Demo: ISAC Real-Time Experimentation Platform

Rafael Ruiz, Marco Canil, Jesus O. Lacruz, Joerg Widmer
IMDEA Networks Institute
Madrid, Spain
{rafael.ruiz, marco.canil, jesusomar.lacruz, joerg.widmer}@networks.imdea.org

Abstract

Integrated Sensing And Communication (ISAC) is a key enabler for future wireless systems, yet realizing it in practice requires executing tightly coupled communication and sensing pipelines under strict real-time constraints. In particular, maintaining a continuous high-rate communication link while extracting sensing features introduces fundamental challenges in computation partitioning and latency management across hardware platforms.

In this demonstration, we present a real-time Millimeter-Wave (mmWave) ISAC system which exposes and addresses these challenges through a cross-layer hardware/software design. Our approach preserves a standard 5G-like communication pipeline and enables sensing by appending bursts of reference signals for beam-sweeping-based channel probing, requiring only minimal modifications to the payload structure. The system partitions the processing across the FPGA logic, an embedded ARM processor, and an external host, enabling synchronous transmission, real-time channel extraction, and continuous sensing without disrupting the communication link. We demonstrate an end-to-end system where Channel State Information (CSI) is extracted *on-device* and streamed for real-time processing and visualization of micro-Doppler signatures from moving targets.

Keywords

6G, ISAC, Wireless Experimentation, SDR, FPGA

1 Introduction

ISAC is a key enabler for 6G systems, where wireless networks not only communicate but also sense the physical environment. While the majority of the works have explored ISAC from a signal processing and theoretical perspective, translating these concepts into real-world systems remains challenging. In practice, enabling simultaneous communication and sensing requires executing both pipelines under strict real-time constraints, where latency, synchronization, and computation placement become critical bottlenecks.

A fundamental challenge lies in determining how to partition ISAC processing across heterogeneous platforms. Executing all processing on embedded platforms is often infeasible, while excessive offloading introduces latency that can disrupt real-time operation and degrade the communication link. As a result, building a system that maintains a continuous high-rate communication link while extracting sensing features in real time is a non-trivial systems problem.

In this demonstration, we present an end-to-end ISAC experimentation platform that explicitly addresses these challenges. Built on the HELIX testbed [2], our system preserves a standard 5G-like communication pipeline and enables sensing through minimal modifications to the payload, appending bursts of reference signals to support beam-sweeping based channel probing. This design allows sensing to be integrated without disrupting the underlying communication link.

To meet real-time constraints, we adopt a distributed processing architecture across the FPGA logic, an embedded ARM, and a host PC. The FPGA performs latency-critical tasks including synchronous packet generation, beam steering, and CSI estimation. The embedded processor orchestrates the measurement loop and aggregates channel data, while micro-Doppler extraction is offloaded to a host PC.

We demonstrate a real-time mmWave ISAC system where the communication link is maintained while sensing is performed concurrently. The system extracts CSI on-device, streams it for further processing, and visualizes micro-Doppler signatures of moving targets in real time.

2 System Architecture

Our system is designed to support real-time ISAC by partitioning the processing pipeline across heterogeneous computation domains, namely FPGA logic, an embedded ARM Processing System (PS), and a host PC. This split is driven by the latency and computational requirements of each stage.

The overall architecture consists of two independent nodes equipped with an RFSoC board and mmWave front-ends, operating in a monostatic configuration. The end-to-end pipeline is divided into synchronous transmission and beam sweeping, hardware-accelerated channel extraction and data aggregation, and offloaded sensing processing and visualization, as detailed in the following paragraphs.

2.1 Synchronous Transmission and Beam Sweeping

To enable sensing while preserving the communication link, we keep a standard 5G-like frame structure and introduce only minimal modifications to the payload. Specifically, bursts of reference signals are appended at the end of each slot, enabling channel probing without disrupting ongoing data transmission. A hardware scheduler implemented in the FPGA ensures deterministic packet generation and precise timing control. It synchronizes baseband transmission with antenna beam steering through GPIO signals, allowing beam switching to happen *exactly* at OFDM symbol boundaries. During each sensing burst, the transmitter sweeps across a set of beam patterns, covering the angular field of view.

2.2 Hardware Channel Extraction and Data Aggregation

Operating in a monostatic configuration, the receiver captures reflections of the transmitted signals to estimate the wireless channel.

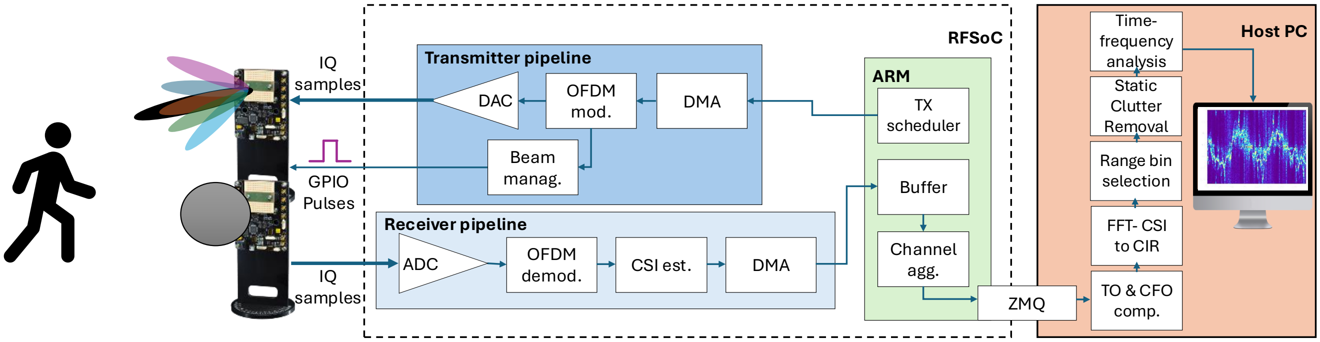

Figure 1: Top-level architecture

The FPGA performs CSI estimation in real-time, meeting the latency constraints imposed by the communication pipeline. The extracted CSI is forwarded to the embedded PS, which orchestrates the measurement loop and aggregates the channel estimates into temporal windows. These windows are constructed over multiple transmission slots to enable subsequent sensing processing. The PS streams these aggregated CSI windows over Ethernet to a host PC using the ZeroMQ (ZMQ) [3] messaging framework.

2.3 Offloaded Processing and Real-Time Visualization

To enable micro-Doppler extraction, the system offloads aggregated CSI windows to an external host PC. This design choice reflects a key trade-off: while offloading introduces additional latency, it enables the execution of computationally intensive algorithms. On the host, the received CSI is processed to derive sensing features. First, residual timing and phase offsets are compensated for. Then, a transformation to the delay domain is performed to obtain Channel Impulse Responses (CIRs). Relevant delay bins are then selected, and static components are suppressed to isolate dynamic targets. Finally, time-frequency analysis is applied to extract micro-Doppler signatures, which capture the motion characteristics of the targets. The processed sensing outputs are visualized in real time, enabling continuous monitoring of target activity while the communication link remains active.

3 Demonstration

We demonstrate a real-time mmWave ISAC system using two RFSoC-based nodes with mmWave front-ends. The transmitter maintains a continuous 5G-like communication link while performing beam sweeping at the end of each slot, as previously described, and the receiver captures channel reflections for sensing. The system is orchestrated via PYNQ [1] on the embedded processor, which extracts and streams CSI to a host PC. The host processes the data in real time to extract micro-Doppler signatures, which are visualized live. Attendees can move within the sensing area and observe how their motion is reflected in the micro-Doppler signatures, demonstrating simultaneous communication and sensing in real time[1]. In our implementation, each aggregated data block sent from the RFSoC to the host PC consists of 128 CSIs, which are used to compute a single column (i.e., one time instant) of the micro-Doppler signature. Using a MacBook Pro connected to the RFSoC via Ethernet, the end-to-end processing time—including signal transmission and reception, processing at the RFSoC, data transfer over the network, host-side processing, and visualization—is on average 73.11 ms (standard deviation 0.62 ms, maximum 78.10 ms, minimum 70.13 ms, computed over 1000 frames), corresponding to an effective frame rate of approximately 13–14 Hz.

4 Acknowledgments

This publication is part of grant JDC2024-055419-I, funded by MICIU/AEI/10.13039/501100011033 and by the ESF+.

The work presented in this paper is (partially) supported by the MultiX project that has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101192521.

This work has been funded by project PID2022-136769NB-I00 funded by MCIN/AEI/10.13039/501100011033 / FEDER, UE.

It has also been supported by the Comunidad de Madrid through the DISCO6G-CM project (TEC-2024/COM-360) and TUCAN6-CM (TEC-2024/COM-460), funded under ORDEN 5696/2024.

References

[1] Advanced Micro Devices, Inc. 2025-09-24. PYNQ. https://pynq.io. Version 3.1.2.

[2] Rafael Ruiz, Jesus O. Lacruz, Bastian Bloessl, Matthias Hollick, and Joerg Widmer. 2025. HELIX: High-speed Real-Time Experimentation Platform for 6G Wireless Networks. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services* (Hilton Anaheim, Anaheim, CA, USA) *(MobiSys '25)*. Association for Computing Machinery, New York, NY, USA, 305–318. doi:10.1145/3711875.3729152

[3] The ZeroMQ Authors. 2026. ZeroMQ: An open-source universal messaging library. https://zeromq.org. Accessed: 2026-04-06.

---

[1] The video of the demo is available at https://youtu.be/Gct3IAKId6s