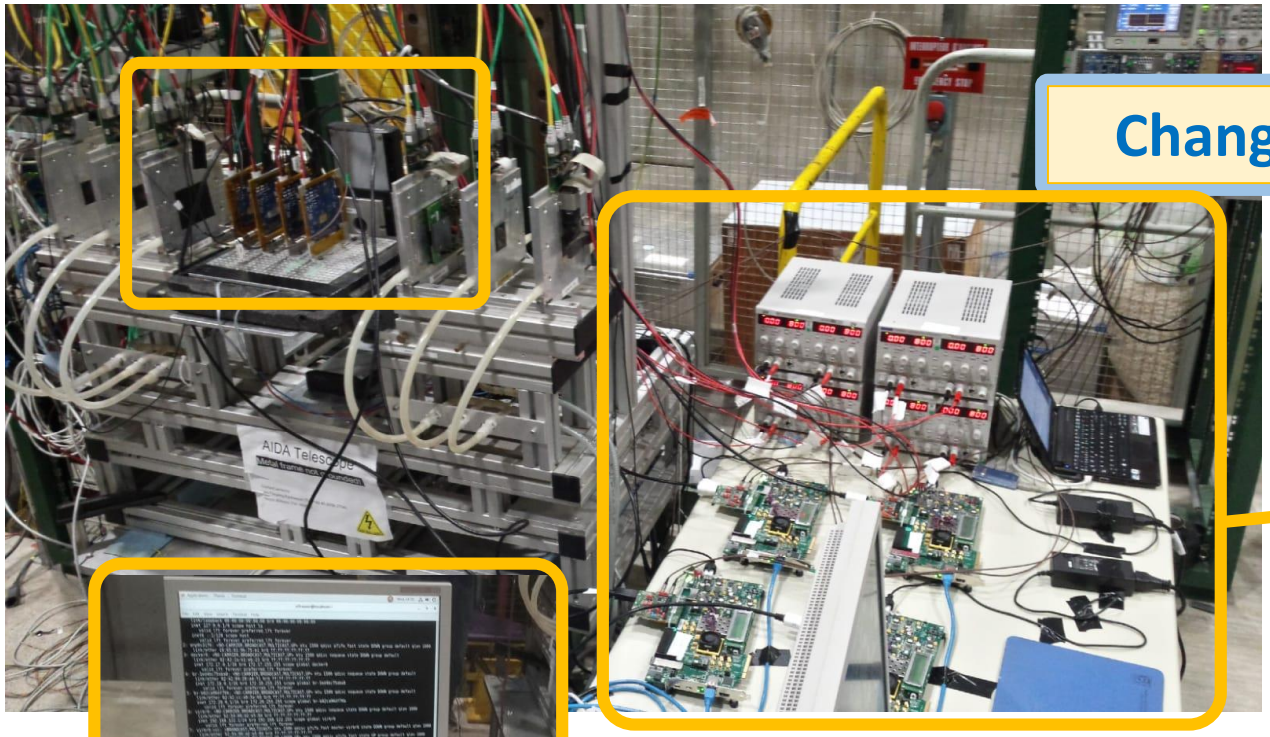
DAQ-ROC4Sens Update

J.M.Hinojo[1] , F.R. Palomo[1] , F.Muñoz[1], R.Millán[1], P.Blanco[1], A.Morales[1], I.Vila[2]
fpalomo@us.es
[1]Departamento Ingeniería Electrónica, Escuela Superior de Ingenieros
Universidad de Sevilla, Spain
[2]Instituto de Física de Cantabria
CSIC, Santander

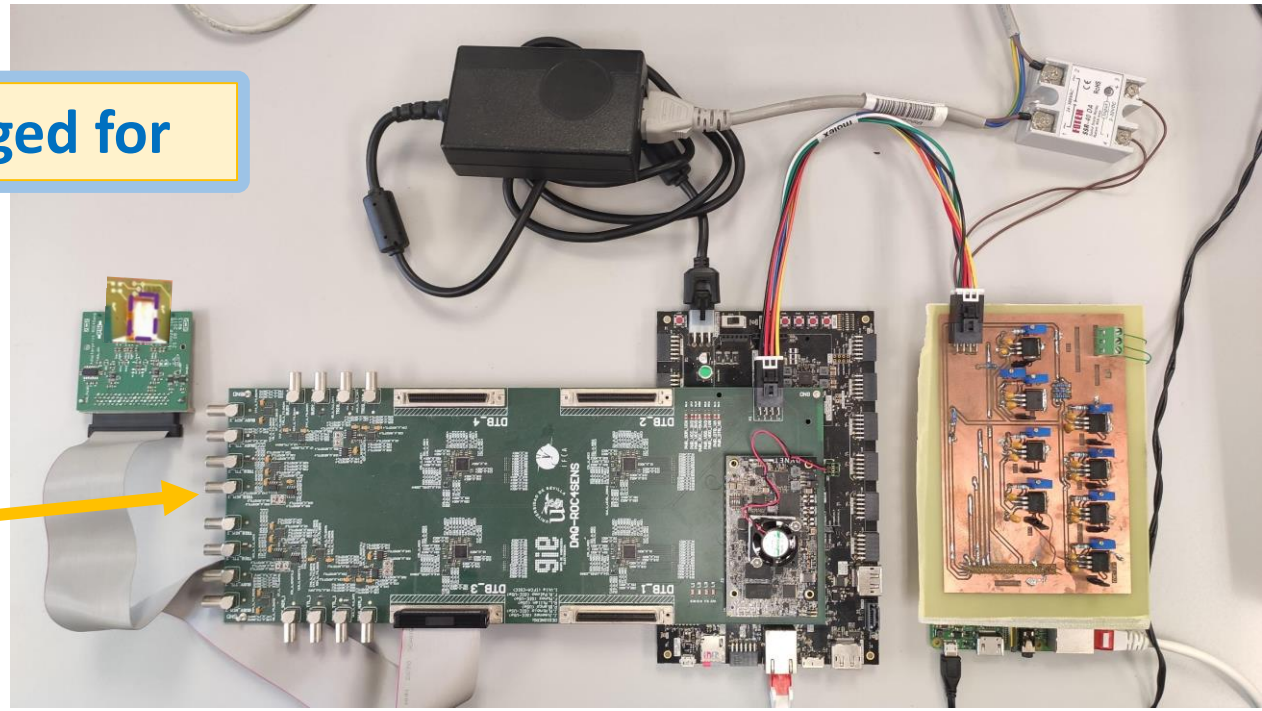
In a Nutshell

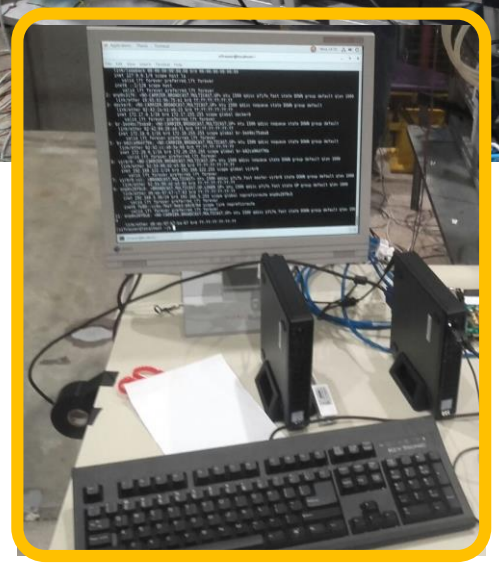
- Backend DAQ ROC4Sens
  - System Description
  - Involved Technologies
  - SoCFPGA Logical Design
  - Human Interface
  - Custom Hardware
  - State of the Project and prospectives

General Description



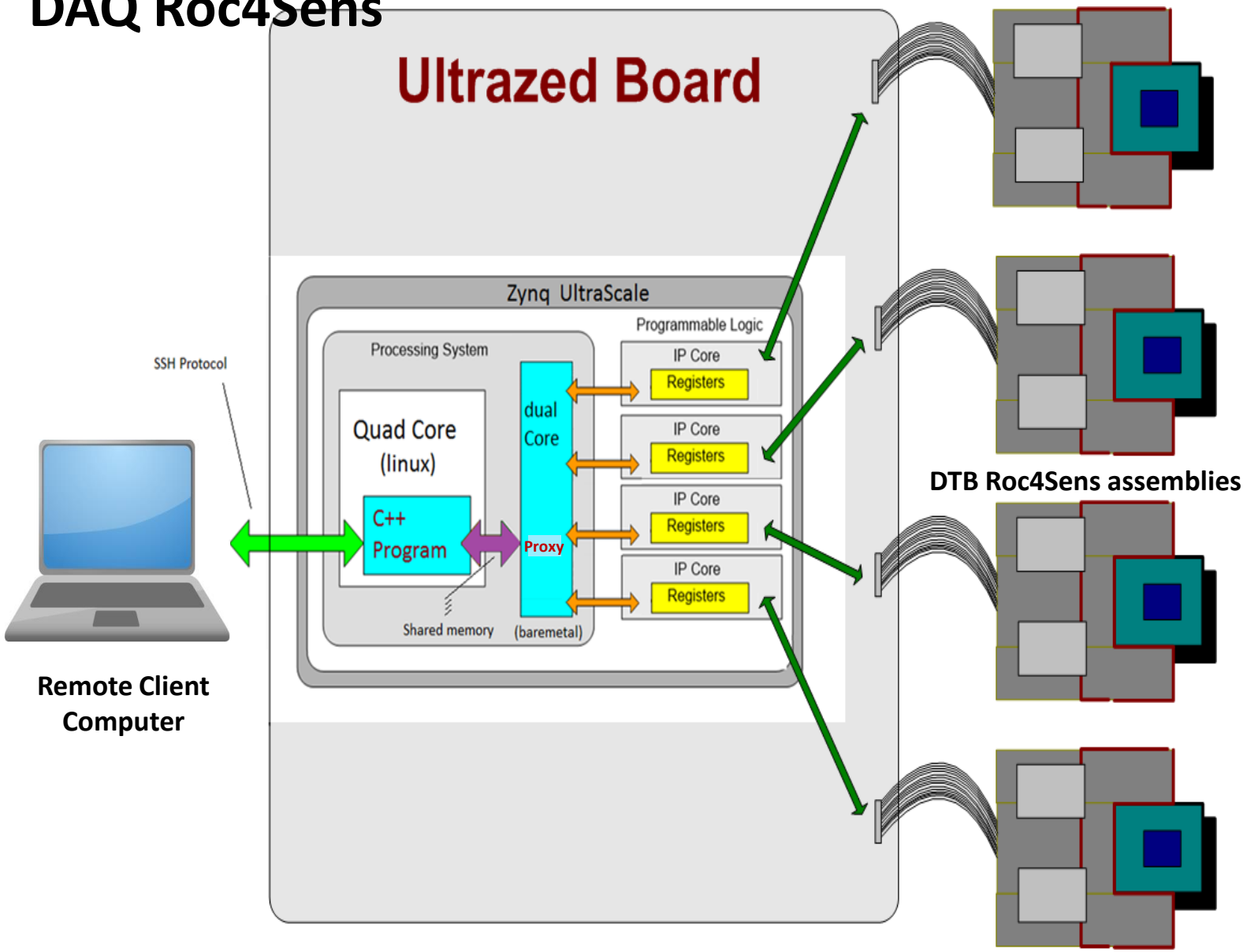
Changed for





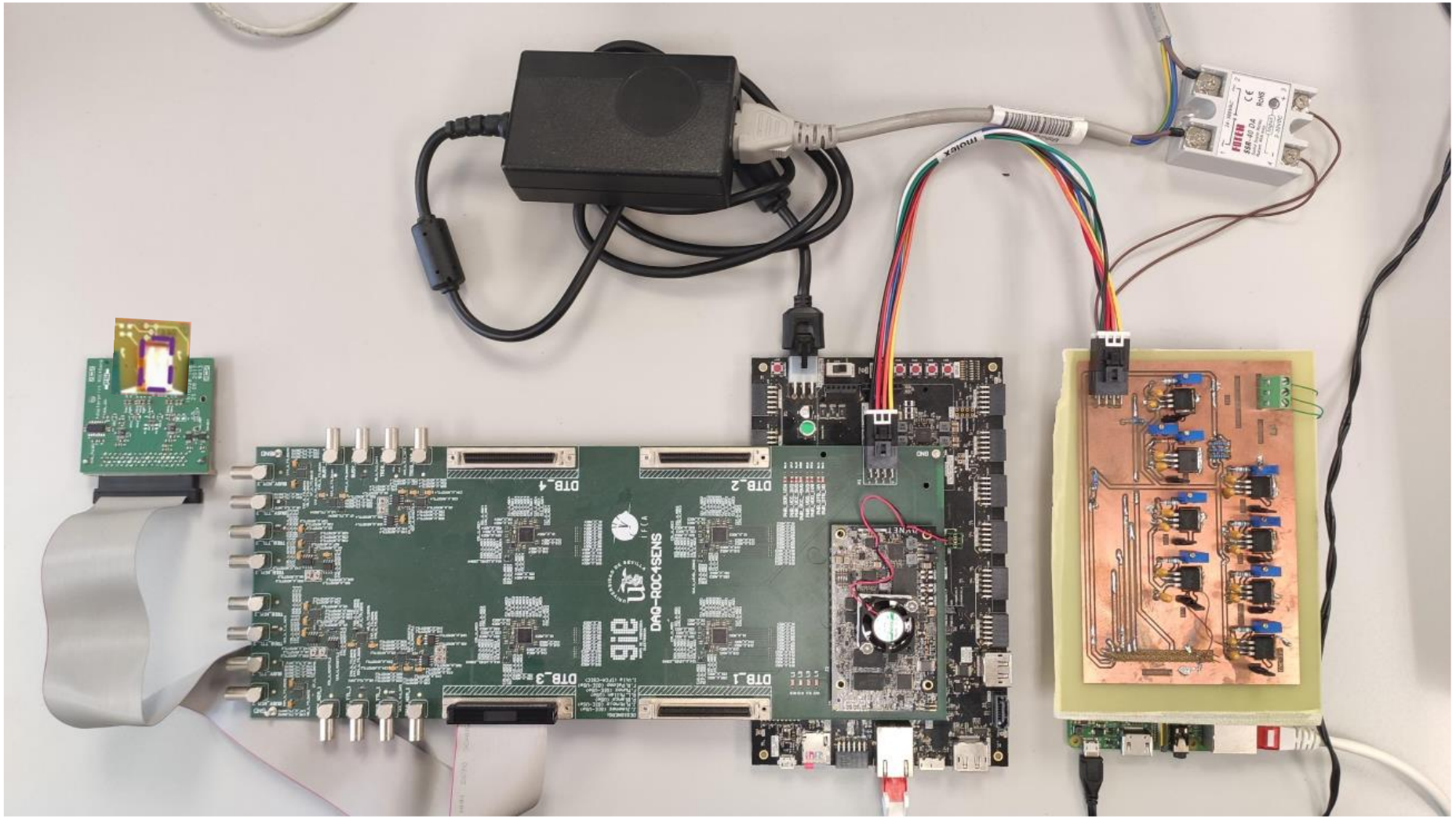
The main idea is to simplify the typical testbeam backend setup (e.g., EUDET telescope) by a standalone device based on powerful novel System-On-Chip-FPGAs, which integrate a programmable logic to implement the readout controllers and a set of multicore ARM processors to provide high-level programmability, such as application based on an architecture client-server, high-bandwidth communications, and high processing capability.

DAQ Roc4Sens



- Sistem On Chip-FPGA
  - Programmable System (Kyntex)
  - Processing System
    - 1 ARM A53 quad core (1.5GHz)
    - 1 ARM R5 dual core (600 MHz)
- AXI internal fast bus for connection between programmable logic and microprocessors
- Shared memory for communications between cores
- ARM R5 for fast microprocessor tasks (no operating system, tipically proxy patterns)
- ARM A53 quad core able to run custom Petalinux Operating System

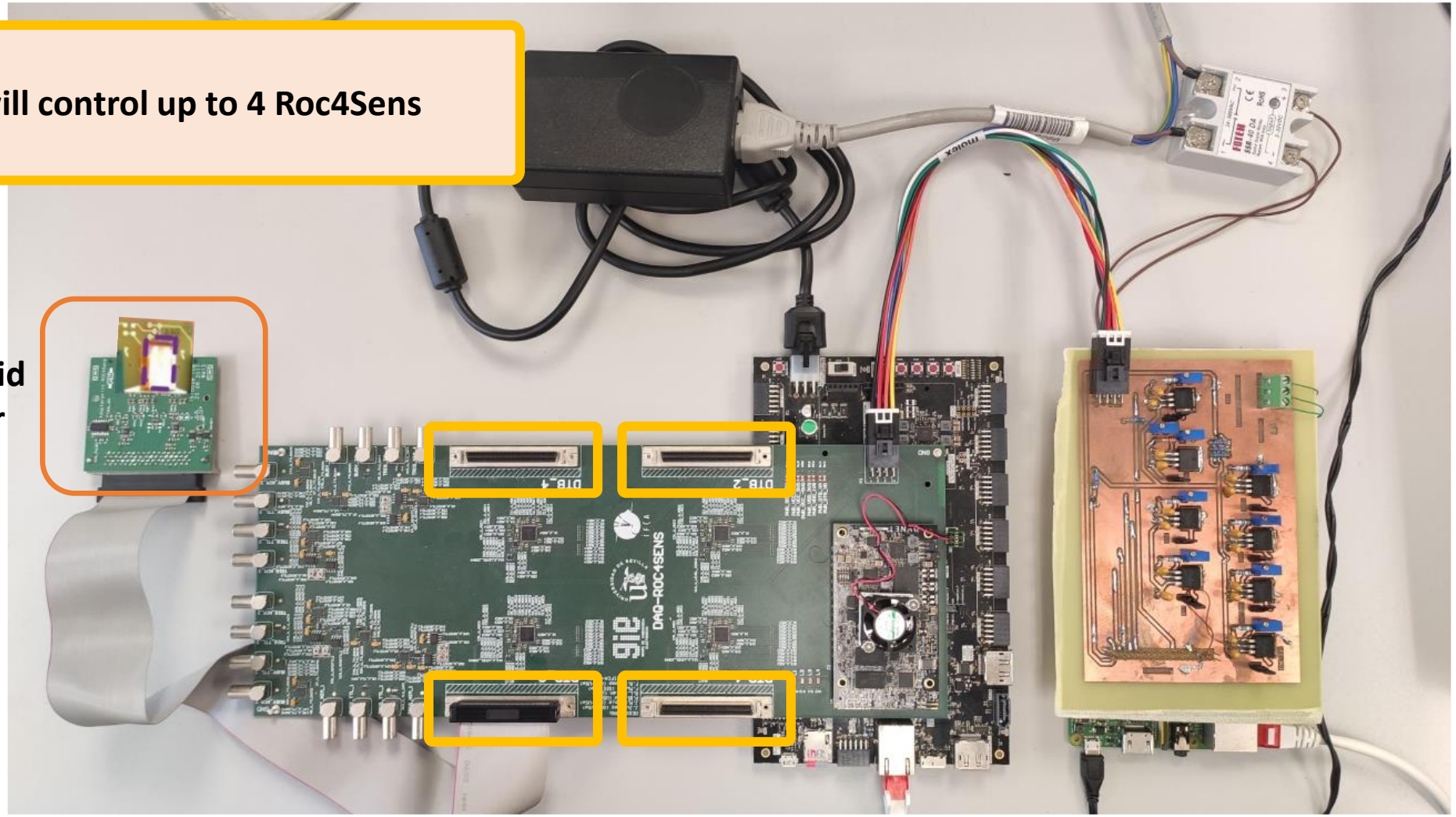
System Description



System Description

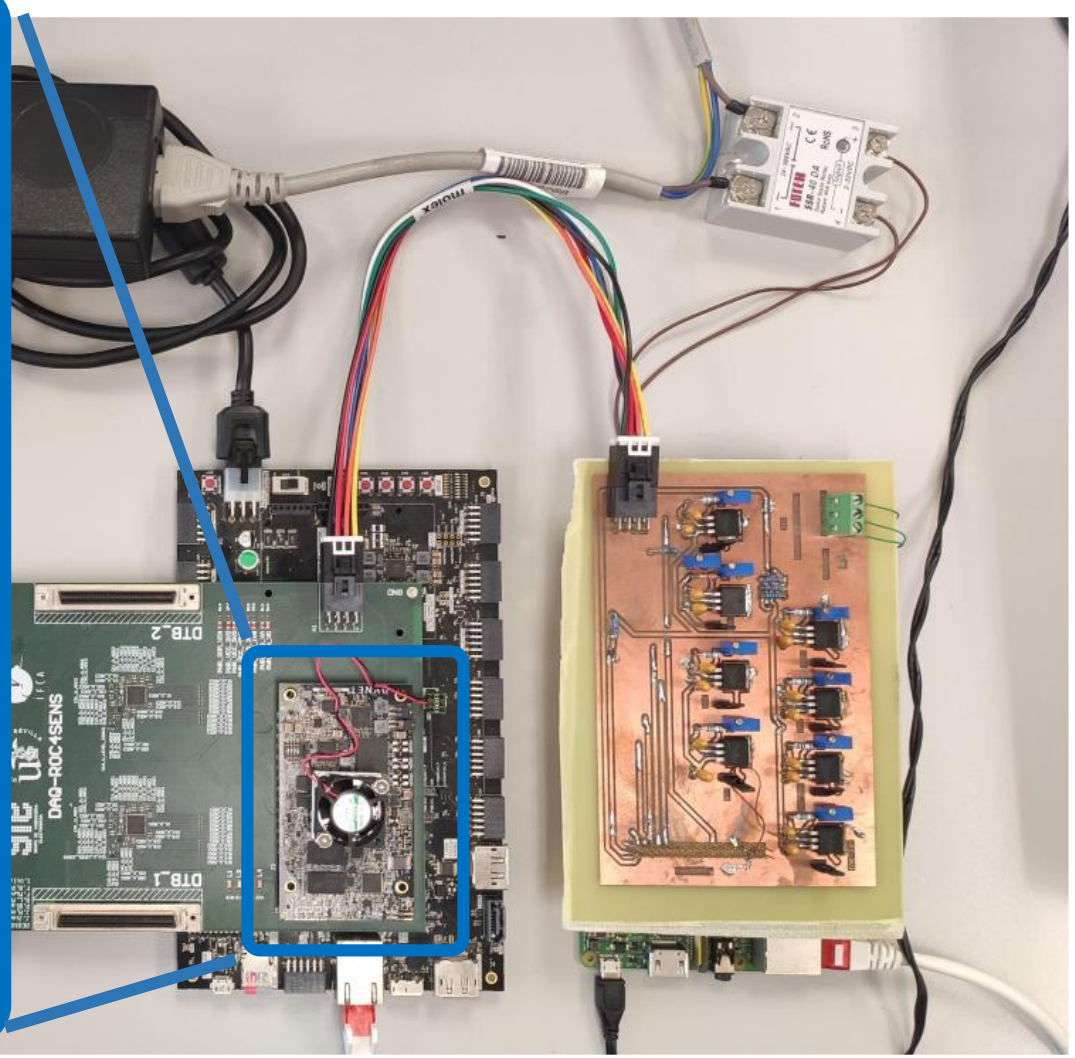
System will control up to 4 Roc4Sens

Roc4Sens Hybrid Pixel Detector

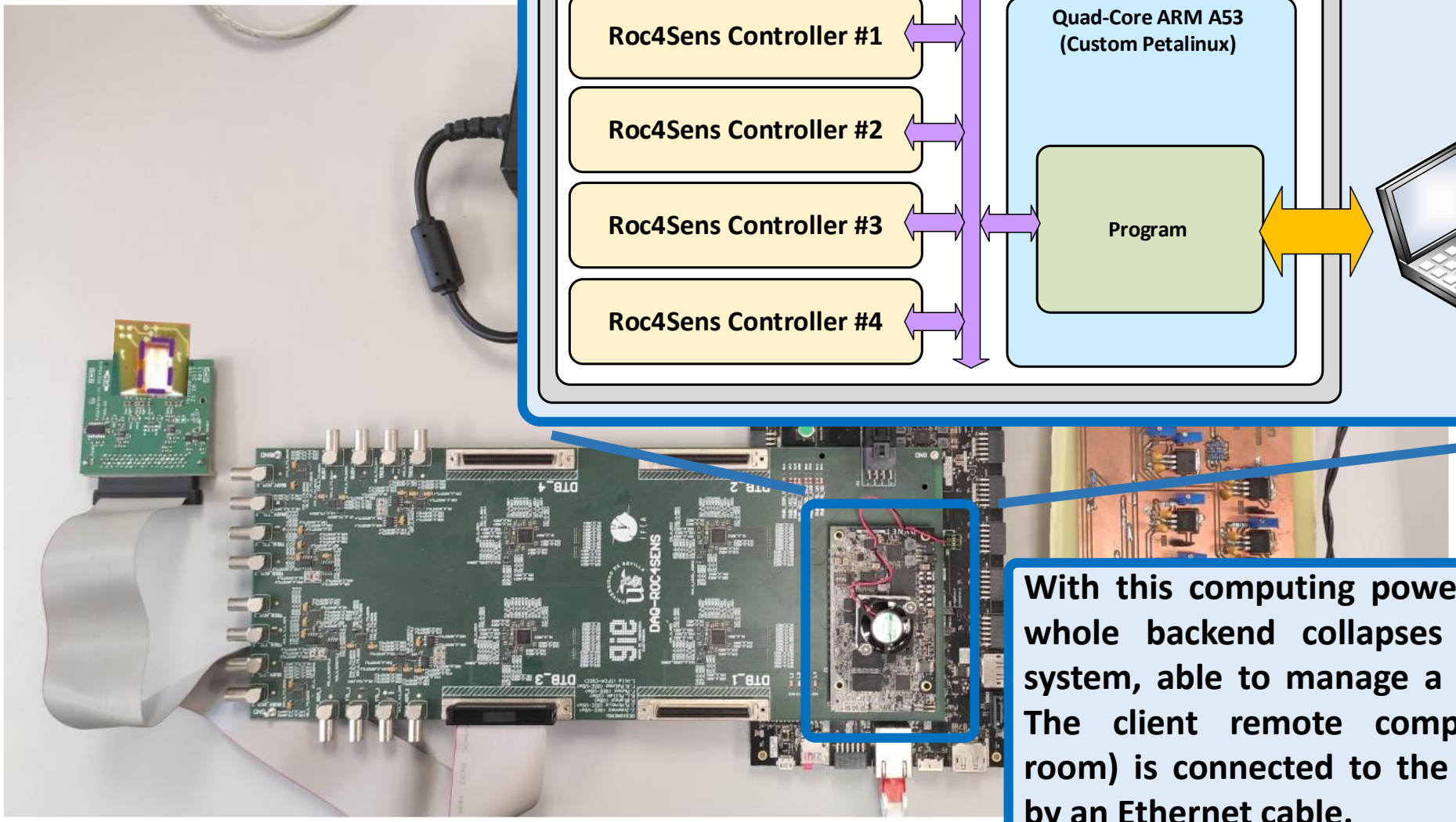


System Description

- Hybrid FPGA
  - Programmable System (Kyntex)
  - Processing System
    - 1 ARM A53 quad core (1.5GHz)
    - 1 ARM R5 dual core (600 MHz)
- AXI internal fast bus for connection between programmable logic and microprocessors
- Shared memory for communications between cores
- ARM R5 for fast microprocessor tasks (no operating system, tipically proxy patterns)
- ARM A53 quad core able to run a custom Petalinux Operating System



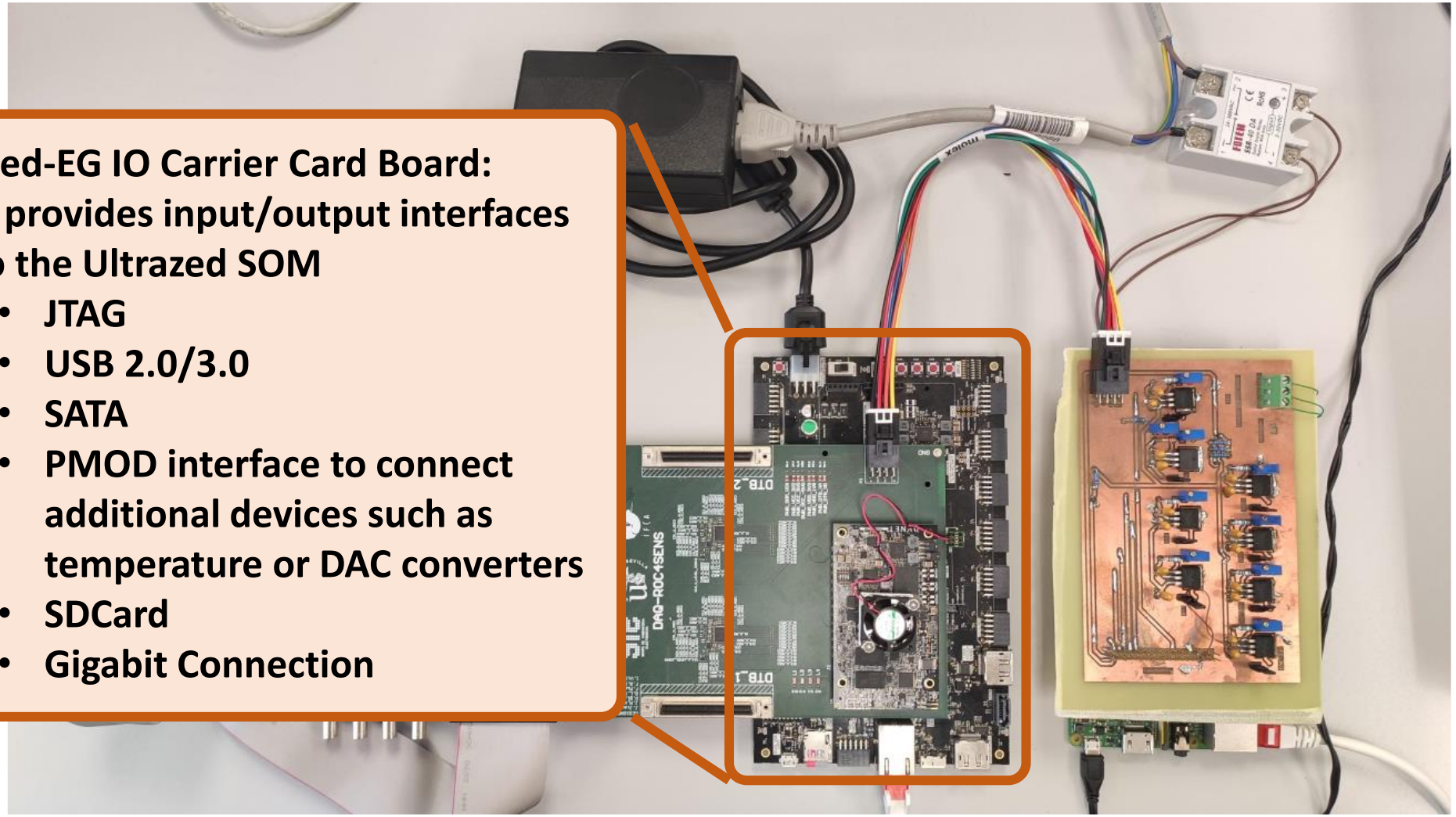
System Description



With this computing power at hand the whole backend collapses to only one system, able to manage a full testbeam. The client remote computer (control room) is connected to the DAQ backend by an Ethernet cable.

System Description

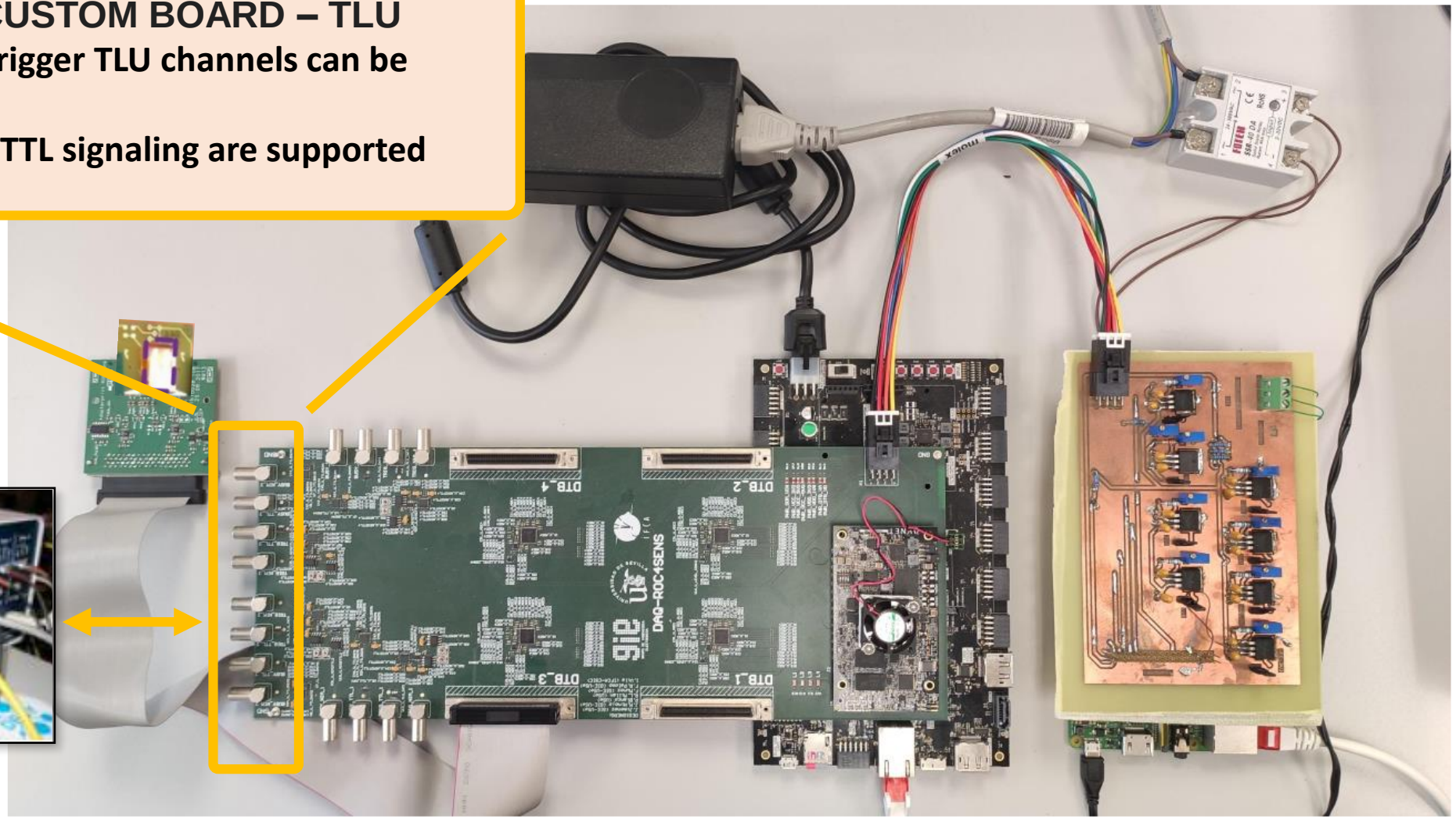
- UltraZed-EG IO Carrier Card Board:
  - It provides input/output interfaces to the Ultrazed SOM
    - JTAG
    - USB 2.0/3.0
    - SATA
    - PMOD interface to connect additional devices such as temperature or DAC converters
    - SDCard
    - Gigabit Connection



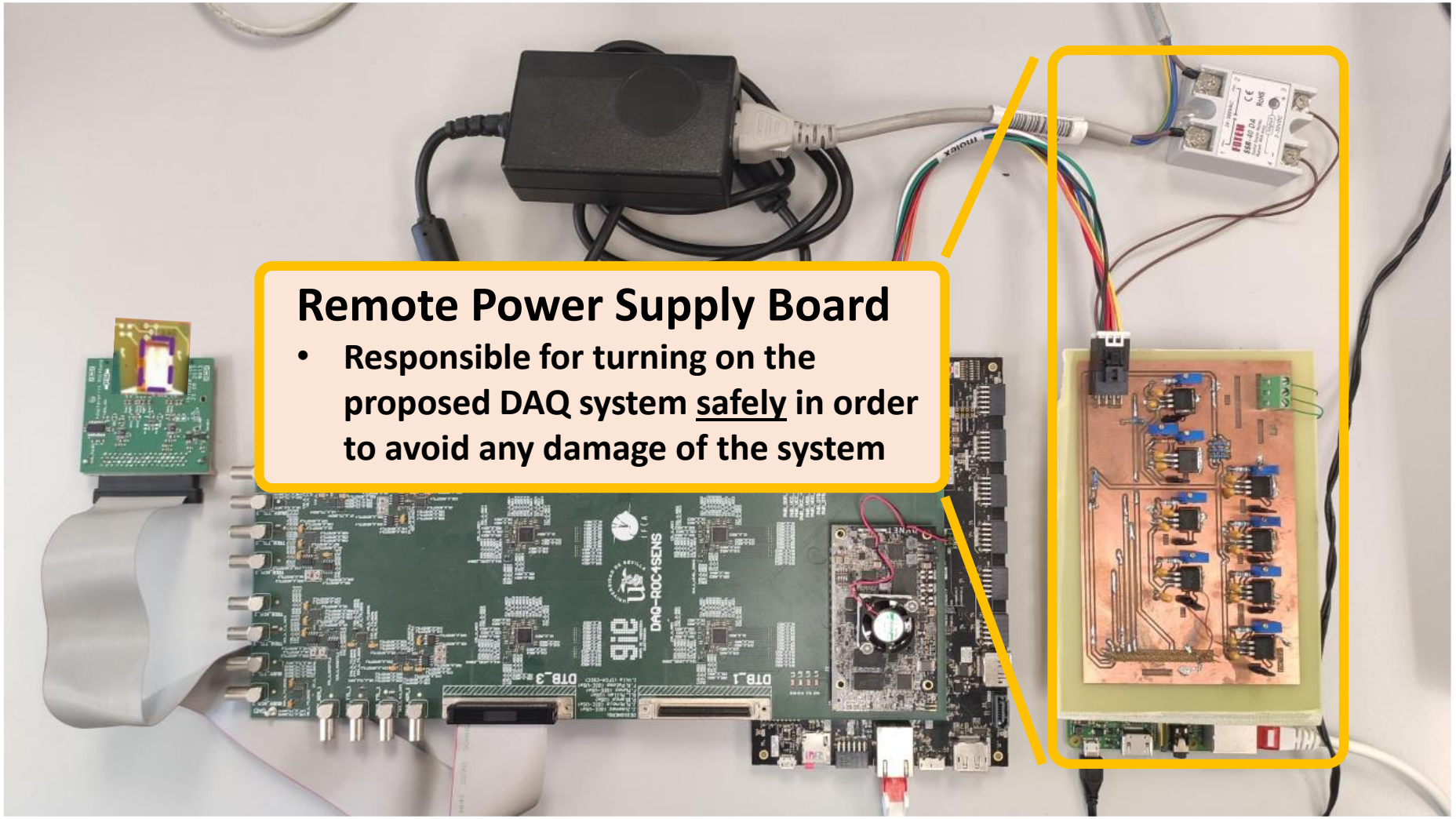
System Description

Interface CUSTOM BOARD – TLU

- **Up to 4 Trigger TLU channels can be used**
- **NIM and TTL signaling are supported**

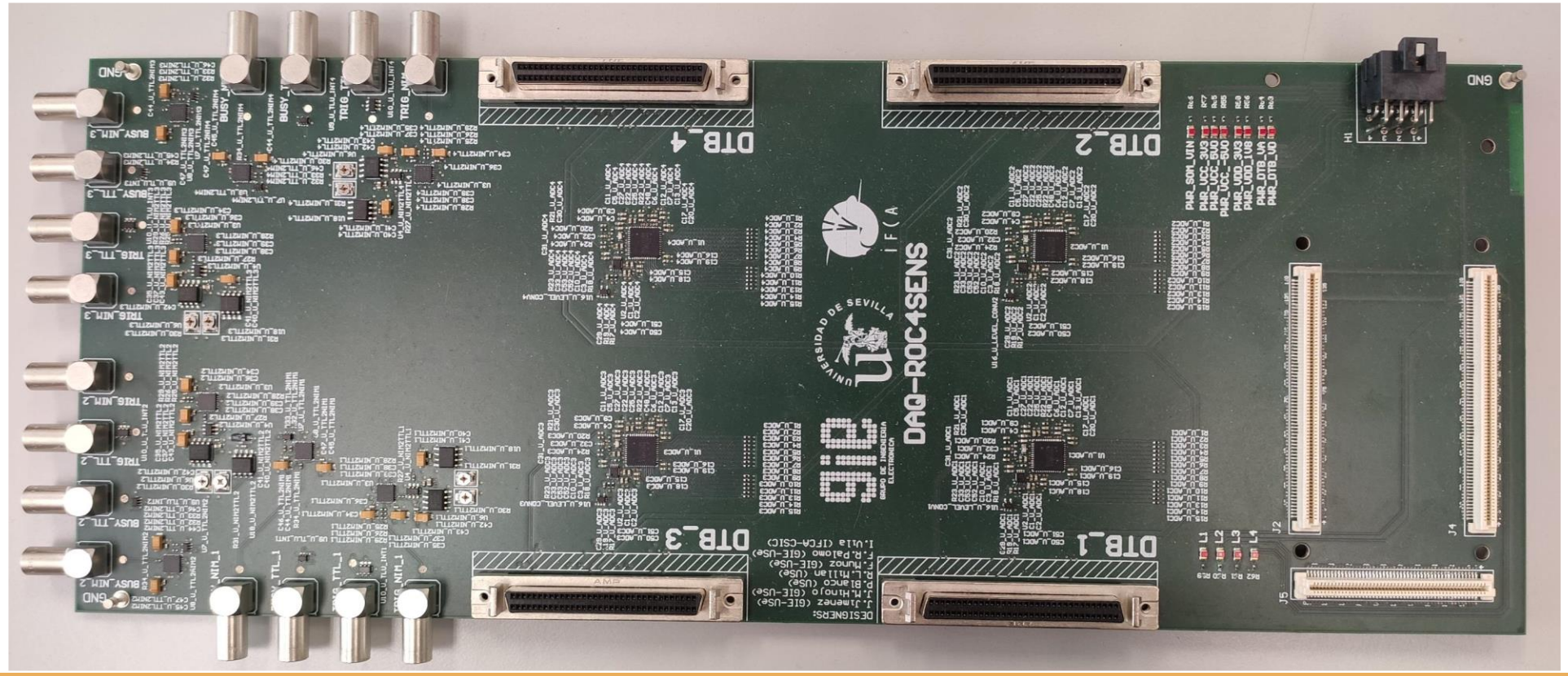


System Description



Custom Board (CB)

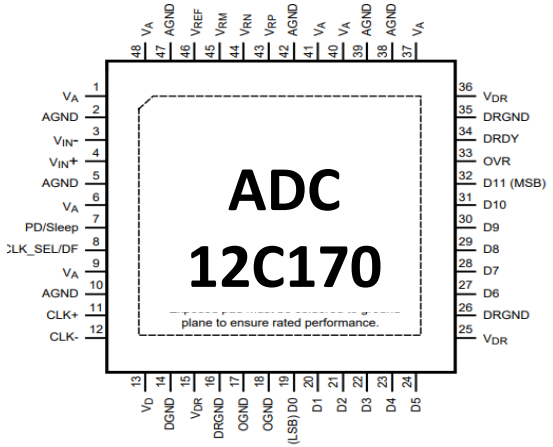
- Printed Circuit Board, responsable of ADC's (x4), Trigger ports (x4) and NIM2TTL conversion
- HV bias pixel sensor goes in parallel by a dedicated cable (not in the PCB, it is safer)
- High frecuency design (differential and single ended lines fully isochronous)
- Ten layers stacked

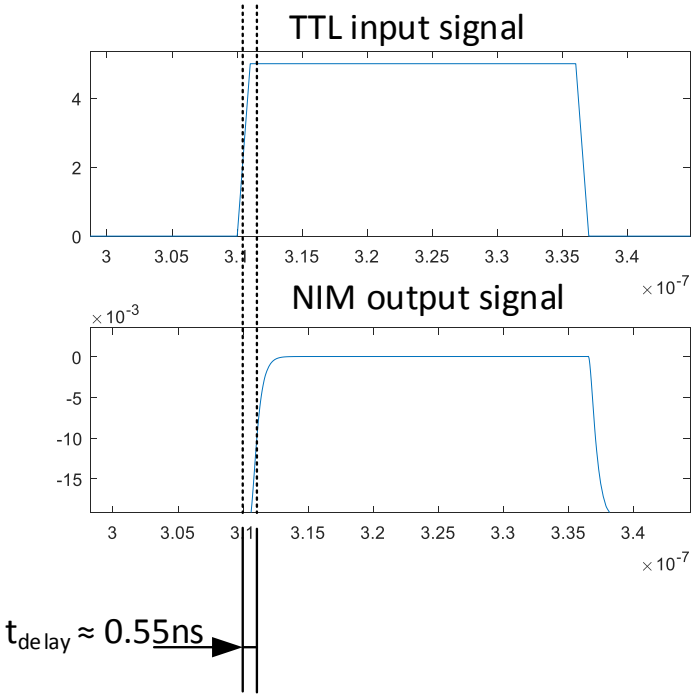


Custom Board

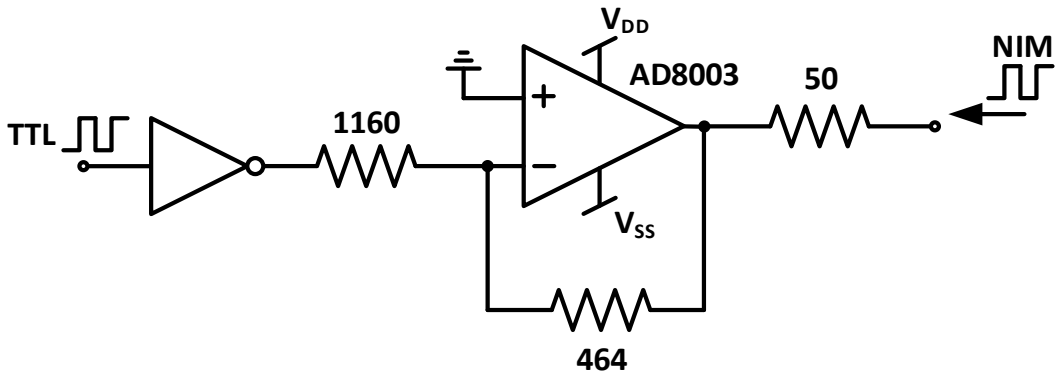
ADC 12C170 Feactures

- Resolution: 12 bits
- Sample Rate (Max): 170 MSPS
- SNR: 67.2 dB
- ENOB: 10.8 Bits
- SFDR: 85.4 dB
- Power Consumption (Typ): 715
- Differential Voltage Input
- Input Voltage Level: 0-2.6 V (lir
- Single Parallel Output CMOS
- Output Voltage Level: 1.8 V
- 1 channel

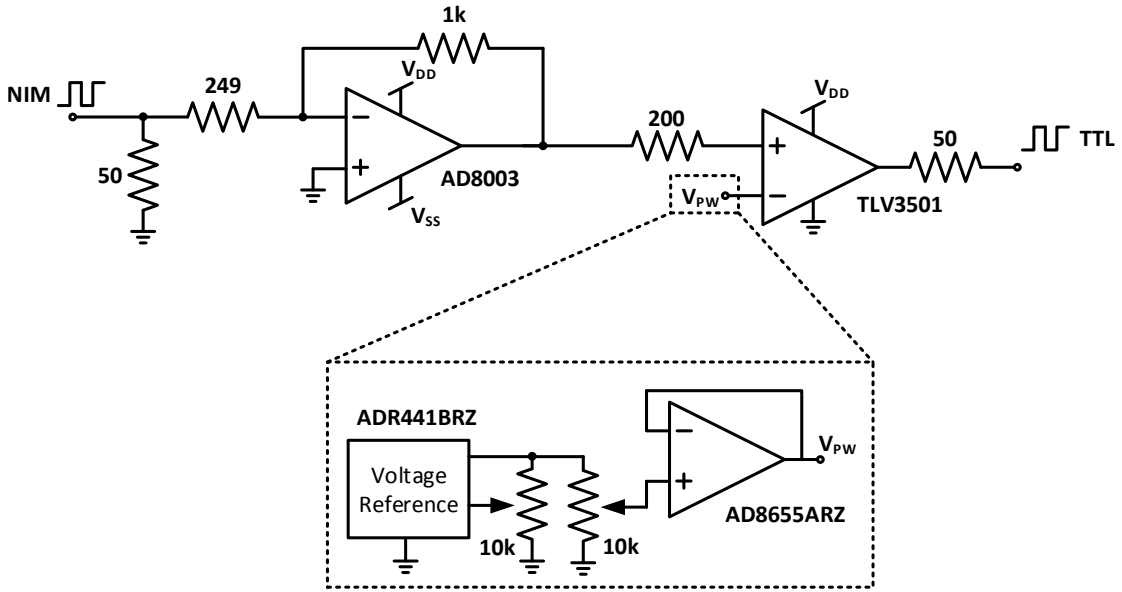




TTL-to-NIM CONVERTER

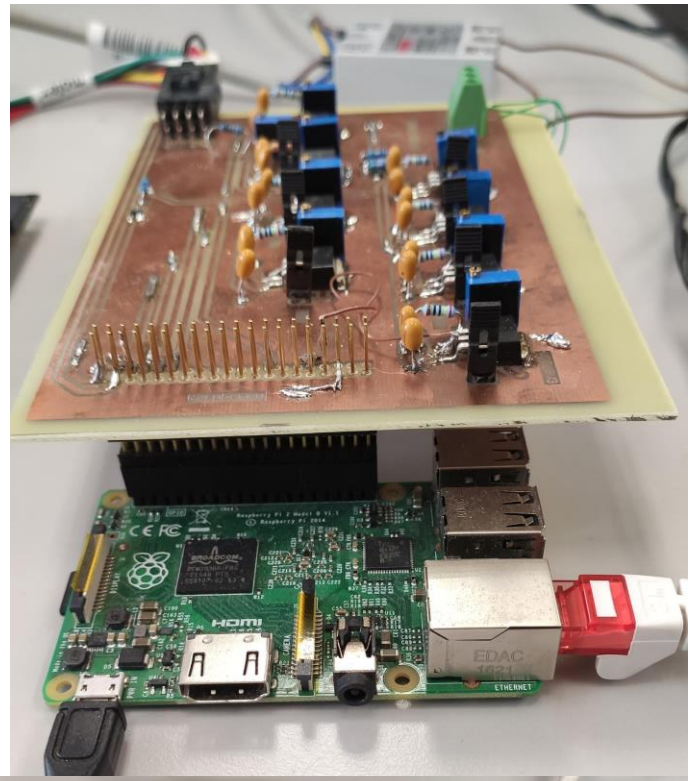


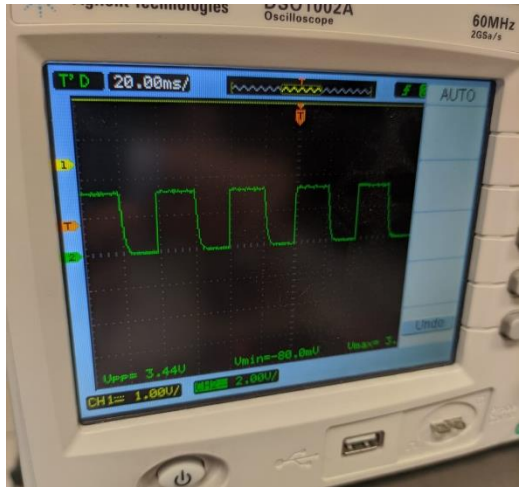
NIM-to-TTL CONVERTER

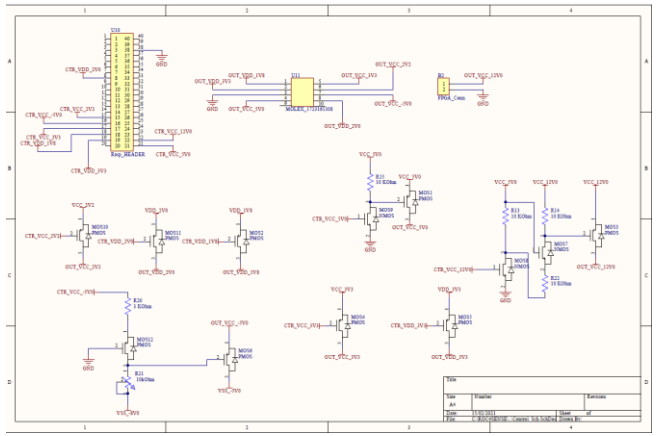


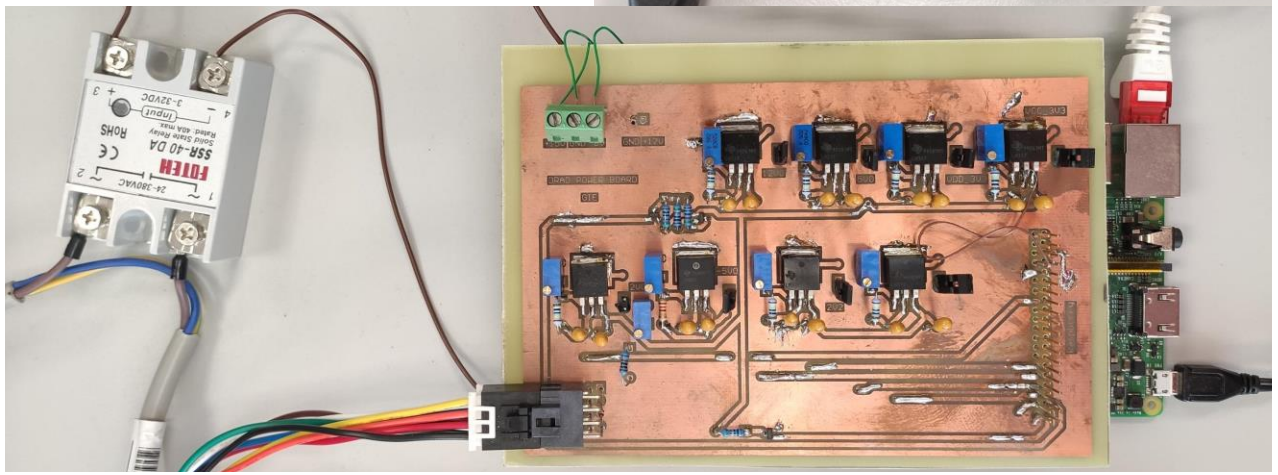
Remote Power Supply Subsystem (R2PS)

- The RP2S allows to turn on the proposed DAQ system remotely and safely
- The start-up sequence is programmable
  - A simple Python app enables individually each supply voltaje
  - Also it allows to sequence each supply voltage to avoid a DAQ system damage.



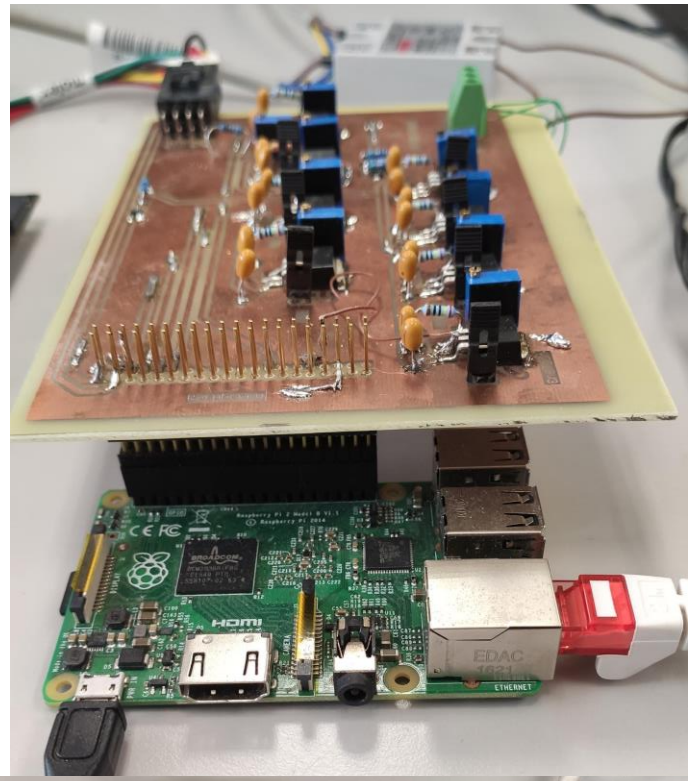


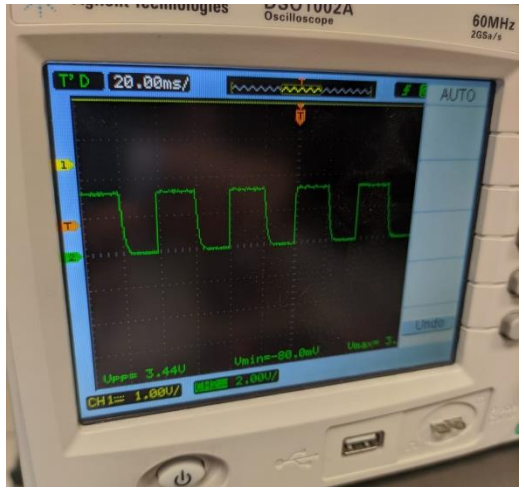


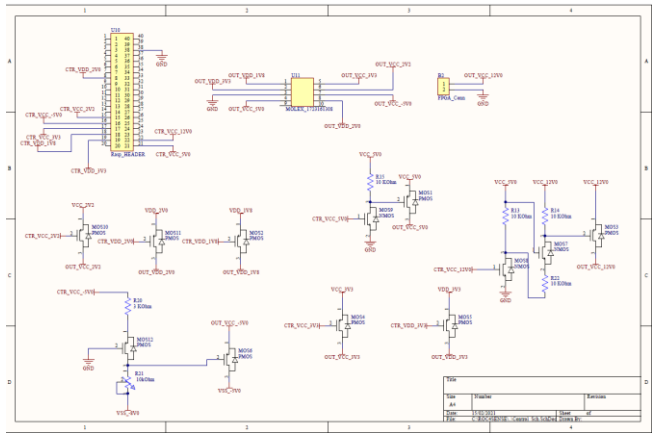


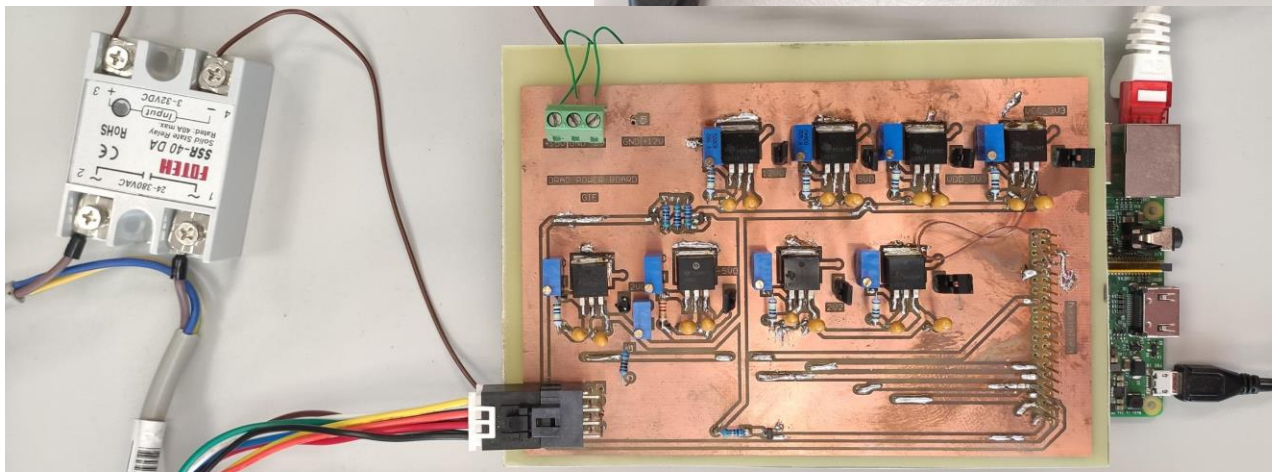
Remote Power Supply Subsystem (R2PS)

- It comprises:
  - Raspberry PI: responsable for managing the start-up sequence and providing remote access
  - Master switch relay that turn on/off the SoC-FPGA
  - A Raspberry PI Hat PCB with switches to turn on/off each supply voltaje to the Common Board (CB) card
  - The HV bias will be managed by SCPI or LXI commands. A second switch relay can be included to turn on/off







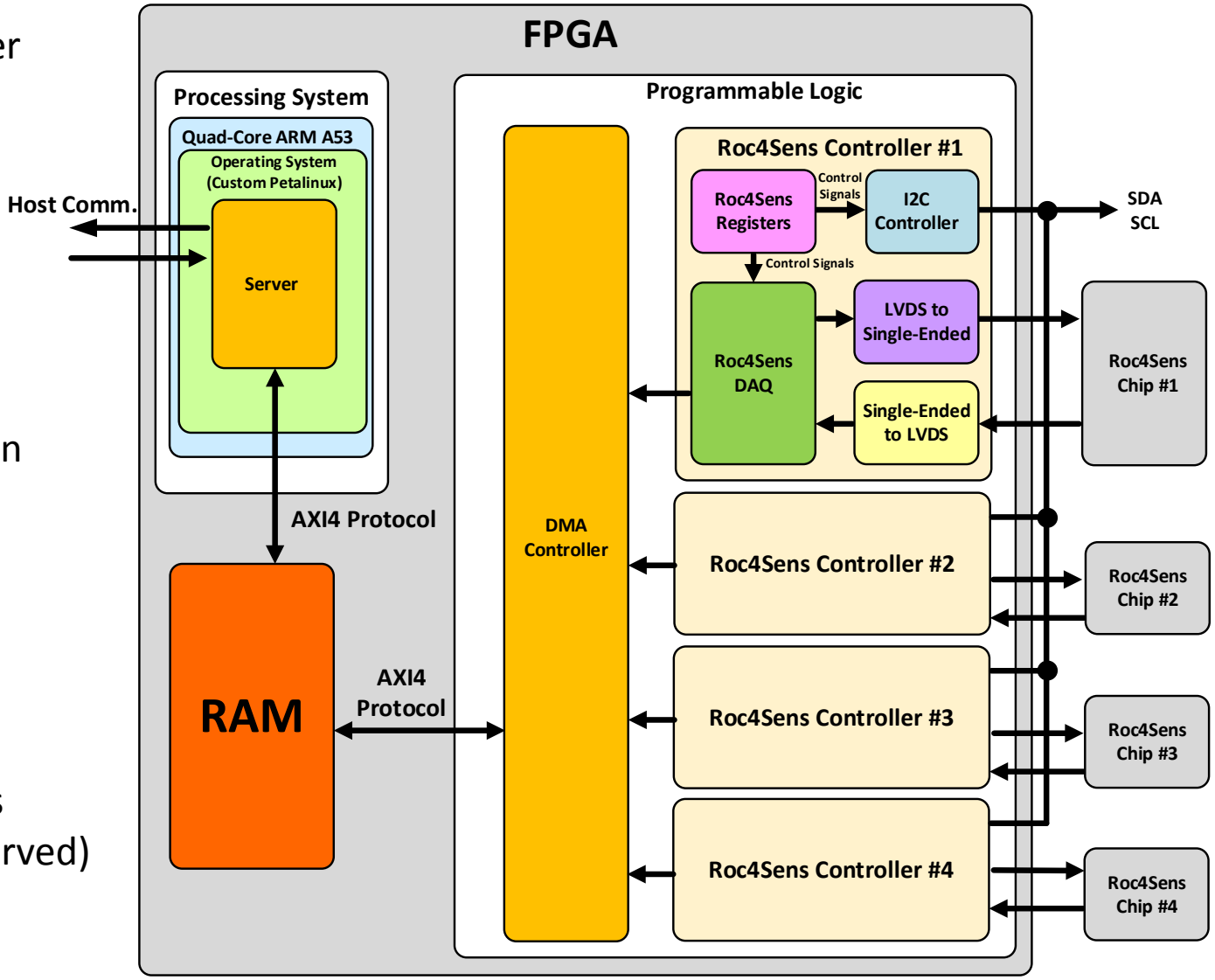


Logical Design

- Kyntex Logic Fabric implements the Roc4Sens controller
  1. I2C (slow protocol)
  2. Roc4Sens propietary (fast protocol)
- Also responsable of the routing IP designs: local registers for control signals and AXI DMA (Direct Memory Access controller) to discharge data into the local RAM. LVDS/SE* services also enabled.
- Up to this moment we do not use the 2core ARM R5 but they are available for proxy tasks (data compression by analysis, data selection, etc).
- We make extensive use of the 4core ARM A53 to run
  - A custom Petalinux Operating System
  - The main DAQ-App, responsible for
    - Remote Procedure Calls (RPC)
    - Data transfers
    - Full duplex data transfer (4x duplicated ports so two data transactions for each ROC are served)

*Low Voltage Differential Signaling/Single Ended signaling

Hybrid FPGA: Block Diagram of the logic implemented



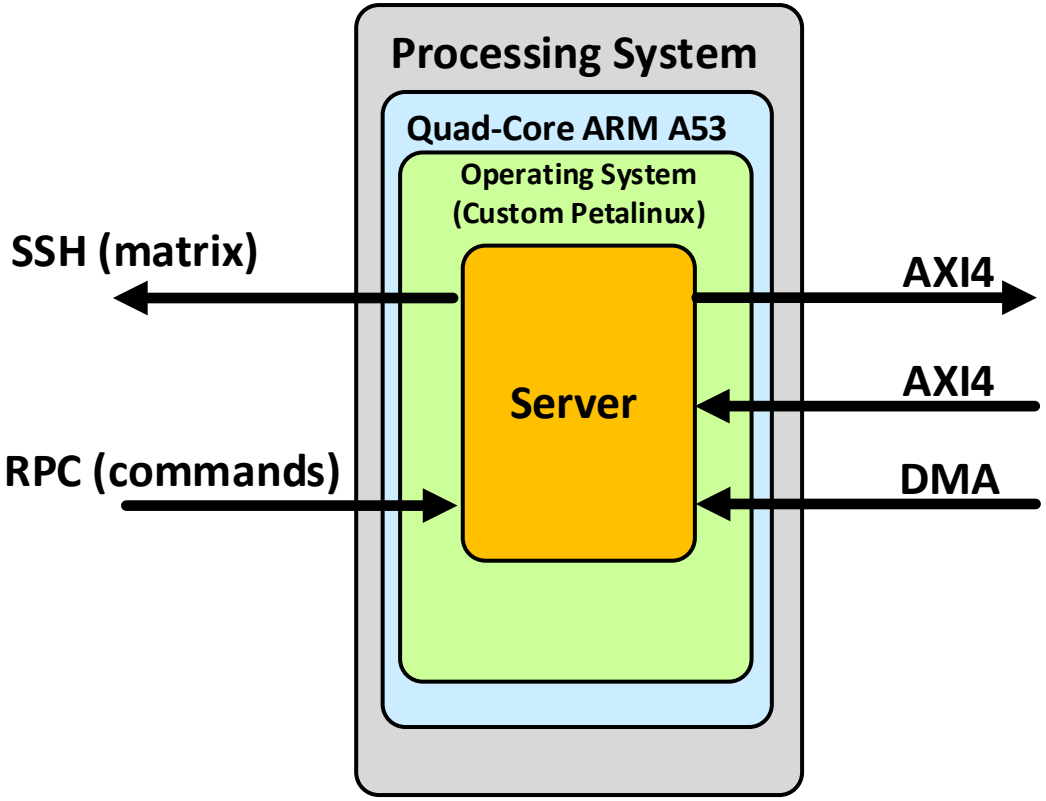
Backend Server

SERVER (on ARM A53 4core, Zynq UltraScale)

- C++ Program
  - Multiple Threads (4)
  - RPC (Remote Procedural Calls) Server
  - Hardware DMA (Direct Memory Access) Manager
  - TCP/UDP Sockets Data Transfer Server (pixel matrix data transfer for each ROC-sensor assembly)

Communications

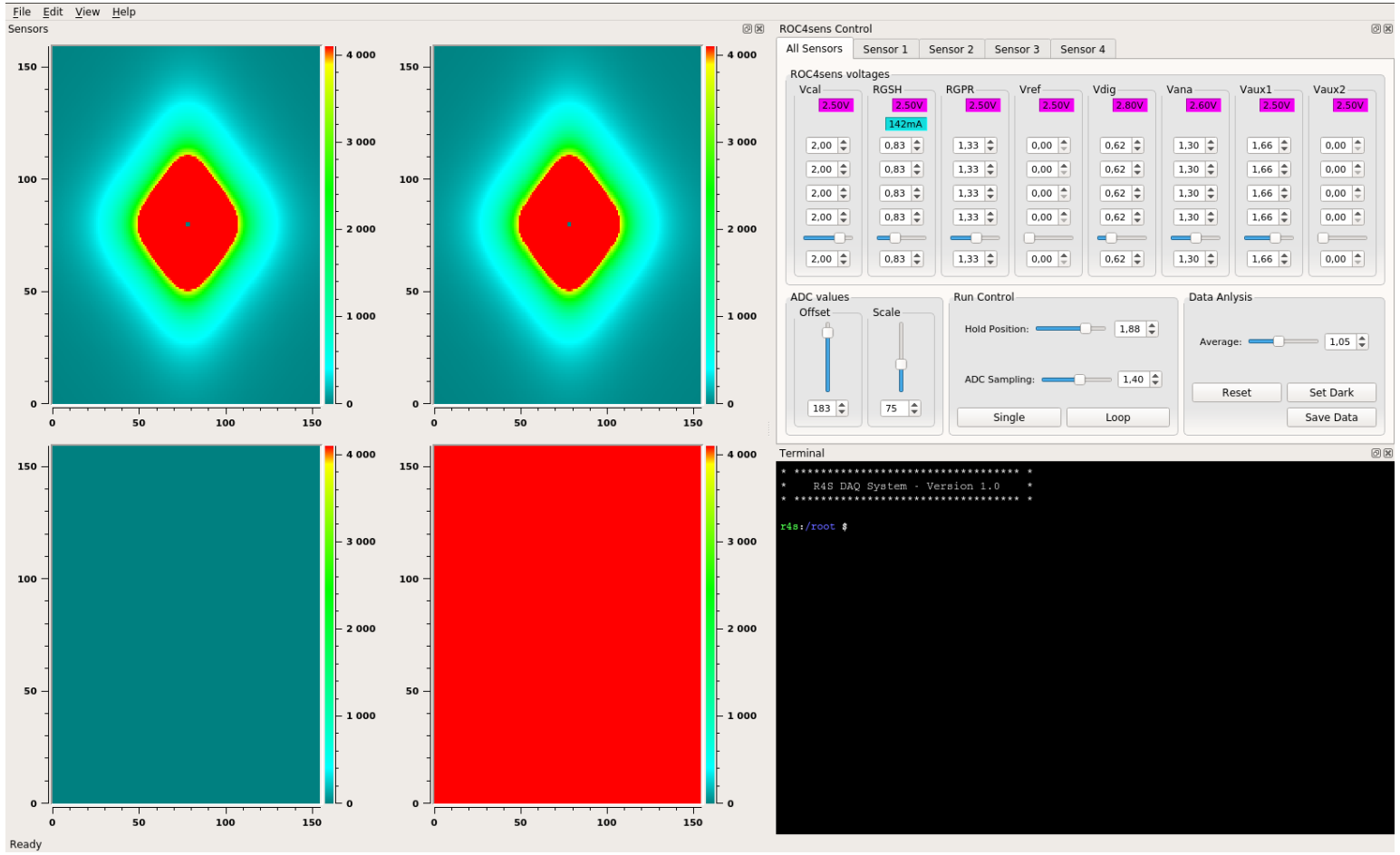
- PC (ports 3550-3553 & 3554-3557 for data transfer, full duplex for triggered and non triggered data transfer)
  - RPC for Command Transfer
  - TCP/UDP Sockets for Data Transfer
    - (handshake-transmission-close procedure)
- **Programmable logic**
  - Hardware DMA
  - AXI4 Fast Bus Transfer



Remote Interface (remote laptop)

- Runs on a remote laptop
- The Client Communicates with the backend by Remote Procedural Calls (RPC) and by UDP/TCP data transfer
- Four pixel displays
- Full set of configuration controls
- Shell available for Python scripts
- Full Duplex (8 ports, 2 associated to each ROC for 2 simultaneous data transfer, triggered and not triggered)
- gClient/Server and Server/Client if necessary (architecture duality)

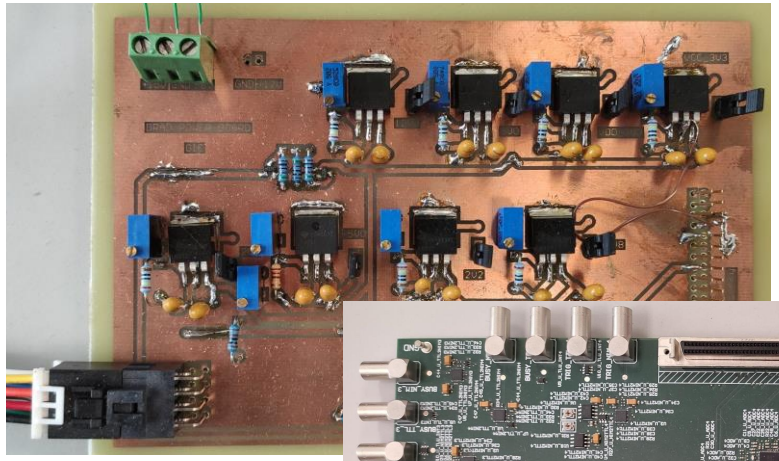
Human Interface (running remotely in a laptop)

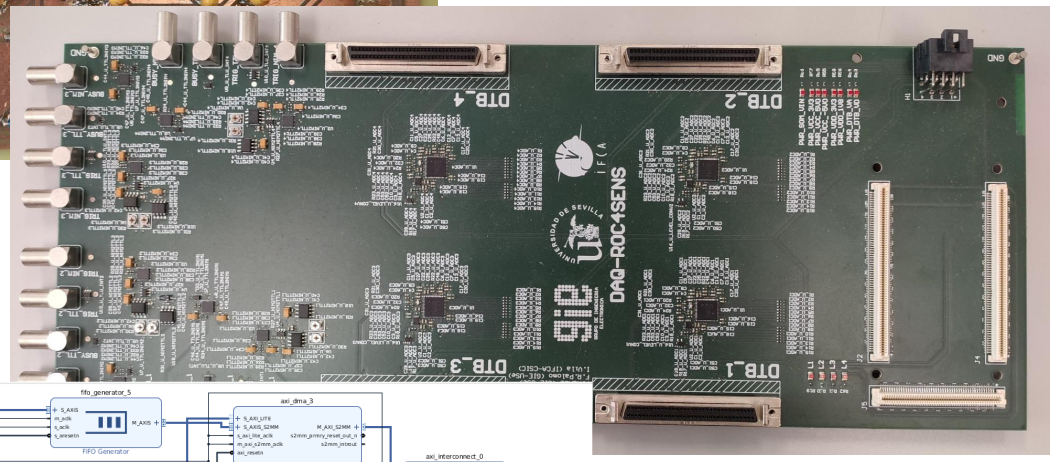


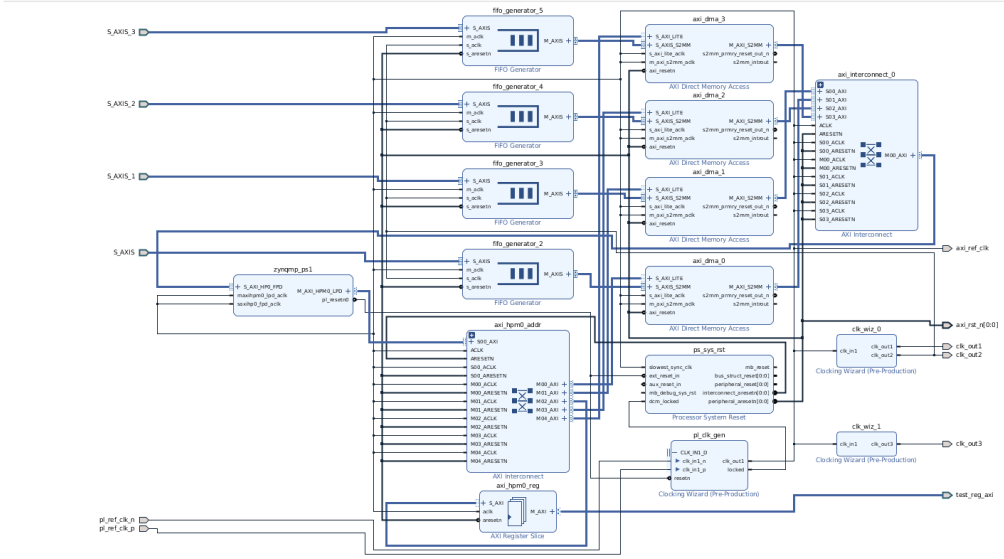
DAQ Roc4Sens: State of the Project

- Development Stage
  - Logical Design: Finished ✓
    - ADC management integrated on Roc4Sens Controller
  - Local Application Software: Finished ✓
  - Client Interface (on the remote host): Finished ✓
  - Custom Board: Finished ✓
  - Remote Power Supply Board: Finished ✓
  - Python API to manage Roc4Sens chips: In progress 

  - Custom Petalinux Image: In progress 





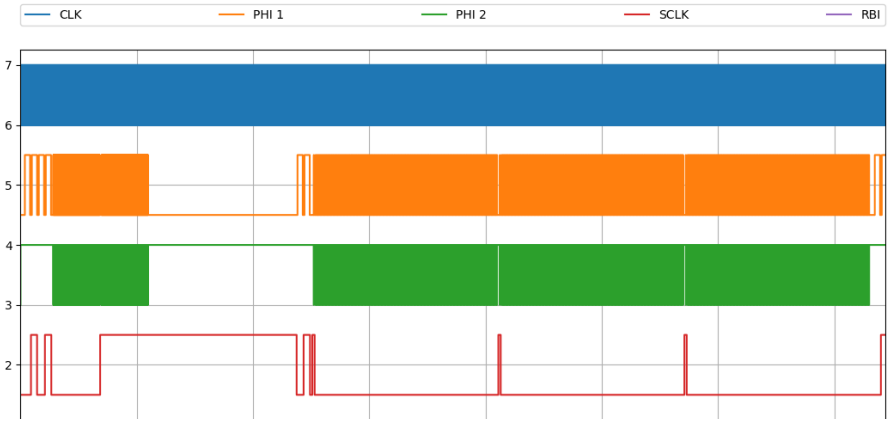


DAQ Roc4Sens: State of the Project

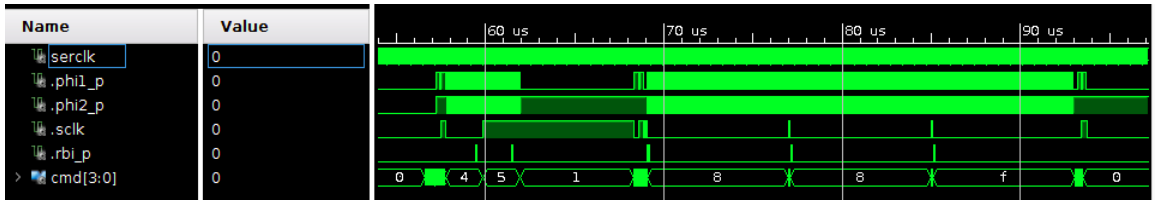
- Testing Stage
  - Logical Design verified using non-synthesizable VHDL code to emulate Roc4Sens chip and ADCs
    - A Roc4Sens command interpreter was developed to ease the debugging process
  - Custom Board (ADC's x 4, TLU ports x 4):
    - Performed Electrical test
    - CB Functional Test 

  - Operating System was upgraded to a custom Petalinux to improve the Ultrazed SOM and Ultrazed IOCC support 

Full system working (foreseen) in Q1 2022



Hybrid FPGA: Digital Design Signals towards the ROC Roc4Sens



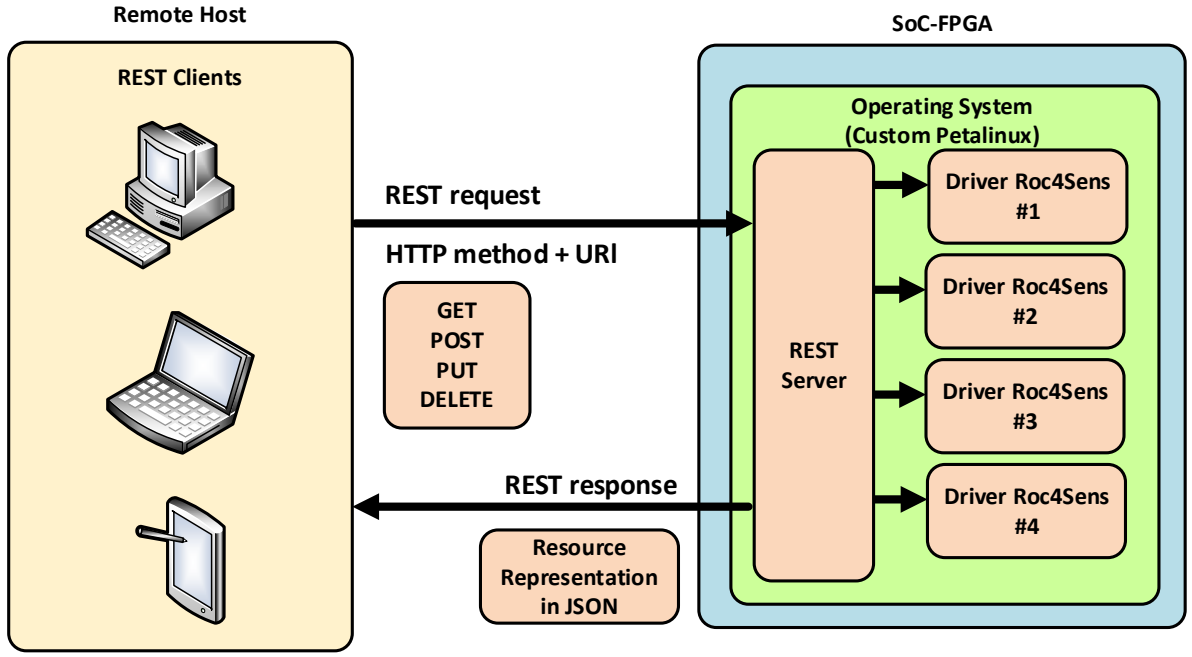
testSensor1.txt: Bloc de notas

Archivo Editar Formato Ver Ayuda

```
|    TIME    | COMMAND CYCLE | PHI1 | PHI2 | SCLK | RBI |
57115.625 ns,         START,    0000,  0000,  0000,  0000;
57140.625 ns,      CYC_WAIT,    0000,  0000,  0000,  0000;
57165.625 ns,      CYC_WAIT,    0000,  0000,  0000,  0000;
57190.625 ns,      CYC_WAIT,    0000,  0000,  0000,  0000;
57215.625 ns,      CYC_WAIT,    0000,  0000,  0000,  0000;
57240.625 ns,      CYC_RES0,    1111,  1111,  0000,  0000;
57265.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57290.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57315.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57340.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57365.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57390.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57415.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57440.625 ns,      CYC_WAIT,    1111,  1111,  0000,  0000;
57465.625 ns,      CYC_DIRX,    0000,  1111,  0000,  0000;
```

DAQ Roc4Sens: Future Work

- Developing a REST API to enhance the usability of the DAQ system
  - Architecture client-server composed of clients, servers and resources
  - Broad client support: any application or framework with support for HTTP could interact with the DAQ
  - The backend could be reused in new DAQ systems
- Including a Microprocessor-based Hardware Event Trigger in the SoC-FPGA



Conclusions

- DAQ-Roc4Sens at 65% of finalization
- Drastic simplification of the typical testbeam backend arrangement: only one backend acting as a server
- High Data Rate Transfer (>> Mbs), it allows data processing in real time
- Fully comprehensive interface (shell included)
- Availability expected in Q1 2022

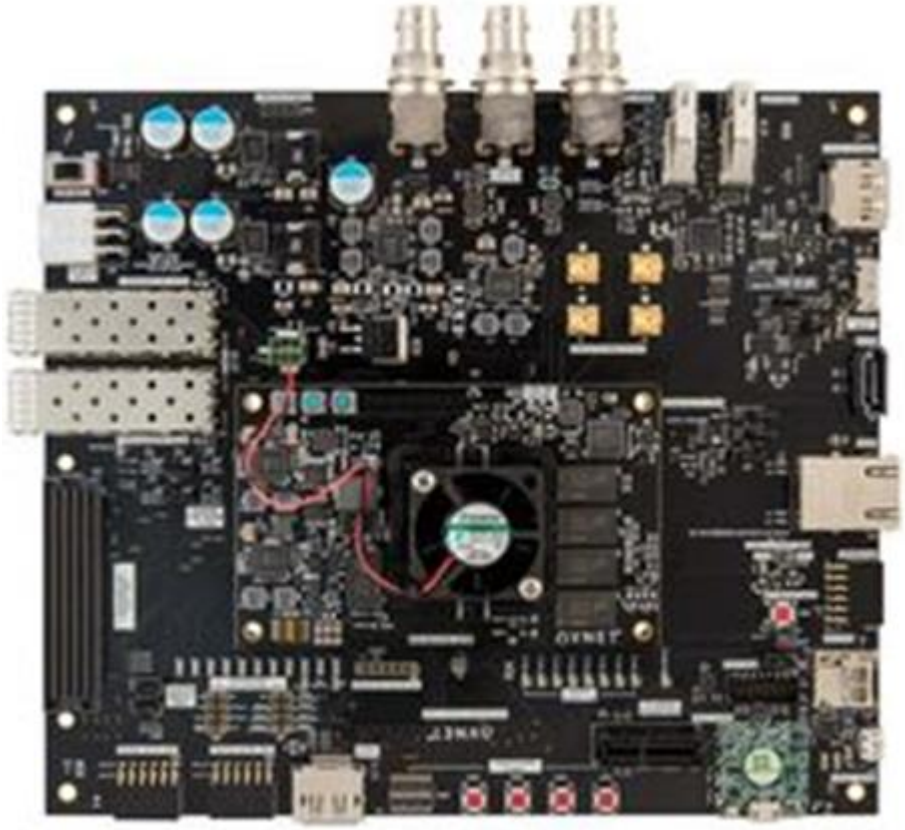
Thanks for your attention

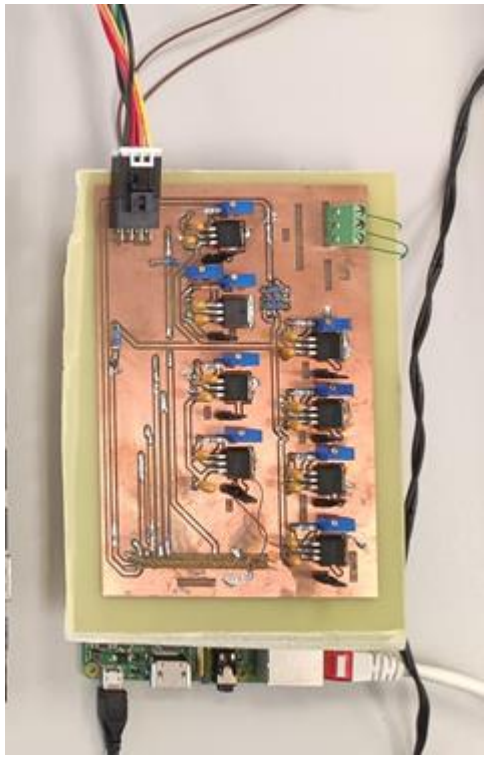
fpalomo@us.es

Backup Slides

New Hardware for Future Compatibility

- UltraZed-EV SOM + UltraZed-EV IOCC
  - Same FPGA architecture: Xilinx ZYNQ Ultrascale+ MPSoC
  - Support an FMC interface (compatible with Caribou DAQ)
  - 2x SFP+ interfaces to high-bandwidth communication or to implement Aurora protocol
  - 4x high-speed PS-GTR transceivers





- Mother board designed in collaboration with the IN2P3 Annecy laboratory
- Features:
  - require +5 V (3A) / -5V (100mA)
  - nanoPC BeagleBone card + FPGA
    - Connected via GPMC bus
    - Flexible programing
    - 40 TTL signals
    - 30 LVDS signals
    - Analog channels available
      - 4 channels SAR ADC's 16 bits (ADS8568 – TI)
      - 10 DAC's 16 bits (DAC8830 – TI)
    - 4 programmable regulators
      - 2 x LT 3021
      - 2 x LT 3026 (up to 1,5 A)
        - Must be tested
    - Monitoring:
      - °C: MAX31865 (RTD-digital converter)
      - current supply: MAX611
  - Lab tests (local mode)
  - irradiation tests (remote mode via LVDS signals)