Software Defined Radio for Time and Frequency applications: example of passive monitoring of TWSTFT and other timing signals

Jean-Michel Friedt, jmfriedt@femto-st.fr

Software Defined Radio (SDR) is a paradigm aimed at minimizing the contribution of hardware and maximizing the contribution of software when processing radiofrequency signals (Fig. 1). While the concept is mostly used in the context of digital communication for agile systems flexible enough to adapt to unexpected conditions, accessing the raw samples right after the digitization of the radiofrequency wave provides opportunities for benefiting from information on the physical system before the signal has been processed. Not only does this access to the physical layer provide security (e.g. GNSS anti-spoofing and anti-jamming using null steering [1]), but also the opportunity to finely recover timestamps in the context of time and frequency transfer. Free, opensource software is readily distributed for reproduction of the experiment [2] and repetition of the measurements at remote sites, a core requirement of a scientific endeavor.

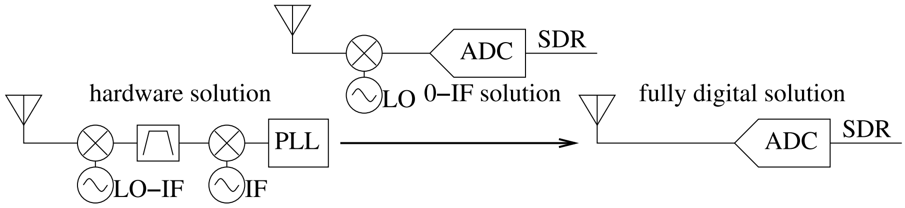

Figure 1: Middle: the hardware implementation of a super-heterodyne radiofrequency receiver, strongly application dependent. Right: the ultimate SDR, with the antenna signal sampled by the analog to digital converter (ADC) for software processing. Middle: the practical solution, with a first frequency transposition from radiofrequency band to baseband prior to ADC conversion.

Hardware is limited to a minimum and mostly comprised of a (complex, real and imaginary) frequency transposition mixer and a fast analog to digital converter. The software provides long term stability (a digital algorithm does not drift), reconfigurability for using the same hardware for multiple applications, and tunability. Furthermore, software implementation provides logging and remote control. The availability of huge non-volatile storage capacity with large bandwidth allows for post-processing with the analysis of as many channels as needed without constraint on computational power. However, the complex SDR framework including efficient hardware access, high performance computing and relying possibly on heterogeneous processing architectures (GP-CPU, GPU, FPGA) are challenging to maintain on the long term.

We will illustrate the use of SDR in various contexts of time and frequency applications, relying on the opensource SDR framework GNU Radio and the associated GNSS processing framework `gnss-sdr`, including reception of timing signals (GNSS, DCF77) and demonstrate the flexibility of the approach with the passive monitoring of the Two Way Satellite Time and Frequency Transfer (TWSTFT) signals transmitted from National Metrology Institutes to the Telstar11N satellite and then broadcast over Europe and North America. The issue of generating a timing signal in addition to analyzing and processing records will be tackled in the context of generating a 1-PPS output from a SDR GNSS receiver, demonstrating distributed computing architecture.

[1] W. Feng, J.-M Friedt, G. Goavec-Merou, F. Meyer, *Software Defined Radio Implemented GPS Spoofing and Its Computationally Efficient Detection and Suppression*, IEEE Aerospace and Electronic Systems Magazine **36**(3) 36–52 (2021)

[2] Oscillator Instability Measurement Platform (OscIMP) Digital repository at `https://github.com/oscimp/gr-satre` for SATRE TWSTFT signal reception or `https://github.com/oscimp/gnss-sdr-1pps/` for GNSS-controlled 1 PPS generation.