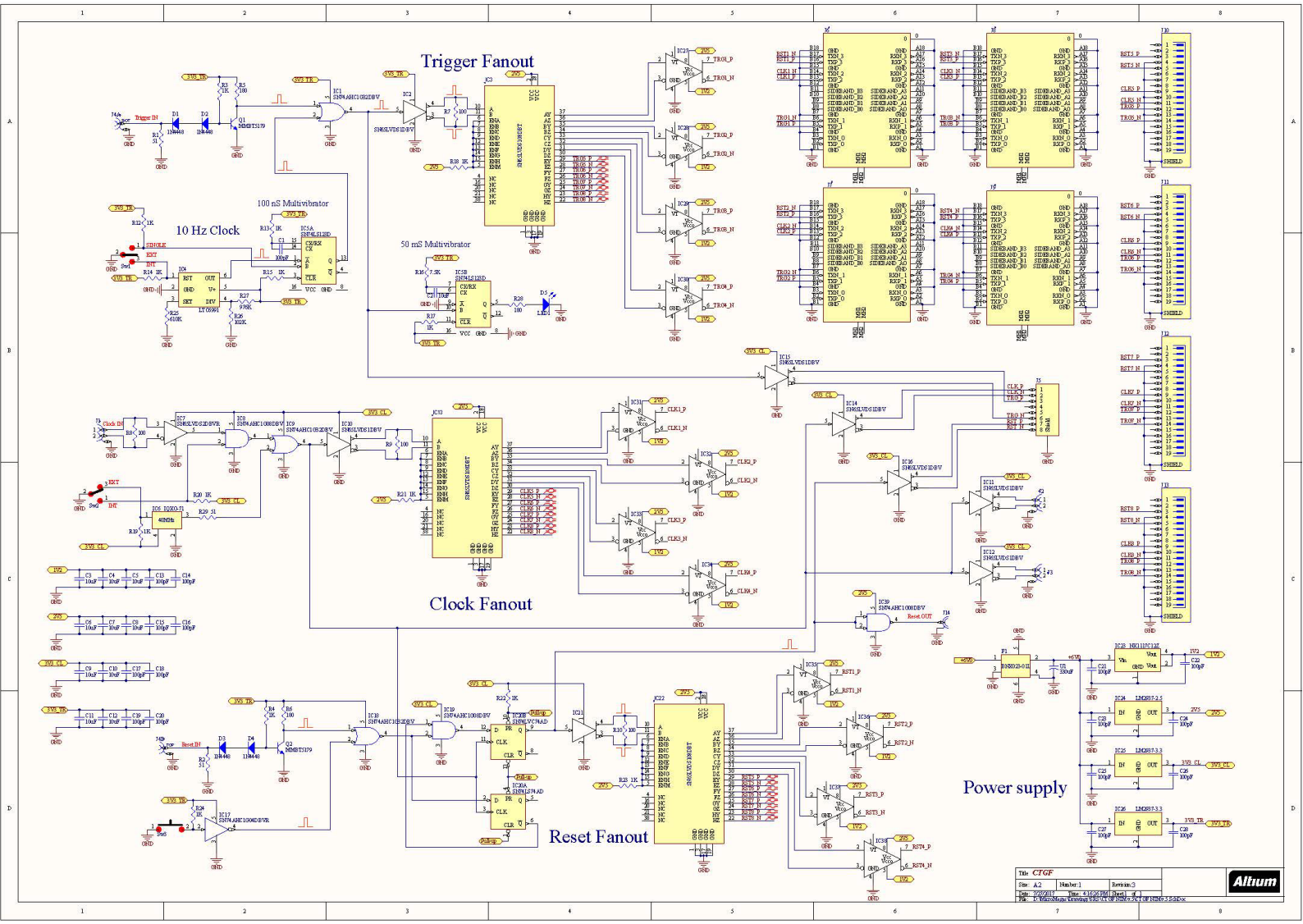
“Reset” signal is synchronizes with internal/external clock falling edge.



Mini-SAS and HDMI connectors' pin-out






Trigger Fanout
10 Hz Clock
100 nS Multivibrator
50 mS Multivibrator
Clock Fanout
Reset Fanout
Power supply
CTGF
Altium

Appendix F

HDMI Adapter for CTF Board

This appendix includes information about the HDMI Adapter for CTF board designed by NTUA.
This boards receives the LVDO HDMI signals of the CTF NIM using an HDMI type A plug connector. LVDO signals are redirected to the corresponding pins of a receptacle HDMI connector according to the standard HDMI type A pinout. The board has the same width as the CTF NIM as well as less than 10 mm height. Therefore, 4 of them can be used simultaneously for all 4 HDMI outputs.

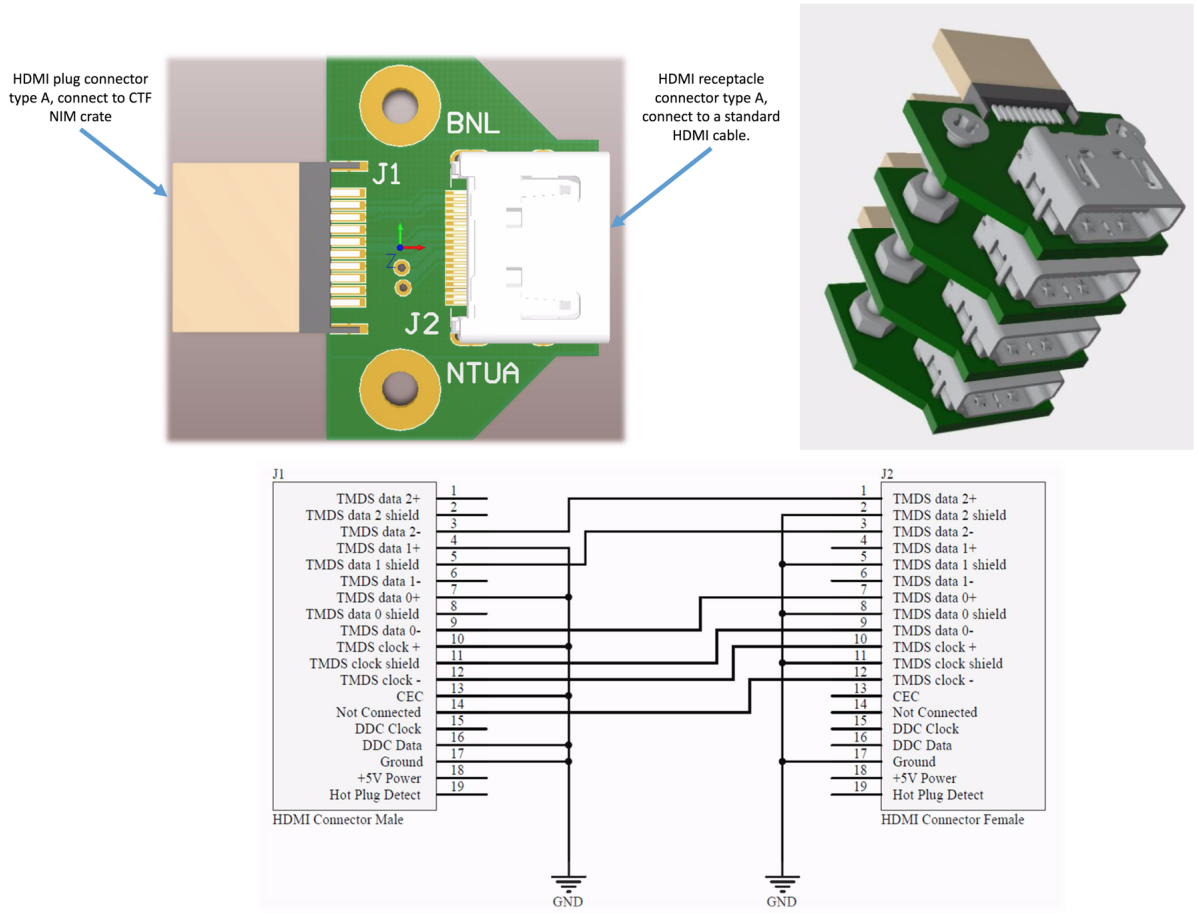


Figure F.1: HDMI Adapter for CTF Board