

Real-Time Navigation Sensor Synchronization via Programmable Logic and SoC Architecture

Adonai Hernández Bolaños

Institute for Applied Microelectronics (IUMA)
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
adonai.hernandez101@alu.ulpgc.es

Héctor Navarro Botello

I+D department
Aerolaser Systems
Agüimes, Spain
hnavarro@aerolaser.es

Alfonso Medina Escuela

Institute for Applied Microelectronics (IUMA)
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
aescuela@iuma.ulpgc.es

Abstract—Accurate temporal synchronization between GNSS and inertial measurement units (IMUs) is a critical requirement for real-time navigation and sensor fusion applications, particularly when integrating with auxiliary sensors such as LiDAR. This work presents the design, implementation, and experimental validation of a low-cost GNSS+IMU data acquisition and synchronization system built on a Zynq-7000 System-on-Chip (SoC), combining custom RTL blocks with embedded firmware running on FreeRTOS. The proposed system uses the Pulse-Per-Second (PPS) signal as the primary synchronization reference, latched deterministically in hardware and propagated to downstream acquisition blocks. The system's performance was characterized through static laboratory tests and validated in a field trial against a professional-grade reference system (GNSS + IMU) mounted on a LiDAR-equipped vehicle. Results show a heading RMS error of 1.58° and a 2D position RMS error of 0.56 m relative to the reference trajectory, along with qualitative agreement in the reconstructed point clouds. These results indicate that the proposed low-cost architecture can deliver navigation data of sufficient quality for real-time preview applications.

Index Terms—GNSS/IMU synchronization, Pulse-Per-Second (PPS), System-on-Chip (SoC), FPGA, sensor fusion, real-time data acquisition, LiDAR

I. INTRODUCTION

Light detection and ranging (LiDAR) is a technique that generates dense point clouds by measuring the time-of-flight of laser pulses reflected off the surrounding environment [2]. This technology has become a key part of high-resolution 3D mapping, autonomous navigation, and infrastructure inspection. However, a LiDAR sensor by itself only provides measurements relative to its local reference frame. In mobile mapping scenarios, where the LiDAR is mounted on a moving vehicle, each individual scan must be transformed into a common, absolute reference frame in order to produce a coherent and accurate point cloud. This transformation requires precise knowledge of the sensor's position and orientation.

This georeferencing process is what makes the integration of GNSS and an Inertial Measurement Unit (IMU) indispensable [1]. The GNSS provides absolute position information at a low rate (1–20 Hz), while the IMU provides high-rate (up to 2000 Hz) orientation and motion data that fills in the gaps between GNSS updates and compensates for vehicle dynamics. The pose estimation resulting from the fusion of

these sensors is then used to project LiDAR points from the sensor's local frame into a global coordinate system. This projection is only as accurate as the temporal alignment between the three data streams.

In this work, we present the design, implementation, and experimental validation of a low-cost GNSS+IMU data acquisition and synchronization system built on a Xilinx Zynq-7000 SoC, combining custom RTL logic in the programmable logic (PL) with embedded firmware running on FreeRTOS in the processing system (PS), intended for integration with a LiDAR sensor. The proposed architecture uses the PPS signal as the primary synchronization reference: it latches a free-running counter and timestamps every GNSS packet, while validating the current count against the frequency of the internal clock.

Finally, the system was validated in a field trial using a professional-grade acquisition and synchronization system as a reference, with both systems mounted on a LiDAR-equipped vehicle.

II. SENSORS

The GNSS receiver used in this work is a u-blox ZED-F9P (EVB-C099-F9P evaluation board), a multi-band receiver communicating over the UBX binary protocol and providing position and velocity updates at 4 Hz. In addition to this navigation data, the receiver outputs a hardware Pulse-Per-Second (PPS) signal aligned to the GNSS time reference, which is used as the primary synchronization signal for the entire system.

Inertial measurements are provided by an Epson M-G366 IMU, a six-axis unit combining a triaxial gyroscope and a triaxial accelerometer with internal temperature compensation, interfaced over UART. The sensor is configured to output attitude information directly at 200 Hz, suitable for real-time use, and each sample is timestamped using the Data Ready Signal (DRDY) with the same hardware counter used for the GNSS, ensuring that both sensor streams share a common temporal reference despite their different native sampling rates and communication protocols.

III. SYSTEM ARCHITECTURE

The proposed system is implemented on a Zybo Zynq-7000 development board, based on a Xilinx Zynq-7000 System-on-

Chip that combines a dual-core ARM Cortex-A9 Processing System (PS) with FPGA-based Programmable Logic (PL) on a single die. This architecture allows timing-critical operations to be implemented as dedicated hardware in the PL, while higher-level data management, storage, and networking tasks are handled by embedded software running on the PS under FreeRTOS.

The PPS signal generated by the GNSS receiver is routed directly into the PL, where it is used to latch a dedicated hardware counter and to validate the internal oscillator frequency. The counter value is then propagated to the other IP cores: a custom AXI-Lite FIFO that stores the latched PPS timestamps so the firmware can correlate the GNSS packets with the most recent PPS event, and a custom AXI master that captures the IMU data, which is latched using the IMU's DRDY signal and written, together with the corresponding hardware timestamp, into a ping-pong buffer in Block RAM (BRAM).

On the PS side, the system is structured as a set of concurrent FreeRTOS tasks: dedicated tasks for GNSS and IMU acquisition, and a storage task responsible for writing timestamped sensor data to a microSD card through the FatFs file system, coordinated through queues and a shared-resource mutex to guarantee safe concurrent access. A separate networking task exposes a WebSocket-based interface, allowing the system to be monitored in real time and enabling captured files to be retrieved over the network without removing the storage card. This division of responsibilities allows the system to meet hard timing requirements in hardware while retaining the flexibility of software for storage, monitoring, and post-processing integration.

IV. RTL

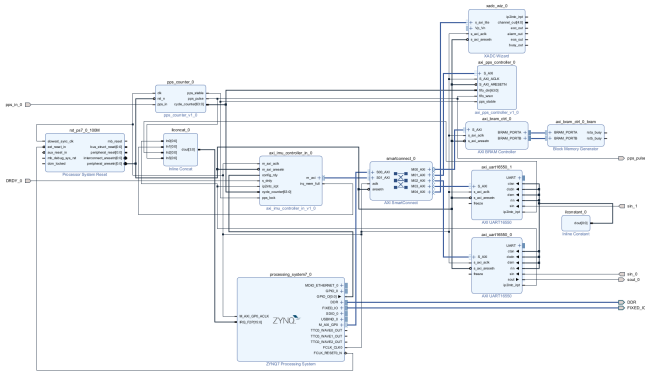


Fig. 1. Vivado block design of the proposed system.

As shown in the block design of Fig. 1, the programmable logic (PL) implements three custom hardware blocks:

- **PPS_COUNTER** It receives the PPS signal from the GNSS receiver and passes it through a two-flip-flop synchronizer to mitigate metastability before use in the FPGA clock domain. On the synchronized edge, the module latches the current value of a free-running 100 MHz counter into a timestamp register, which serves as

the common hardware time base shared by all downstream acquisition blocks. An FSM validates the count between consecutive PPS pulses; if the count falls within 100,000,000 clock cycles $\pm 1\%$, a lock signal (PPS lock) is asserted.

- **PPS_AXI_CONTROLLER** Each latched PPS timestamp is pushed into an AXI-Lite slave FIFO, exposed as a memory-mapped peripheral to the processing system. The GNSS driver reads the FIFO to retrieve the timestamp corresponding to the most recent PPS edge and associates it with the incoming UBX message stream — specifically the PVT, RAWX, and SFRBX packet types — as each is parsed and stored.
- **AXI_IMU_CONTROLLER** IMU data is captured with a dedicated AXI4 master burst-write FSM, which reads incoming packets from an AXI UART16550 peripheral connected to the IMU and writes them, together with the corresponding hardware timestamp and PPS lock status, into a ping-pong Block RAM (BRAM) buffer. Each stored record follows a fixed packet layout combining the IMU payload (accelerometer, gyroscope, and quaternion attitude data) [3], a 64-bit hardware cycle counter, and PPS lock status flags. Burst transfers are sized in 16-word blocks to align with the AXI4 4 KB address boundary restriction, avoiding split transactions and simplifying the master state machine. When 256 packets have been stored in the BRAM, an output signal connected to the PS interrupt input is driven high to notify the firmware that it can begin downloading the 256 packets from the BRAM.

This design reflects a deliberate asymmetry between the two acquisition paths: the high-rate IMU stream is captured through a dedicated AXI4 master and BRAM buffering to avoid data loss under bursty PS load, while the low-rate GNSS stream is handled entirely in software, relying on the AXI-Lite FIFO only to obtain an accurate hardware timestamp rather than to buffer the GNSS data itself. Both paths ultimately reference the same 100 MHz hardware counter, ensuring a common time base across sensors despite their different capture strategies.

V. FIRMWARE

The processing system runs FreeRTOS, structured as a set of concurrent tasks that mirror the software architecture shown in Fig. 2.

The IMU Controller task is interrupt-driven, triggered by the AXI4 master datapath described in Section IV whenever a new IMU record is available; it retrieves data from the ping-pong buffer over the AXI bus and forwards it to the Storage Task. In parallel, the GNSS Controller task reads incoming UBX packets from the receiver over UART into a stream buffer, associates each packet with the corresponding hardware PPS timestamp obtained from the AXI-Lite FIFO (Section IV), and pushes the timestamped data into its own ping-pong buffer before handing it off to the Storage Task. A dedicated task also samples the Zynq-7000's internal XADC to log system

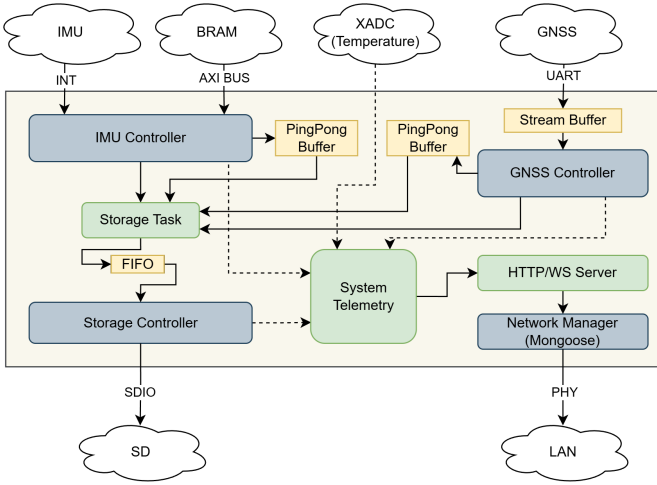


Fig. 2. Software architecture of the embedded firmware.

temperature alongside the sensor data, providing context for thermal drift analysis in post-processing.

The Storage Task acts as the central synchronization point, collecting timestamped records from the IMU Controller, the GNSS Controller, and the XADC task, and forwarding them through a FIFO to the Storage Controller, which writes the data to a microSD card over SDIO using the FatFs file system. Decoupling the Storage Task from the Storage Controller through this FIFO absorbs the variable latency of SD card write operations without blocking the higher-priority acquisition tasks.

Independently of the storage path, a System Telemetry task aggregates status information from the IMU Controller, GNSS Controller, XADC, and Storage Controller — including PPS lock status, buffer occupancy, and temperature — and exposes it to an HTTP/WS server, served over Ethernet through a lightweight network manager (Mongoose). This provides real-time visibility into system state and allows captured files to be retrieved over the network without physically removing the SD card, without interfering with the deterministic timing of the acquisition and storage tasks.

VI. VALIDATION

The system was evaluated in two stages: static laboratory characterization of the PPS latching and timestamping mechanism, followed by a field trial assessing overall navigation accuracy under real operating conditions.

A. Timing Characterization

Prior to the field trial, the synchronization mechanism was characterized in a static laboratory setup. A 30-minute recording was captured and subsequently analyzed using Python scripts; the results of the timing analysis are presented below.

Table I shows that the current synchronization mechanism is capable of delivering consistent sub-microsecond accuracy for every timestamped event. This level of precision is required for direct LiDAR georeferencing.

TABLE I
TIMING JITTER CHARACTERIZATION

Parameter	PPS Signal	DRDY Signal
Frequency	1 Hz	200 Hz
Tick Resolution	10 ns	10 ns
Jitter RMS	189 ns	165 ns

B. Field Trial

To validate the system in a field trial, a professional-grade acquisition and synchronization system was used as a reference. This reference system consists of a tactical-grade fiber-optic gyroscope (FOG) IMU (iMAR iIMU-FSAS) and a professional-grade GNSS receiver (Javad TR-3S). A comparison of the two IMUs is given below:

TABLE II
PARAMETER COMPARISON

Parameter	EPSON G366	iMAR iIMU-FSAS
Technology	QMEMS	FOG
Gyro Bias Instability	1.2 °/h	< 0.1 °/h
Gyro Bias	360 °/h	0.75 °/h
Random Walk (ARW)	0.08 °/√h	0.15 °/√h
Gyro g effect	18 (°/h)/g	Negligible (FOG)
Accel Bias Instability	24 μg	< 10 μg
Accel Range	±8 / ±16 g	±5 g (optional 10/20 g)

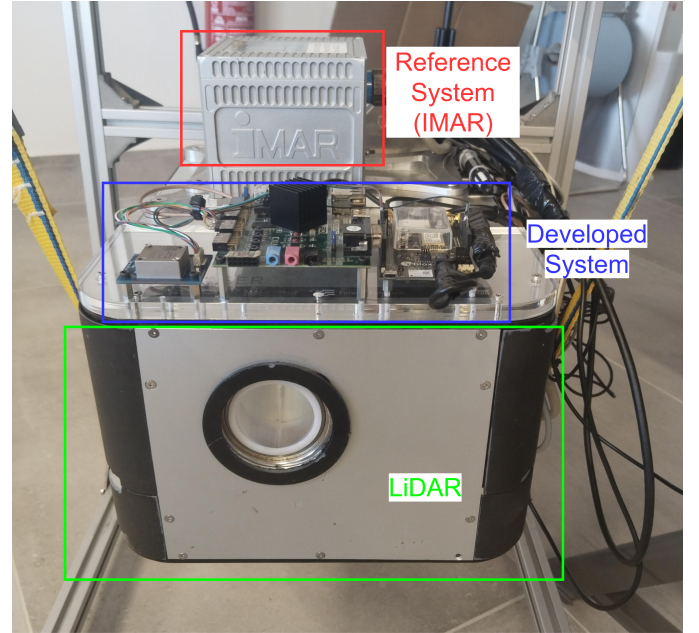


Fig. 3. Final assembly

The field trial consisted of a LiDAR capture performed with a ground vehicle (car); both acquisition and synchronization systems were assembled alongside the LiDAR and mounted on the roof of the vehicle. The selected route was the GC-104 road.

LiDAR data was then captured using both acquisition and synchronization systems. Afterward, all data was processed



Fig. 4. Route followed by the vehicle.

and converted into a format compatible with professional navigation software, and a loosely coupled fusion algorithm [3] was applied to obtain position and attitude for each IMU record.

The resulting position and attitude data were verified against the iMAR system, yielding the following errors:

TABLE III
STATISTICAL METRICS OF THE HEADING ERROR (EPSON VS. IMAR)

Statistic	Heading Error ($^{\circ}$)
Mean (Bias)	1.0892
Standard Deviation (1σ)	1.1484
RMS	1.5828
Maximum Absolute	7.0463
Minimum Absolute	0.0008

Table III summarizes the descriptive statistics of the Epson system's heading error relative to the iMAR reference throughout the entire capture. The mean (bias) of 1.089° indicates the presence of a systematic bias, suggesting the possible existence of an uncorrected angular misalignment (boresight) between the two systems [4]. The standard deviation ($1\sigma = 1.148^{\circ}$) quantifies the dispersion of the error after discounting this bias, that is, the random component of the error. The RMS (1.583°) is the most representative metric of the total expected error.

TABLE IV
STATISTICAL METRICS OF THE 2D SEPARATION (EPSON VS. IMAR)

Statistic	2D Separation (m)
Mean (Bias)	0.5574
Standard Deviation (1σ)	0.0815
RMS	0.5633
Maximum Absolute	0.7259
Minimum Absolute	0.3255

Unlike the behavior observed in attitude, the 2D separation metrics in Table IV reveal that the main component of positional error corresponds to a static bias (0.557 m), characterized by a very reduced dispersion ($\sigma = 0.081$ m).

Finally, the LiDAR point cloud was generated using the captured data. The one shown in Fig. 5 is a high-voltage tower, captured from a distance of approximately 200 meters; good overall quality was obtained.



Fig. 5. LiDAR reconstruction of a power transmission tower

VII. CONCLUSIONS

The results obtained in the field tests experimentally validate the acquisition and synchronization system implemented in this work.

This work addresses the design of a low-cost acquisition and synchronization platform capable of providing data with the precision required for real-time visualization.

Therefore, the differences observed in the comparisons against the reference system are likely attributable to differences in sensor quality and physical misalignment (boresight errors) that have not yet been characterized [4], although this hypothesis remains pending confirmation through the tests proposed in future lines of research.

REFERENCES

- [1] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Boston, MA: Artech House, 2013.
- [2] J. Shan and C. K. Toth, *Topographic Laser Ranging and Scanning: Principles and Processing*, 2nd ed. Boca Raton, FL: CRC Press, 2018.
- [3] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
- [4] J. Skaloud and D. Lichti, "Rigorous approach to bore-sight self-calibration in airborne laser scanning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 61, no. 1, pp. 47–59, 2006.