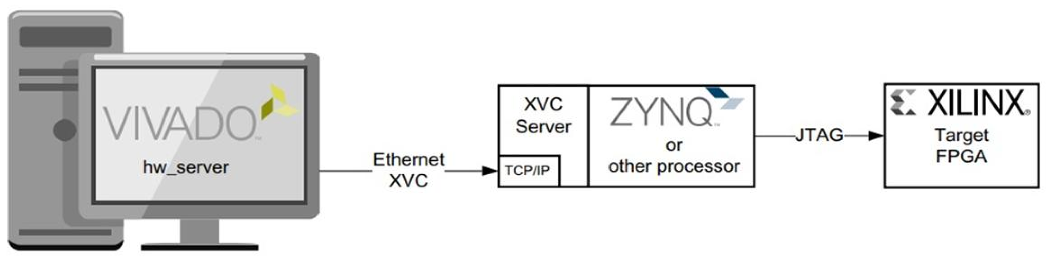
Study of Decoupler: Empowering FPGA Debugging with ESP32 and IoT

Yifan Yang (yang.yifan@ulb.be), Feng Gao, Pierre-Alexandre Petitjean, Barbara Clerbaux
Inter-University Institute For High Energies, Université libre de Bruxelles (ULB)

Common features of high energy physics experiment electronics system

- Front-end electronics(~1000s of channels).
- Back-end electronics(~100s of boards) .
- Large number of custom designed FPGA based PCBs.
- Either easy or difficult to access after installation.
- Require constantely parameter monitoring before and after installation at low frequency(~minutes).

Limitations of traditional solution



- Using JTAG programmer at early stages.
- Remote configuration and slow control system at late stages.
- Solution requires extra logic and space resource and not available at the beginning.

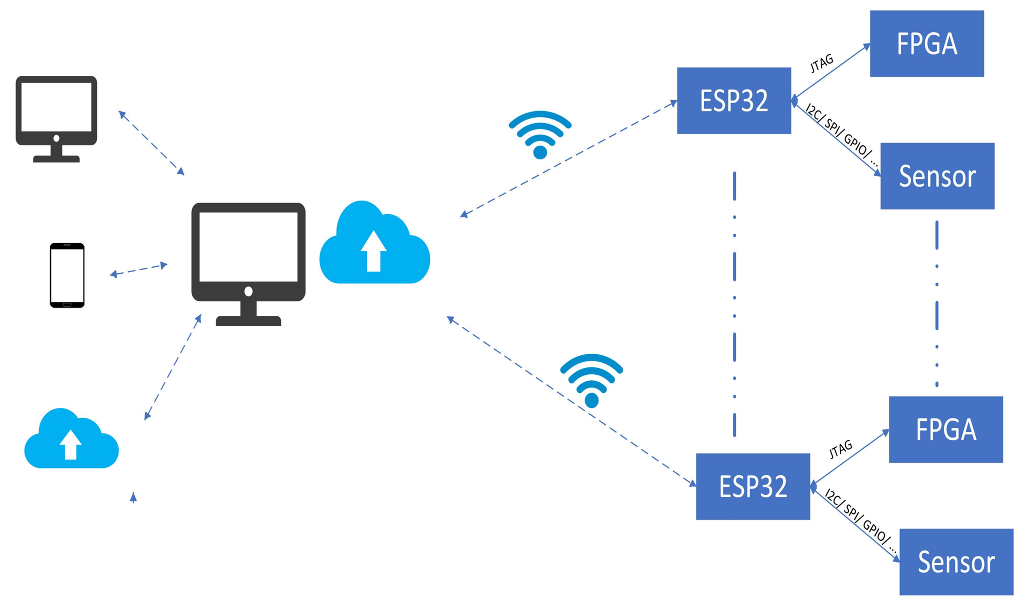
Decoupler

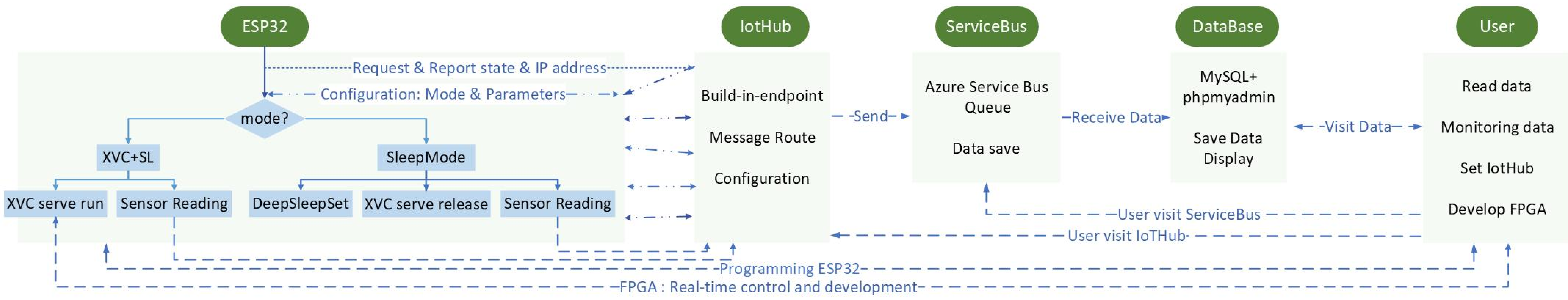
By implementing remote configuration and environment monitoring independently :

1. Decouple the core function design with peripheral functions such as environment monitoring.
2. Decouple the installation with maturation of the firmware design.

Key advantags:

- Better security and reliability.
- Lower logic and PCB resource.
- Less material and power consumption.
- Easier maintainability and scalability.





ESP32

-Xtensa® dual-core 32-bit LX6 microprocessor, up to 240 MHz
-Wi-Fi included/Up to 34 GPIOs
-As IoT device send data to IoTHub
-As XVC server access to FPGA

XVC

-TCP/IP-based protocol that acts like a JTAG cable.
-Facilitate hardware debug for designs that: have the FPGA in a hard-to-access location, where a "lab-PC" is not close by.

Azure IoTHub

-Security-enhanced communication channel for sending and receiving data from IoT devices.
-Route message to different destinations automatically.

Azure servicebus queue

-Queues store messages until the receiving application is available.
-Messages in queues are ordered and timestamped on arrival.
-Messages are delivered in pull mode, only delivering messages when requested.

DataBase

-MySQL+phpMyAdmin
-Robust and reliable local database.
-Insert data immediately after pulling from servicebus.

Test Setup & Real-Time Debugging and Environment Monitoring & Data Management

- Test Setup: ESP32 with a sensor DTH11 is connected to a Kintex 7 FPGA through JTAG interface and is powered by the JTAG's VREF.
- Real-Time Debugging with XVC server.
- Environment Monitoring by distributing sensor data through Azure IoTHub and service bus queue.

Fig: Setup for ESP32+Sensor+FPGA

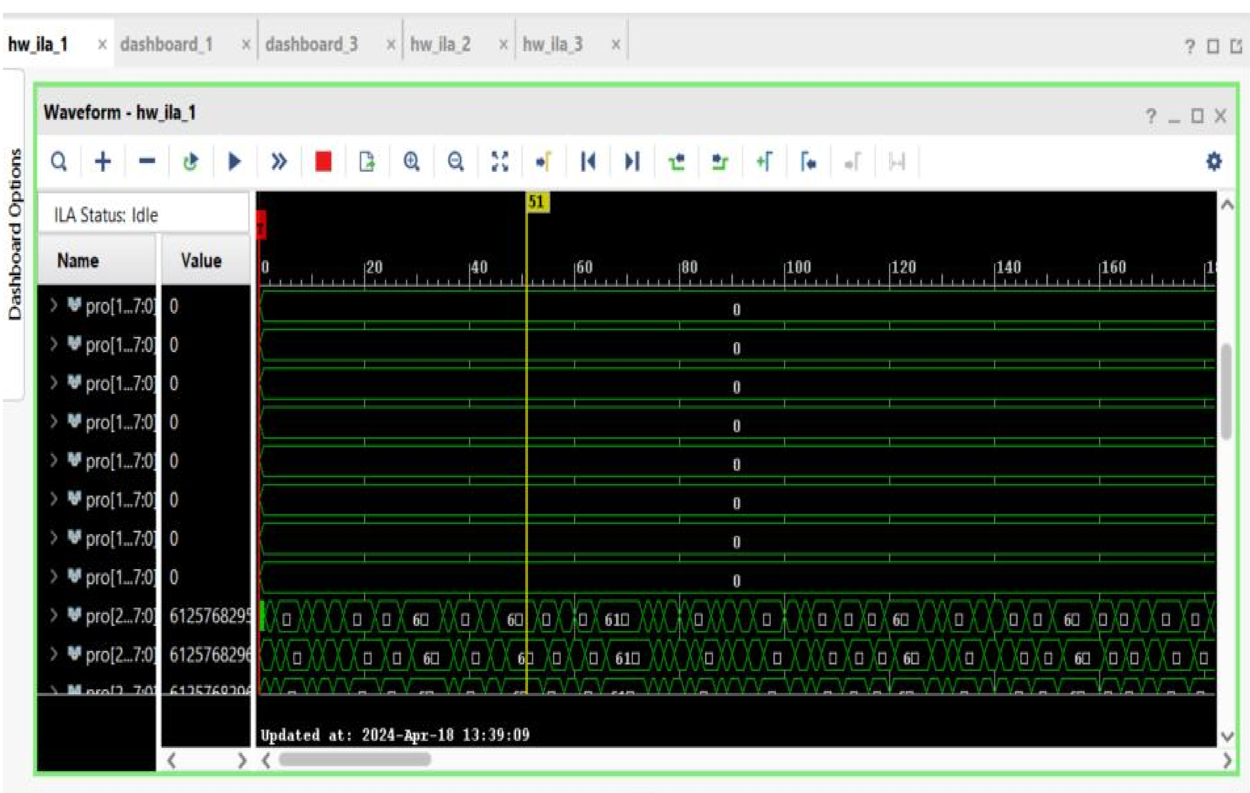


Fig: Interface for FPGA debug

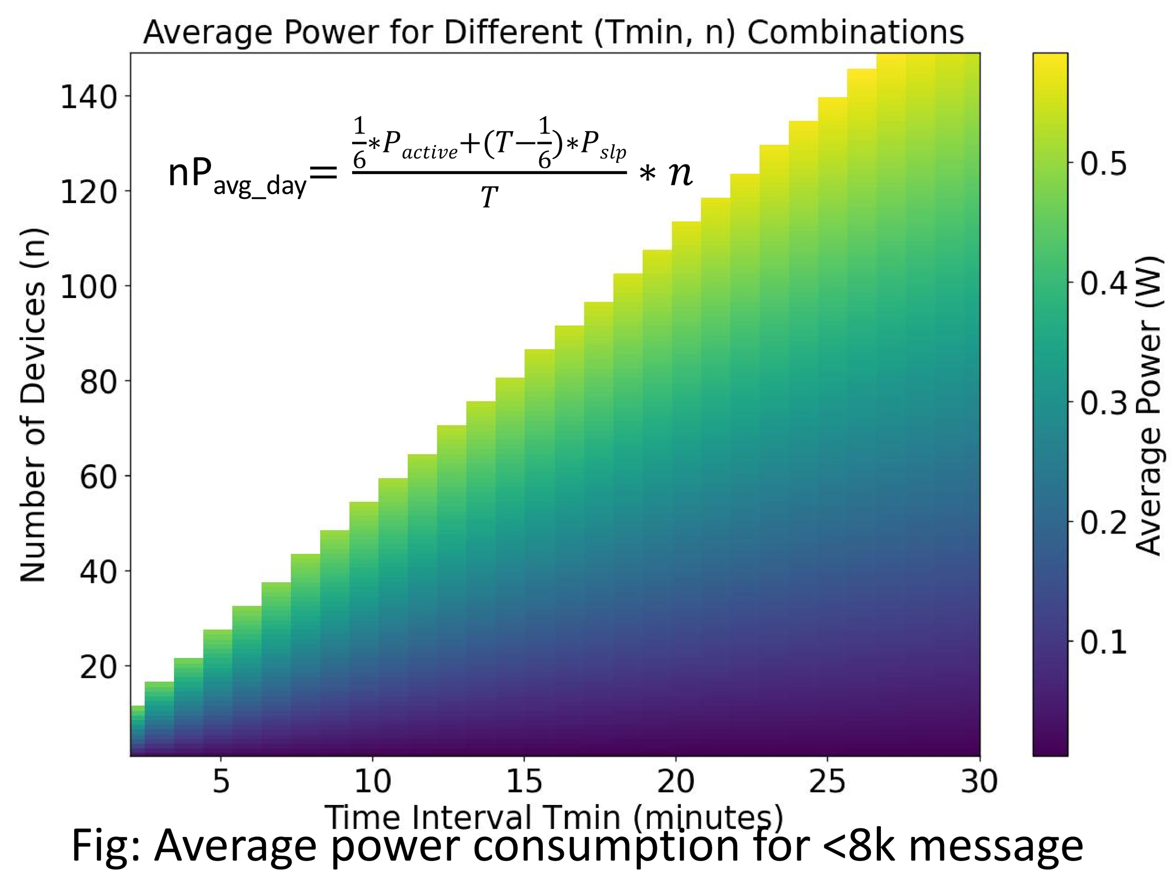


Fig: Average power consumption for <8k message

- Data Management and Quality Assurance:
  - Data from service bus queue is systematically written into a local database.
  - Successfully monitoring data during 42.8 hours continuous test period. There are 27 message loss, resulting in a data loss rate of 0.53%.
- Power consumption:
  - ESP32 works in XVC mode and sensor monitoring mode, in the latter mode, ESP32 switches periodically between normal operation and deepsleep.
  - The power with XVC run is 0.7W, the power for normal operation is 0.5W, the power with deepsleep is less than 1mW.
- Extra features:
  - Simultaneous Debugging of Multiple FPGAs.
  - Monitoring I2C sensors located on FPGA card via slighly modified JTAG connector pinout.