Porting SPEC drivers to current Linux kernels and embedded ARM boards for WRTD applications

Mathis MARION, Gwenhael GOAVEC-MEROU, Jean-Michel FRIEDT

FEMTO-ST Time & Frequency department, Besançon, France

October 7, 2021
Introduction: distributed passive RADAR

Principle of multistatic passive RADAR

- RADAR: RAdiofrequency Detection And Ranging provides target range (time of flight resolution inverse of bandwidth) and velocity (Doppler shift)
- multistatic: separate emitter and receiver(s) for spatial diversity ...
- ... requires time and frequency synchronization of all components.

Application to GRAVES ¹:

- Continuous Wave (CW) illumination of space over the French metropolitan territory for satellite detection
- bistatic RADAR with 143.05 MHz emitter located north-east of France and receiver located south of France
- >400 kW emitted power makes planes easily detected as frequency shift from carrier
  \[ \Delta f = 2f_0 \frac{v}{c} \simeq v @ 143.05 \text{ MHz} \]
- subsonic planes at \( v < 280 \text{ m/s} \) exhibit frequency shift up to 300 Hz at most

Multistatic RADAR: complement velocity (CW source) with direction of arrival (phase analysis)

\[ \Rightarrow \text{ requirement on phase synchronization: } 3^\circ \text{ phase } @ 143.05 \text{ MHz is } \frac{3}{360 \times 143.05} \mu s = 60 \text{ ps} \text{ within the capability of WR} \]

Moving target: Doppler shift

Need for stable local oscillator: $300 \text{ Hz} @ 143 \text{ MHz} = 2 \text{ ppm long term stability}$

- each moving target is identified with a different Doppler shift after FFT of the recorded signal
- low streaming datarate: 600 S/s sufficient, 1 kS/s safe
Moving target: Doppler shift

Need for stable local oscillator: 300 Hz @ 143 MHz = 2 ppm long term stability

- each moving target is identified with a different Doppler shift after FFT of the recorded signal
- low streaming datarate: 600 S/s sufficient, 1 kS/s safe
Phase analysis: direction of arrival (DoA)

- GRAVES emits CW ⇒ **no range** information
- frequency shift at each antenna = Doppler of the projected velocity vector
- phase difference between antennas:

\[
\Delta \varphi = \vec{k} \cdot \vec{d} = \frac{2\pi}{\lambda} d \cos \vartheta
\]

with \( d \) spacing between antennas and \( \vartheta \) angle between antenna baseline and signal direction of arrival
Initial architecture

Software Defined Radio (SDR) based multistatic RADAR system
▶ classical architecture: frequency transposition from radiofrequency band to baseband using a phase locked loop (PLL) generated local oscillator (LO)
▶ Analog to Digital Conversion (ADC) of the baseband signal, here at sampling frequency $f_s > 600$ Hz

**Problem:** phase synchronization of spatially distributed PLL LOs even if ADCs are time and frequency synchronized

**Solution:** aliasing

Ettus Research X310 Basic RX frontend (balun) feeding the 200 MS/s ADC detects the 143.05 MHz signal as $100 - 43.05 = 56.95$ MHz alias with $\approx 5.3$ dB loss (sinc transfer function)

ADC is time (1-PPS) and frequency (10 MHz) synchronized: functional demonstration of consistent Doppler shift and phase difference from colocated receivers.
Problem statement

Waste of resources:

- one Ettus Research X310 + White Rabbit Switch at each receiver node
- SPEC board allows for 100 MS/s acquisition in a White Rabbit synchronized board when fitted with 4-channel ADC (direction of arrival at each receiver station)
- replace dedicated PC with embedded board: Raspberry Pi4 OEM version (Compute Module 4) provides 1-line PCIe interface compatible with SPEC

Buildroot \(^2\) with latest Linux kernel is used for Raspberry Pi 4 binary image cross-compilation: can White Rabbit operate in such an environment?

WRTD on latest Linux kernels

WRTD (White Rabbit Trigger Distribution) provides exactly the targeted functionalities ...
... on paper

- diverse source of repositories for the various parts of the project
- runs on CentOS7 with 3.x kernels

First development: porting WRTD to a current (summer 2021: 5.12) Debian/stable distribution

Kernel development = multiple reboots when kernel panics ⇒ qemu for fast reboot when memory is corrupted: see
https://github.com/oscimp/WRTD-FMC-ADC/blob/main/qemu_setup/launcher.sh for qemu access to PCIe hardware

```
qemu-system-x86_64 -enable-kvm -cpu host -machine type=q35,accel=kvm -hda $HDA -device intel-iommu,caching-mode=on -device vfio-pci,host=$PCI_ID ... with PCI_ID=$(lspci | grep CERN | awk '{print $1}')
```
WRTD on latest Linux kernels

WRTD (White Rabbit Trigger Distribution) provides exactly the targeted functionalities ...
- diverse source of repositories for the various parts of the project
- runs on CentOS7 with 3.x kernels

```
Kernel development = multiple reboots when kernel panics ⇒ qemu for fast reboot when memory is corrupted: see
https://github.com/oscimp/WRTD-FMC-ADC/blob/main/qemu_setup/launcher.sh for qemu access to PCIe hardware
```

```
3 qemu-system-x86_64 -enable-kvm -cpu host -machine type=q35,accel=kvm -hda $HDA -device intel-iommu,caching-mode=on -device vfio-pci,host=$PCI_ID ... with PCI_ID=$(lspci | grep CERN | awk '{print $1}')
```
WRTD on ARM-based CPUs

- Challenge of software emulation accessing hardware: artifacts introduced by qemu?
- Replace qemu with another platform with short reboot time: CM4...
- ... requires testing support of PCIe on ARM architecture.

Problem: PCIe class code identifier not recognized ⇒ change identified in EEPROM (see https://github.com/oscimp/WRTD-FMC-ADC/tree/main/wrtd_installation/change_pci_class)

4https://github.com/oscimp/oscimp_br2_external with WRTD for Buildroot in packages
Functional WRTD on x86 & ARM-based computers

Functional acquisition demonstration but ... random phase between multiple channels collected by distributed boards (cross-correlation maximum of broadband noise)

The ADC (Si570) clock is not synchronized on the White Rabbit clock: although trigger time is common to all boards, the sampling rate and phase within 10-ns time slot is random
ADC clock synchronization

ADC clock synchronization on WRTD requires ...

- tuning ADC clock frequency
- compare ADC clock with White Rabbit clock
- control ADC clock on White Rabbit clock: soft-PLL

Current status: userspace programming capability of the ADC clock
Conclusion & perspectives

1. WRTD functional on kernel 5.12 (and possibly later) for x86, ARM & qemu
2. demonstration of remote trigger of analog acquisitions despite ADC free running clock

Resources:
- detailed documentation of this work at https://github.com/oscimp/WRTD-FMC-ADC
- https://github.com/oscimp/oscimp_br2_external with WRTD for Buildroot in packages

Work in progress:
- discipline ADC Si570 on White Rabbit clock (generalize soft-PLL in the FPGA?)
- datastream over White Rabbit dark fibers (functional with WRS ... to be verified with SPEC)