electronics

MDPI

Article

Intelligent Embedded Systems Platform for Vehicular Cyber-Physical Systems

Christopher Conrad *, Saba Al-Rubaye and Antonios Tsourdos

Centre for Autonomous and Cyberphysical Systems, Cranfield University, Cranfield MK43 0AL, UK; s.alrubaye@cranfield.ac.uk (S.A.-R.); a.tsourdos@cranfield.ac.uk (A.T.)
* Correspondence: christopher.conrad@cranfield.ac.uk

Abstract: Intelligent vehicular cyber-physical systems (ICPSs) increase the reliability, efficiency and adaptability of urban mobility systems. Notably, ICPSs enable autonomous transportation in smart cities, exemplified by the emerging fields of self-driving cars and advanced air mobility. Nonetheless, the deployment of ICPSs raises legitimate concerns surrounding safety assurance, cybersecurity threats, communication reliability, and data management. Addressing these issues often necessitates specialised platforms to cater to the heterogeneity and complexity of ICPSs. To address this challenge, this paper presents a comprehensive CPS to explore, develop and test ICPSs and intelligent vehicular algorithms. A customisable embedded system is realised using a field programmable gate array, which is connected to a supervisory computer to enable networked operations and support advanced multi-agent algorithms. The platform remains compatible with multiple vehicular sensors, communication protocols and human–machine interfaces, essential for a vehicle to perceive its surroundings, communicate with collaborative systems, and interact with its occupants. The proposed CPS thereby offers a practical resource to advance ICPS development, comprehension, and experimentation in both educational and research settings. By bridging the gap between theory and practice, this tool empowers users to overcome the complexities of ICPSs and contribute to the emerging fields of autonomous transportation and intelligent vehicular systems.

Keywords: intelligent system; cyber-physical system; vehicular; development platform; networking

check for updates

Citation: Conrad, C.; Al-Rubaye, S.; Tsourdos, A. Intelligent Embedded Systems Platform for Vehicular Cyber-Physical Systems. *Electronics* **2023**, *12*, 2908. https://doi.org/10.3390/electronics12132908

Academic Editor: Yolanda Blanco Fernández

Received: 26 May 2023
Revised: 29 June 2023
Accepted: 30 June 2023
Published: 2 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

This section introduces intelligent cyber-physical systems (ICPSs) as a catalyst for advancing intelligent and autonomous transportation. For brevity, subsequent references to ICPSs will thereby imply intelligent vehicular cyber-physical applications. ICPS literature is first reviewed to determine the core methods used to develop ICPSs and summarise the barriers impeding their widespread adoption in vehicular applications. These challenges highlight the need for a simple yet comprehensive platform to facilitate ICPS development and promote research in intelligent and autonomous transportation systems. Specifically, field programmable gate arrays (FPGAs) are identified as a robust technology to realise an adaptable network of embedded systems to explore, develop and test ICPSs and intelligent vehicular algorithms. This work thereby prototypes an FPGA-based embedded system that serves as a versatile tool to advance ICPSs in both educational and research settings.

1.1. Intelligent Vehicular Cyber-Physical Systems

ICPSs are an advanced technology that bring together the physical and digital worlds through the use of sensors, actuators, and intelligent software. These systems are designed to interact with a physical environment, collect data from sensors, and analyse that data in real-time using advanced algorithms and machine learning (ML) techniques. Consequently, ICPSs can make intelligent decisions and execute actions based on incoming data [1]. ICPSs have the ability to transform various industries such as manufacturing, transportation, and

energy. They can enhance efficiency, reduce costs, and improve safety in these industries by providing real-time data and intelligent insights to help optimise the underlying system processes. Nonetheless, Rani et al. [2] emphasize that the increased connectivity of ICPSs also brings about significant privacy and security concerns. As ICPSs become more prevalent, it is crucial to ensure that they are accurate, reliable, secure, and protected from cyberattacks. Despite these risks, ICPSs represent the next frontier of technology, with the potential to revolutionise the way we live, work and commute in the future [3].

Raj et al. [4] conclude that the integration of Internet of Things (IoT) with ICPSs is paramount for the advancement of transportation systems in smart cities. The authors assert that autonomous ground vehicles can significantly enhance the efficiency of traffic management systems, thereby improving road safety by establishing a smart and collaborative transportation infrastructure. Similarly, Mahrez et al. [5] confirm that real-time intelligent transportation systems (ITSs) safeguard vulnerable road users by employing collision warning systems, speeding alerts, safety indicators, and enhanced vision, radar and navigation systems. Moreover, they emphasize the potential of ITSs in predicting passenger, driver and traffic behaviour, which facilitates more efficient traffic management and routing algorithms. The authors also present ITSs as a key enabler of electric vehicles with varying degrees of autonomy, ranging from assisted driving to semi-automated or fully-autonomous vehicular systems. Liu [6] further presents a business case for autonomous vehicles equipped with infrastructure-to-vehicle communication capabilities, and concludes that a collaborative ITS is more efficient and cost-effective than relying solely on on-board autonomy. Similarly, Pan et al. [7] showcase a smart cloud-based commuting system that uses collaborative autonomous vehicles to improve passenger experience, vehicle utilisation and system efficiency within an ITS.

Nevertheless, the implementation of an ITS poses significant challenges. The transportation cyber-physical system (CPS) is a complex entity encompassing diverse physical components, including vehicles, road infrastructure, human drivers, machines, and sensors. In particular, the topology of the vehicular transportation network is in constant flux due to the high-speed movement of vehicles [8]. Moreover, human driver behaviour directly impacts the network topology, as drivers can join or exit the network at any given time. The density of vehicles within the transportation CPS also varies depending on the location. Furthermore, vehicular networks distinguish themselves from other wireless networks by boasting virtually unlimited power, storage, and computing capabilities [9]. Additionally, most existing wireless access technologies are not specifically designed to accommodate fast-moving vehicles. In the vehicular CPS, low latency is crucial for safety-critical applications that require the timely transmission of emergency messages, but bandwidth limitations arise, especially when considering multimedia and infotainment transmissions.

ICPSs play a critical role in enabling autonomous vehicles within the transportation CPS. Chen et al. [10] conduct a systematic review of computing and communication systems proposed for intelligent vehicles. Specifically, they divide the vehicular architecture into power and thermal management, storage management, security features, communication systems, computational hardware, and a variety of sensors and algorithms used to enable advanced perception, localisation, control and decision making. The authors identify light detection and ranging (LiDAR), radar, cameras, global navigation satellite systems (GNSS) and ultrasound sensors as the most commonly employed sensing systems within autonomous vehicles. They also emphasize the importance of robust communication protocols for vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-cloud/network (V2C/N) communication. Similarly, Novak et al. [11] review the relationship between autonomous driving behaviours and the underlying sensing and computing technologies of an ICPS within an ITS framework. Additionally, Kliestik et al. [12] review state-of-the-art sensor fusion techniques and perception algorithms that enable autonomous vehicles in an ITS. Consequently, a development platform for ICPSs must remain compatible with typical sensing and communication systems, while facilitating research in each architectural layer of an intelligent vehicle.

The conclusions drawn for autonomous ground vehicles and self-driving cars can be extended to airborne vehicles, such as unmanned aircraft systems (UASs) and air taxis. In fact, the emerging fields of advanced air mobility (AAM) and UAS traffic management (UTM) envision a comprehensive aerial ITS for passenger and cargo deliveries. Pan et al. [13] systematically review advancements in flying car transportation systems and discuss the potential of AAM to revolutionise the transportation sector. Dense AAM operations, however, require advanced communication, navigation and surveillance (CNS) capabilities, robust cybersecurity systems and enhanced collaborative decision-making frameworks. Specifically, ICPSs for AAM impose stricter requirements compared to ground-based vehicles due to the greater safety risks associated with aerial operations. Nonetheless, the underlying system architecture remains analogous to that of traditional transportation systems. In fact, Zhou et al. [14] propose a computation and communication framework for UASs, and highlight the importance of collaborative V2V and V2I communication in improving the efficiency of the ITS. Furthermore, Shrestha et al. [15] emphasize the necessity of reliable communication protocols for widespread adoption of AAM, suggesting that 6G-enabled ecosystems will eventually supersede current 5G proposals.

Javaid et al. [16] review collaborative communication technologies for UASs as a key enabler of multi-UAS systems. Furthermore, Wilson et al. [17] review the sensors, communication technologies, and ML techniques commonly employed in UAS applications. Typical UASs incorporate camera, LiDAR, radar, ultrasonic, and inertial measurement unit (IMU) sensors, similar to ground-based vehicles. Additionally, wireless-fidelity (Wi-Fi), Bluetooth, long-term evolution (LTE) and long-range (LoRa) communication technologies are found to be prevalent in UAS applications. These findings further reinforce the hypothesis that a robust ICPS development platform must be compatible with a diverse range of common sensors and communication technologies.

1.2. Intelligent Vehicular Cyber-Physical System Methods

ICPS methods are employed to design, develop, and operate the systems that integrate physical processes with digital technologies. These methods are essential for creating safe, reliable, and efficient CPSs within the transportation CPS [18–21]. By integrating computing, communication, and control with physical tasks, CPSs can often optimise industrial operations to achieve improved reliability, efficiency and flexibility [22]. Moreover, the ability to monitor and control operations in real-time often results in quicker fault resolution and improved safety, especially in the context of autonomous vehicles [1]. In fact, CPSs serve as the foundation of Industry 4.0, a term used to describe increased automation, connectivity and data exchange in industrial processes. Some commonly used CPS methods are hereby presented:

- Model-Based Design: This approach involves developing mathematical models of the physical system and using them to design the control algorithms and software. The models can be simulated to assess system performance prior to implementation;
- Control Theory: Control theory is used to design control systems that regulate physical processes and maintain stability. It involves analysing the system dynamics and developing feedback control algorithms to adjust system behaviour;
- Real-Time Systems: Real-time systems ensure that a CPS responds to changes in the physical environment in a timely and predictable manner. This involves developing software that can execute in real-time and meet system timing requirements;
- Cybersecurity: Cybersecurity is essential for safeguarding CPSs from cyberattacks that can compromise their safety and reliability. CPS methods for cybersecurity include the development of secure communication protocols, intrusion detection systems, and access control mechanisms;
- Machine Learning: ML is used to develop ICPSs that can adapt to changing environments and optimise their performance. This involves training algorithms using data from sensors and other sources to make intelligent predictions and decisions.

Notably, on-going developments in artificial intelligence (AI), ML and cloud computing foster the design of intelligent systems capable of managing multiple intelligent sensors embedded in ICPSs, as depicted in Figure 1 [23]. These systems support fault prediction, increased autonomy and self-healing or self-adapting algorithms for safer and more reliable implementations. Additionally, embedded AI solutions can optimise energy usage and bandwidth utilisation of interconnected agents, leading to improved system performance and efficiency. Shuvo et al. [24] conduct a review of deep learning techniques applied to edge devices such as CPSs, IoT, autonomous systems, embedded hardware, and intelligent sensors. They highlight the challenges associated with optimising AI algorithms for embedded systems, often addressed through novel algorithms, improved optimisation, and more effective hardware acceleration or hardware–software co-design. Consequently, the evolution of ICPSs is inevitable to support the increased levels of autonomy and integration demanded by industry. A robust development platform for ICPSs must therefore facilitate the deployment of intelligent algorithms at both embedded and networked levels.

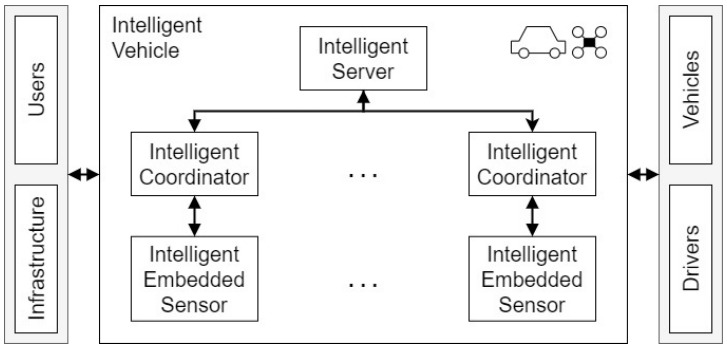

Figure 1. Generic architecture of an ICPS.

1.3. Cyber-Physical Challenges and Technical Barriers

Despite the benefits of ICPSs, several challenges and technical barriers hinder their widespread adoption, as outlined in Figure 2. Pundir et al. [25], in fact, suggest that the heterogeneity and complexity of the transportation CPS makes an ITS particularly susceptible to cyber vulnerabilities and technological challenges.

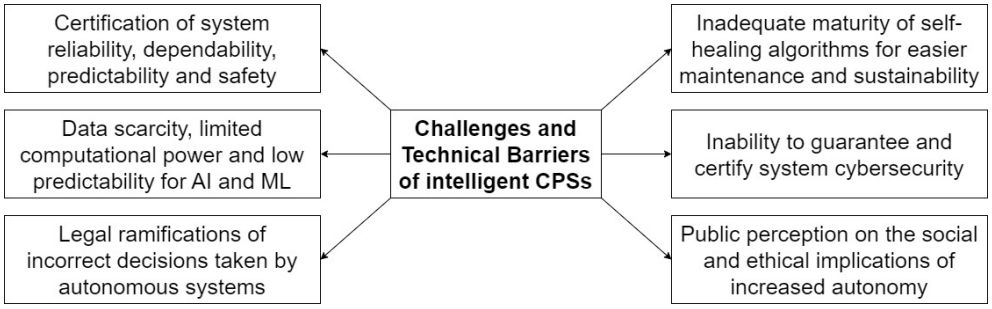

Figure 2. Challenges and technical barriers of ICPSs.

Reliability, dependability and predictability are fundamental requirements of any safety-critical system. In autonomous vehicles, for instance, unpredictable behaviour may lead to fatalities and significant damage to surrounding infrastructure. As technologies evolve, however, sensors, interfaces and CPS capabilities are becoming more diverse [26]. The lack of a unified set of standards makes it difficult to ascertain the interoperability and reliability of a system without undergoing numerous and potentially expensive in-field tests. Moreover, accurately modelling complex CPSs remains a considerable challenge for their widespread adoption. A unified set of cyber-physical standards must therefore be developed for different use-cases and applications.

AI algorithms introduce additional uncertainty, and may take erroneous decisions in scenarios for which they were not properly trained. This issue is particularly problematic when autonomous vehicles rely on external infrastructure for navigation, since road markings and landmarks often vary between different locations and tend to deteriorate over time. AI must therefore become more explainable and deterministic to support its use in safety-critical applications. Airbus, in fact, identifies the certification of AI as one of the biggest hurdles to introduce autonomous aerial vehicles into the airspace [27]. Additionally, regulations on data privacy may limit the data available to train intelligent algorithms. Consequently, system performance may deteriorate since AI models are trained over an insufficiently generic data pool [28]. Furthermore, the large computational power needed to train AI models poses a significant barrier to their adoption in low-cost systems [29].

The legal ramifications of incorrect decisions taken by ICPSs must also be properly defined. Legislation should clearly outline who has legal responsibility and insurance liability in the case of damage caused by autonomous robots or vehicles [30]. In fact, appropriate governmental policies are deemed necessary by Tran et al. [31] to develop and implement a successful ITS infrastructure with autonomous vehicles. Ethical dilemmas may also arise if, for example, an autonomous vehicle must choose between protecting the driver or safeguarding a passer-by in emergency situations.

Furthermore, technologies supporting the sustainability and maintainability of CPSs are insufficiently developed for many real-world applications. Several self-adapting and self-healing algorithms have been proposed in the literature, with most taking inspiration from biological and physical processes [32–34]. Such techniques, however, have yet to be adopted in many practical applications, and considerable testing is still needed to ensure their reliability in safety-critical scenarios. Additionally, Shakeri et al. [35] note that many existing data management and network technologies are inadequate to handle the increasing amounts of expected data traffic, highlighting the need for significant improvements in the scalability of CPSs.

Cybersecurity poses another significant challenge in CPSs, with Alguliyev et al. [36] suggesting that predicting, modelling and preventing cyberattacks are the most complex problems in any CPS. Al-Mhiqani et al. [37] comprehensively review historical cyberattacks on government infrastructure, transport systems and industrial companies and propose a risk analysis technique to evaluate the threat associated with each attack. While research on cybersecurity is ongoing, considerable work is still needed to cater for emerging hacking techniques and develop appropriate security standards. Cyberattacks on autonomous vehicles could otherwise cause catastrophic outcomes, serving as an entry point for cyber warfare or remote terrorist attacks.

The public perception of AI also presents a fundamental obstacle to the widespread adoption of ICPSs. Kurniawan et al. [38] argue that research on the social implications of autonomous vehicles is insufficiently developed. Understanding and shaping the social acceptance of ITSs can therefore be achieved by identifying the social driving forces behind autonomous vehicles. Redundancies due to automated labour, for instance, raise several social and ethical concerns. Community engagement is therefore critical to foster public acceptance and ensure that increased autonomy is mutually beneficial for both industry and society. Moreover, Pelau et al. [39] suggest that incorporating anthropomorphic features and emulating empathy in AI-based systems can enhance consumer acceptance. Denis et al. [40] also advocate for a socially equitable approach to ICPSs in urban mobility, and highlight the significant development required by developing countries to meet the infrastructural demands of ITSs.

A user-friendly development platform for ICPSs can help bridge the gap between theory and practice by providing a versatile tool to address the outstanding challenges of ICPSs. Technological, cybersecurity, and AI-related barriers can be directly addressed through exploratory research conducted within such a system. Furthermore, the platform can be utilised for educational purposes to facilitate discussions on regulations, social implications, and ethical concerns surrounding ICPSs, ITSs, and autonomy.

1.4. Field Programmable Gate Arrays

Overcoming the technical barriers of CPSs requires robust hardware platforms to support increasingly more complex intelligent algorithms. FPGAs are a promising candidate, and are defined by Xilinx as:

> semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects, [that] can be reprogrammed to desired application or functionality requirements after manufacturing [41].

Boutros et al. [42] highlight the dynamic reconfiguration capabilities of this technology, thereby supporting self-healing and adaptive hardware algorithms. Moreover, the high degree of parallelism offered by FPGAs allows them to better replicate parallel physical processes and support quicker execution of AI algorithms.

Magyari et al. [43] analyse state-of-the-art FPGA applications in IoT networks, highlighting their use in ICPSs. They suggest that hardware designs can achieve different trade-offs between minimising latency, power consumption and form factor, and maximising throughput, security and data integrity. FPGAs, in fact, facilitate prototyping, and support a shorter time to market and a simpler design cycle when compared to traditional integrated circuit (IC) implementations. Consequently, FPGAs are particularly suitable for demonstrating, developing and testing new ICPS solutions.

1.5. Literature Gaps and Contributions

Existing literature on ICPSs reveals several gaps that must be addressed. Fundamentally, mechanical and cybernetic design philosophies must be better integrated in a single design approach, backed by reliable regulatory, legal and standardisation frameworks that support the certification of AI for increased autonomy. Additionally, ML algorithms must become more reliable to support their use in safety-critical applications such as self-driving cars and autonomous aerial vehicles. Similarly, advances in swarm intelligence [44], organic computing [45] and multiagent collaboration [46] will facilitate the integration of multiple cyber-physical agents into a cohesive ICPS with adaptive and self-sustaining characteristics.

Optimising both hardware and software components is crucial to improve the efficiency of embedded systems and support the transition towards a net-zero transportation industry [47]. Moreover, cloud computing must improve to cater for the high data management demands brought about by data-driven algorithms [48]. Cybersecurity protocols must also evolve to combat the cyber vulnerabilities of safety-critical ICPSs [49–51].

Research to overcome these technical barriers is ongoing, but addressing state-of-the-art concerns of ICPSs falls beyond the scope of this work. A gap, however, exists for a simple yet comprehensive ICPS platform that can be used for both educational purposes and exploratory cutting-edge research. Mohamed et al. [52] review engineering tools and languages for CPSs. Most existing platforms, however, tend to (1) require a large initial learning curve, making them unsuitable for educational purposes; (2) offer insufficient performance capabilities to support their use for prototyping state-of-the-art algorithms and ICPSs; or (3) solely focus on a single aspect of ICPSs, restricting their ability to demonstrate and explore all the elements that make up a complete ICPS [53].

This work introduces a simple yet robust platform, suitable for both beginners and experienced users, to explore the various elements of ICPSs. The platform supports the integration of multiple sensors, human–machine interfaces (HMIs) and data/communication links in an FPGA-based embedded system. Multiple embedded agents can also interface with a supervisory computer for high-level networking, data capture, processing and display. The entire system thereby allows for both standard and intelligent algorithms to be implemented at both embedded and networked levels, and remains compatible with a variety of sensor interfaces, hardware platforms, cybersecurity algorithms and communication protocols. This platform is suitable for research and educational endeavours, promoting discussions on ICPSs and ITSs while bridging the gap between ICPS theory and practice. The main contributions of this work are as follows:

- Design techniques and considerations for intelligent embedded systems in ICPSs are comprehensively discussed and summarised;
- A robust FPGA-based embedded system is proposed as an exploratory development platform for ICPSs, supporting a range of hardware and software implementations for intelligent vehicular applications;
- The basic functionality of the proposed platform is validated by designing, implementing and testing a simple FPGA-based embedded system with a soft microprocessor core. This interfaces with an IMU and rotary encoder, as typical elements of a vehicular CPS, using embedded software to process the collected data and transmit the system location/orientation and encoder position to a supervisory computer whenever the encoder shaft is rotated. These data are captured and plotted in real-time by a Matlab application running on the host computer;
- The developed prototype is reviewed to discuss the suitability of a simple FPGA-based platform for educational and research endeavours concerning ICPSs and ITSs.

2. System Design and Implementation Methodologies

This section discusses design considerations that are crucial when developing an intelligent embedded system for ICPSs, and summarises the design objectives and requirements of this work. It subsequently presents the designed system architecture, including all hardware and software elements. This is complemented by an overview of all data processing algorithms and ancillary HMI features included within the developed prototype.

2.1. Design Considerations for Intelligent Vehicular Embedded Systems

Lee [54] suggests that existing design abstractions for hardware and software systems are insufficiently compatible to support the development of CPSs, and must be redefined to better complement the realities of physical processes. Nonetheless, Khaitan et al. [55] provide a comprehensive analysis of design techniques and applications of CPSs, highlighting common approaches found in the literature. They propose that effective hardware and software co-design is a well-established approach that can streamline and parallelise system development. Similarly, Segura et al. [56] use a distributed co-simulation protocol to propose a generic interface for CPS simulation and development.

A co-design methodology is particularly important for an embedded system, where hardware and firmware must be seamlessly integrated in a single product. An example of such a co-design flow is illustrated in Figure 3 [57] and promotes a simple yet effective methodology for the design of intelligent embedded systems. The system description is first used to identify the primary objectives of the embedded platform. This is partitioned into unique software and hardware requirements, such that both elements can be developed in parallel. Sufficient inter-communication is further required to guarantee hardware–software compatibility. At each development milestone, remaining tasks are re-partitioned, and the process is repeated until the final embedded system is complete.

When adopting a co-simulation design approach, several design trade-offs need to be considered based on the specific system under development. Some key considerations for intelligent embedded systems are depicted in Figure 4 and hereby discussed.

Primarily, an embedded system should maximise its performance in terms of data throughput, accuracy and speed, particularly in real-time systems [58]. This often imposes a set of memory and processing requirements on the hardware platform, especially when complex AI algorithms must be implemented. Notably, autonomous vehicles require high CNS performance capabilities to remain connected to the ITS, while safely navigating their environment and avoiding collisions. Bijjahalli et al. [59], in fact, suggest that accurate navigation is critical for self-driving cars, and propose a low-cost solution to meet this requirement. Additionally, Namuduri et al. [60] systematically review the current research on CNS systems for AAM, and highlight the challenges that must be addressed for wide-scale AAM operations. These include secure V2V communications, enhanced navigation, and advanced detect and avoid (DAA) capabilities in hostile environments.

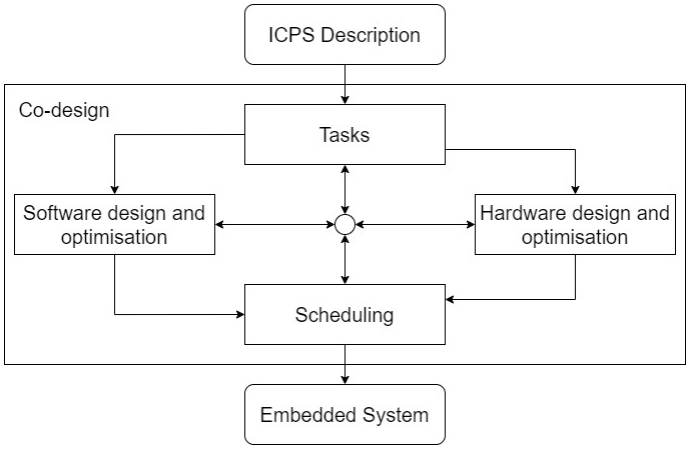

Figure 3. Overview of hardware–software co-design for embedded systems.

In addition to performance, embedded systems must also strive for low power consumption, particularly in battery-operated vehicles and devices [61]. Electric vehicles, for example, require efficient power usage to minimise the frequency of charging. Moreover, Swaminathan et al. [62] affirm that the battery life directly impacts the range, utility and business case of AAM, underscoring the significance of power efficiency in vehicular systems. A trade-off between performance and power efficiency must therefore be achieved. Furthermore, embedded solutions must respect the limitations of available hardware resources, which may further constrain circuit design and optimisation requirements [63]. Timilsina et al. [64], for instance, demonstrate that batteries in electric vehicles have a shorter lifespan than other on-board components, imposing the most stringent design requirements on the vehicular architecture.

Safety and reliability are also critical design factors, especially when devices are embedded in an ICPS. Compliance with industrial standards for electromagnetic interference (EMI), electromagnetic compatibility (EMC), and electrostatic discharge (ESD) protection is particularly important to ensure reliable operation in noisy environments, including industrial machinery and autonomous vehicles. Zhang et al. [65], in fact, highlight the potential damage to an autonomous vehicle's transceiver caused by a high-power microwave pulse, while Arandhakar et al. [66] emphasize the impact of EMI and temperature effects on electric vehicle performance. Printed circuit boards (PCBs) must also be well-designed to withstand harsh environments in which they may be deployed. PCBs within moving vehicles on rough terrain, for instance, require greater resilience to vibrations, while those embedded close to high-power components require a higher thermal tolerance. Fault tolerant designs and hardware redundancies may also enhance system reliability, particularly in high-risk environments with limited accessibility, such as satellites in outer space [67].

Hardware reliability must be complemented by good cybersecurity features, such that the embedded system is well-protected from invasive attacks that may compromise system integrity. Sharma et al. [68] review state-of-the-art cybersecurity developments for connected autonomous vehicles, stressing the importance of AI-based solutions to combat cyberattacks. Moreover, Kukkala et al. [69] discuss the cybersecurity challenges surrounding autonomous vehicles, and propose a roadmap to address outstanding concerns. Open challenges include ensuring data privacy, developing tamper-proof AI solutions and evolving cybersecurity protocols to adapt to emerging cyberattack methodologies. Establishing public acceptance and trust in autonomous vehicles and ITSs is also crucial. Consequently, Khan et al. [70] suggest the use of cybersecurity digital labels to inform consumers and passengers on the cybersecurity health status of intelligent vehicles.

Furthermore, a well-designed embedded system should support future upgrades and modifications, thereby promoting flexibility and adaptability. Proper documentation is therefore necessary to ensure that future developers can readily understand the existing system design and implementation. Moreover, embedded systems must offer a simple

and effective interface to interact with other agents in the ICPS. In particular, networked channels and communication buses must be efficiently managed to prevent a single embedded system from saturating a common bus and stalling the entire system. To address this concern, Choi et al. [71] propose technical solutions to meet the data transfer and latency requirements of ICPSs using a communication bus topology.

Apart from technical considerations, an embedded system must conform to a specific form factor according to its allocated enclosure. The practicality of the system should also be taken into account by minimising costs, predicting a reasonable time-to-market, and considering potential supply chain shortages [72]. Additionally, the design should incorporate a suitable HMI that augments human capabilities within the ICPS. Tan et al. [73] systematically review the challenges and opportunities surrounding HMIs for intelligent vehicular applications. They highlight the complexity and heterogeneity of human intentions as a significant challenge for predictive HMI systems, and suggest that recognition performance must be improved to enable more advanced HMIs. They further observe a lack of datasets to adequately train HMI models for autonomous vehicle applications.

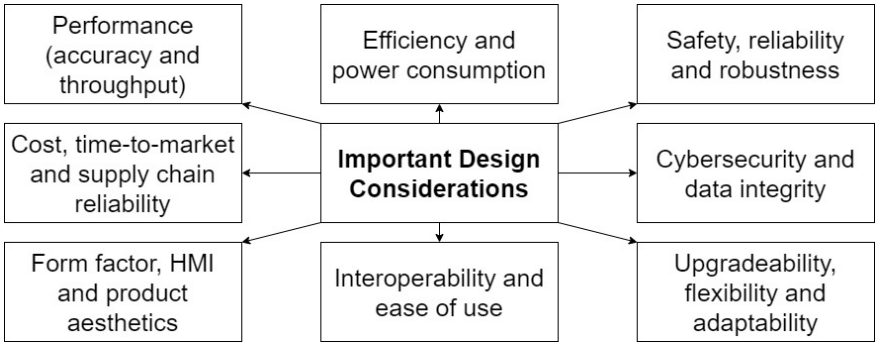

Figure 4. Design considerations for intelligent embedded systems.

2.2. Design Requirements and Objectives

This work focuses on the development of a simple FPGA-based embedded system that serves as a versatile tool for educational and research purposes in the field of ICPSs. The objective is to create a platform that remains compatible with various CNS technologies while accurately representing the unique requirements of intelligent vehicular systems. To demonstrate its capabilities, a simple sensory use-case is considered, showcasing how the FPGA-based embedded system can integrate different aspects of an ICPS and bridge the gap between theory and practice.

The selected approach involves using an FPGA development board, which provides a simple solution for interfacing the embedded system with various commercial off-the-shelf (COTS) components. The FPGA is connected to an IMU and rotary encoder, representing common sensors and interfaces found in intelligent vehicles. Whenever the encoder shaft is rotated, the device transmits its location/orientation and the shaft position to a supervisory computer over a serial communication interface. A high-level application subsequently captures and plots incoming data in real-time. The prototype thereby encapsulates typical components of a vehicular architecture. It serves as a preliminary demonstration on how the proposed platform can be extended to realise a complete ICPS for educational or research purposes, by integrating a wide variety of different sensors, actuators, processor implementations, communication links and HMIs. Consequently, this prototype does not aim to meet specific performance, safety, or security requirements. Instead, its purpose is to present a simple, robust, flexible, and upgradeable approach for deploying and testing ICPS solutions and intelligent vehicular algorithms.

Flexibility and upgradeability are identified as critical design features for an experimentation platform which aims to support future ICPS designs and algorithms. To achieve these characteristics, a COTS FPGA development board is used, allowing for hardware reconfiguration and interfacing with different COTS sensors and modules. Additionally, a

soft microcontroller core with limited memory is employed to run embedded firmware. This offers a familiar coding platform for users that are unfamiliar with hardware description languages (HDLs), while simulating the memory and processing constraints of real embedded systems. Nonetheless, HDL development is still supported, ensuring that the platform remains relevant for low-level implementations of embedded ICPS solutions.

To ensure a user-friendly CPS, an on-board HMI is implemented with common features like a reset switch and light emitting diode (LED) indicator. The system also establishes a simple data link between the embedded agent and a supervisory computer, which can be replaced with more complex wired or wireless communication links for testing communication protocols and cybersecurity algorithms. The supervisory computer can further act as an intelligent coordinator or server, facilitating communication with multiple embedded agents to realise a complex multi-agent ICPS. Mizutani et al. [74] even propose a network solution for inter-FPGA communication in collaborative networking applications. This confirms that an FPGA-based embedded solution can be readily extended to collaborative multi-agent vehicular applications.

In summary, the prototyped architecture involves a single FPGA-based embedded agent interfacing with on-board HMIs, displays, and sensors. In the considered use-case, the FPGA controls an LED display and communicates with an IMU and rotary encoder. Additionally, a soft microcontroller core is implemented within the FPGA to enable the deployment of custom embedded software within the ICPS agent. Each agent is connected to a supervisory computer via a wired data link, which can be upgraded to support wireless communication systems. The supervisory computer captures, processes, and displays incoming data, and can also transmit commands to the embedded agent for controlling data transmission. Furthermore, the computer can be connected to multiple agents to create a networked set of embedded ICPS agents, resembling connected vehicles or vehicle components within an ITS. This is illustrated in Figure 5.

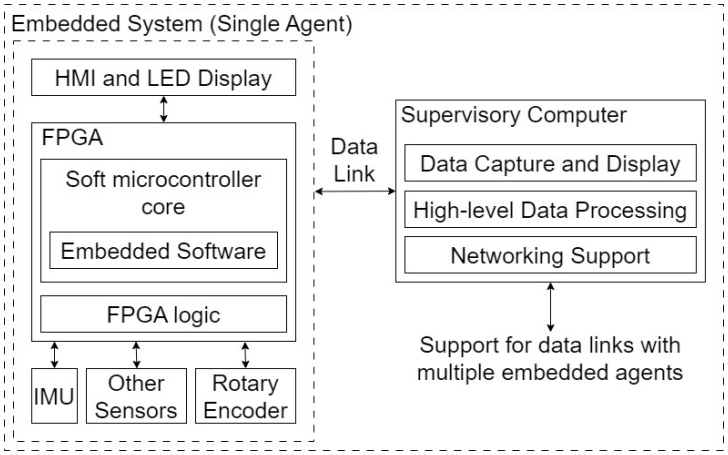

Figure 5. High-level overview of the required system architecture.

2.3. Hardware Design

The embedded system utilises the COTS ARTY A7 development board, housing an XC7A35TICSG324-1L FPGA with 88,280 logic cells, four 6.6 GB/s transceivers and 250 input/output (I/O) slices. This offers a robust yet low-cost development platform that can readily interface with several sensors, particularly through the peripheral module interface (PMOD) standard. Digilent offers well-documented PMOD packages that enable the quick connection of a wide range of sensors, actuators, memory systems, transceivers, and displays to the ARTY A7 board. These include sensors such as time-of-flight, infrared, colour, IMU, humidity, pressure, temperature, and ultrasonic sensors commonly found in intelligent vehicles. The PMOD packages also offer secure digital (SD) or micro SD card slots, flash memory extensions, wireless communication modules or transceivers, as well as a variety of wired communication interfaces, facilitating the deployment and testing of

embedded algorithms for ICPSs with different memory and communication constraints. Additionally, there are various button, keypad, joystick, LED, and display boards available to enable simple HMIs within the embedded system. The wide range of COTS modules ensures that the platform can be easily customised for different vehicular requirements without the need to develop application-specific hardware from scratch. The proposed solution thereby remains an effective tool for facilitating ICPS education and expediting exploratory ICPS research. Moreover, the ARTY A7 development board offers extensive community support and resources, including online forums, tutorials, and example projects. This ensures that users have access to a wealth of knowledge and can easily seek assistance when developing their ICPS solutions. Nonetheless, custom extension boards can also be developed and interface with the ARTY A7 board if specific components must be included for cutting-edge research on a specific vehicular system.

To implement a softcore Microblaze processor on the FPGA, Xilinx intellectual property (IP) cores are used. The Microblaze processor has 128 kbits of block random-access memory (BRAM) and incorporates the necessary interrupt, clock, and peripheral controllers in FPGA logic. This approach allows the microcontroller to be dynamically synthesized and mapped onto the FPGA's configurable logic fabric, rather than being introduced as a physical hardware component. The softcore microcontroller thereby ensures a flexible and re-configurable design that is suitable for different vehicular applications. All the implemented cores are inter-connected through the advanced extensible interface (AXI) protocol that forms part of the ARM advanced microcontroller bus architecture (AMBA), commonly used in system on chip (SoC) designs. An AXI bus is also employed to connect the microcontroller to Xilinx IP cores designed to interface with the rotary encoder and NAV PMOD devices, connected to the JA and JB headers of the development board, respectively. Moreover, the lite Xilinx IP core for UART communication is included to establish a serial communication link with the host computer. A set of on-board red–green–blue (RGB) LEDs are also connected to the microcontroller peripheral controller, to act as a simple on-board HMI. The use of Xilinx IP cores provides a well-established and reliable foundation for implementing various functionalities within the FPGA. These IP cores are rigorously tested and validated, ensuring compatibility and reducing the risk of errors or design flaws. Consequently, only customised settings, logic blocks and embedded firmware that differ from default block implementations are detailed throughout this work.

A summary of the hardware design is illustrated in Figure 6. The default clocking wizard IP core is used to clock the soft-core Microblaze processor and ancillary peripheral and interrupt controllers. Additionally, an on-board reset switch is used to enable a hard reset of the clocking and processor logic. A memory module is also connected to the soft-core microcontroller, with the specified 128 kbits of BRAM, and the default debug module is implemented to facilitate embedded firmware development. The default AXI peripheral controller is subsequently used as an interface between the processor core and all peripheral controller IP cores. Specifically, the default IP cores developed for the PMOD encoder, PMOD NAV IMU, on-board GPIOs and UART communication are employed and connected to external peripherals through the ARTY A7 connection headers. Additionally, the on-board UART-Universal Serial Bus (USB) bridge is used to facilitate communication with the host computer through a simple micro-USB to USB connection.

The flexibility of the FPGA-based embedded system allows for iterative development and rapid prototyping. Users can easily modify and reconfigure the hardware and firmware to accommodate the evolving requirements and experiment with different design approaches, accelerating the development cycle. This platform can also be modified to include different processor implementations. Khairullah [75], for example, showcases the possibility of realising a reduced instruction set computing (RISC) processor within an FPGA. The prototype developed in this work, however, solely aims to showcase the potential of a COTS FPGA board to deploy complete ICPS systems. More complex developments must also consider additional FPGA design concepts, such as pipelining, occupancy and datapath restrictions, as reviewed by Monmasson et al. [76].

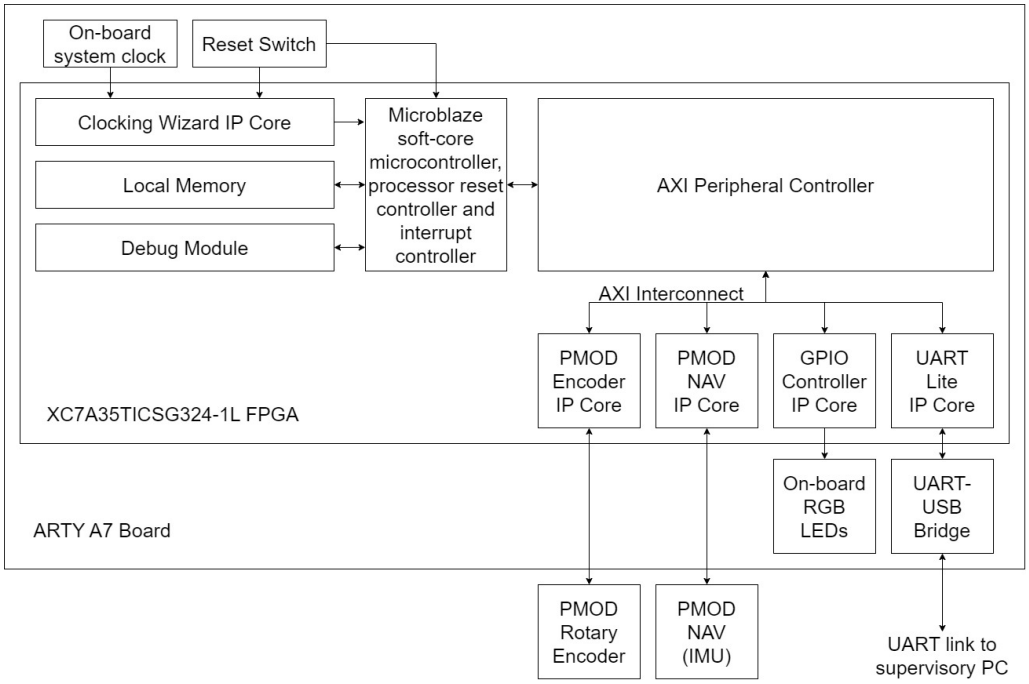

Figure 6. Overview of the implemented hardware design.

2.4. Sensor Interfaces and Communication Protocols

Selecting appropriate communication protocols is crucial in any hardware design. Most ICs support serial peripheral interface (SPI) and/or inter-integrated circuit (I2C) protocols for intra-board communication, while UART or Ethernet protocols are commonly used for communication with external devices. Considerable literature [77] exists describing the operation of such protocols. In general, however, SPI tends to support faster data transfer than I2C, but requires more connections between peripherals. Additionally, Ethernet offers higher data rates and better noise immunity than UART, at the expense of non-deterministic behaviour and higher protocol complexity. A summary of this comparison is presented in Table 1. SPI is thereby used to communicate with the PMOD NAV, ensuring swift IMU data transfer. Conversely, the rotary encoder is monitored through a set of general purpose I/O (GPIO) pins, and UART is used to communicate with the host computer. Consequently, the system can be readily monitored using open source UART virtual terminals. Nonetheless, future platform upgrades may incorporate controllers for additional communication protocols to better support diverse system requirements.

Table 1. Comparison of typical communication protocols.

Feature	UART	SPI	I2C	Ethernet
Data rate	<460 kbps	<20 Mbps	<1 Mbps	>1 Gbps
Type of communication	Asynchronous	Synchronous	Synchronous	Synchronous
Hardware complexity	Low	Moderate	Moderate	High
Benefits	Simple and widely supported	Full duplex communication and low power consumption	Supports multiple masters and only needs a two-line bus	Good quality and high data rates over long distances
Disadvantages	Restricted to two devices at a pre-defined data rate	Requires a large number of connections for multiple slaves	Half duplex communication with low overhead	Performs poorly in high data traffic and is non-deterministic

2.5. Embedded Software Design and Data Processing

All sensor interfacing, data collection, low-level processing and data transmission algorithms are implemented on the softcore microcontroller using the C programming language. These are optimised to respect the memory limitations of the Microblaze processor, ensuring a more realistic prototype scenario. After traversing a custom start-up sequence, the embedded system has four modes of operation, configured by the switch on the PMOD encoder board. A high-level overview of the embedded firmware is shown in Figure 7.

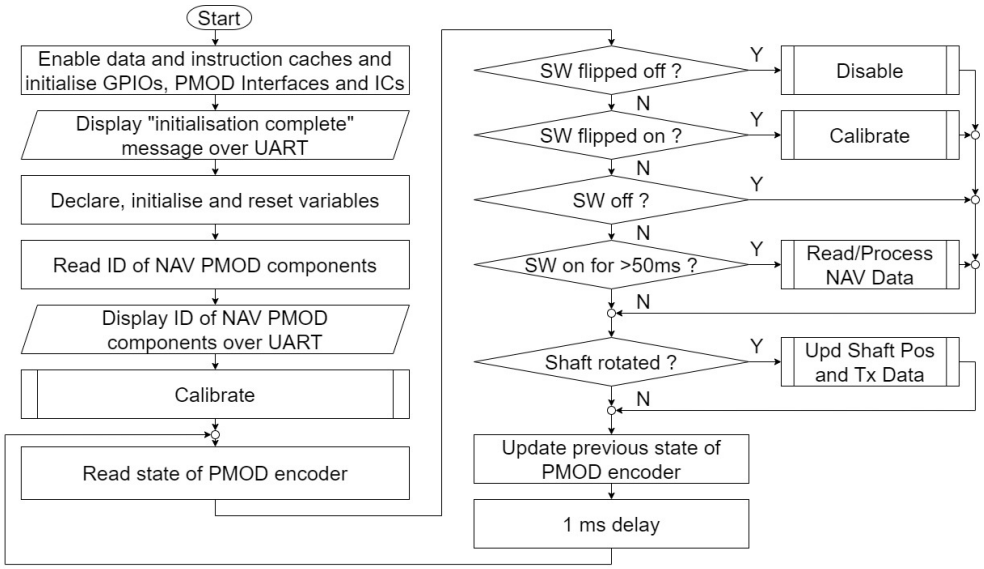

Figure 7. Flow-chart detailing a high-level overview of the implemented embedded firmware.

The system is designed to appropriately initialise all interfaces, peripherals and caches on start-up, according to the datasheet of each respective device. Specifically, the default libraries for SPI communication with the LSM9DS1 IC on the PMOD NAV were modified to initialise the device with a sampling frequency of 50 Hz, supporting a quicker data update rate that is more conducive to the requirements of autonomous vehicles. Moreover, the internal accelerometer filters were enabled by appropriately configuring the device registers, reducing the need for subsequent digital filtering in software. Upon completion of this startup sequence, an appropriate message is transmitted to the supervisory computer over UART. The code subsequently initialises and resets all run-time variables, configures the PMOD NAV and reads the IDs of all on-board sensors. These are transmitted over UART for validation by the supervisory computer.

Calibration offsets for each sensor are determined, by averaging the readings obtained over 100 successive sampling instances, while instructing the user to not touch the device. Throughout this process, appropriate messages are transmitted over UART, and the on-board LEDs are set to blue to indicate that the user should avoid moving the board. At 1 ms intervals, the processor polls the state of the encoder PMOD modules, and disables the platform when the switch is flipped off. This involves resetting all run-time variables (effectively resetting the reference frame of the system and the zero position of the encoder shaft) and setting the on-board LEDs to red. Whenever the switch is flipped back on, the system re-collects calibration offsets to account for possible temperature variations or device movements, corresponding to a different effect of gravity along each accelerometer axis. This calibration procedure ensures that the prototype retains reliable and accurate results throughout all stages of its operation. Trinh et al. [78] discuss more sophisticated IMU calibration procedures using Kalman filter techniques. Such algorithms, however, were deemed beyond the scope of this prototype.

Upon shaft rotation, the number and direction of discrete ticks or rotary motions are recorded and converted to the shaft position in degrees using:

$$P = \frac{\#\ ticks}{max\ \#\ ticks\ in\ 1\ revolution} \times 360^\circ = \frac{\#\ ticks}{20} \times 360^\circ, \quad (1)$$

where P is the angular position of the shaft. The result is transmitted over UART to the host computer, together with the current 3D location and orientation of the board. The encoder state is polled every 1 ms to ensure that all tick rotations are detected by the device. A summary of these procedures is illustrated in Figure 8.

If the switch is on, the processor sets the on-board LEDs to green and reads IMU data at 50 ms intervals. This avoids stalling the processor while being frequent enough to reliably capture user movements. Incoming accelerometer and gyroscope data are subsequently processed, as shown in Figure 9. On-board processing eliminates the need for continuous data transmission to a host computer, thereby improving the communication bus efficiency. While computationally intensive sensor fusion algorithms could not be implemented on the limited memory available, several data processing techniques were employed to reliably and accurately obtain the device position and orientation from the raw sensor data.

Data measurements are calibrated by removing zero offsets and accounting for any scale factor deviations. Since integration is used to derive position and orientation data from the accelerometer and gyroscope readings, small numerical offset errors will cause output saturation and system instability. A mechanical filtering window is thereby implemented, whereby readings below a particular threshold are set to zero to prevent integrator saturation. A numerical integrator is subsequently used to determine the device orientation along each axis, with the result manipulated to lie between -360° and 360° and prevent register saturation. Similarly, accelerometer readings are converted to metres per second squared, and a double integration process is used to determine the 3D device position.

For ease of testing, position and orientation are computed in the body frame of the inertial sensor. The algorithm, however, can be readily used to compute the absolute device position and orientation in an external reference frame, as described in Section 3.5 below. Moreover, numerical integration is implemented according to the trapezoidal rule for computational efficiency. Nonetheless, more accurate yet memory-intensive numerical integration techniques can be used when higher accuracy is needed.

Finally, NXP [79] suggests that small residual velocities after integrating acceleration data will cause the output of the second position integrator to quickly saturate and yield large output errors. Consequently, an end-of-movement check is performed, such that the velocity output of the first integrator is reset to zero if several successive accelerometer readings indicate that the device is stationary. This is a reasonable assumption, unless the vehicle is expected to move at a perfectly constant velocity for long periods of time.

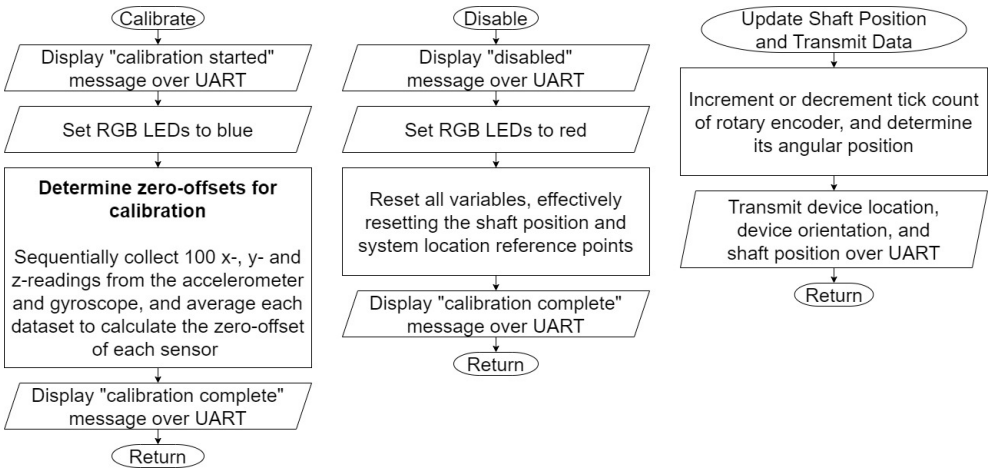

Figure 8. Flow-charts summarising device operation during calibration (**left**), disable operations (**centre**), and upon rotation of the encoder shaft (**right**).

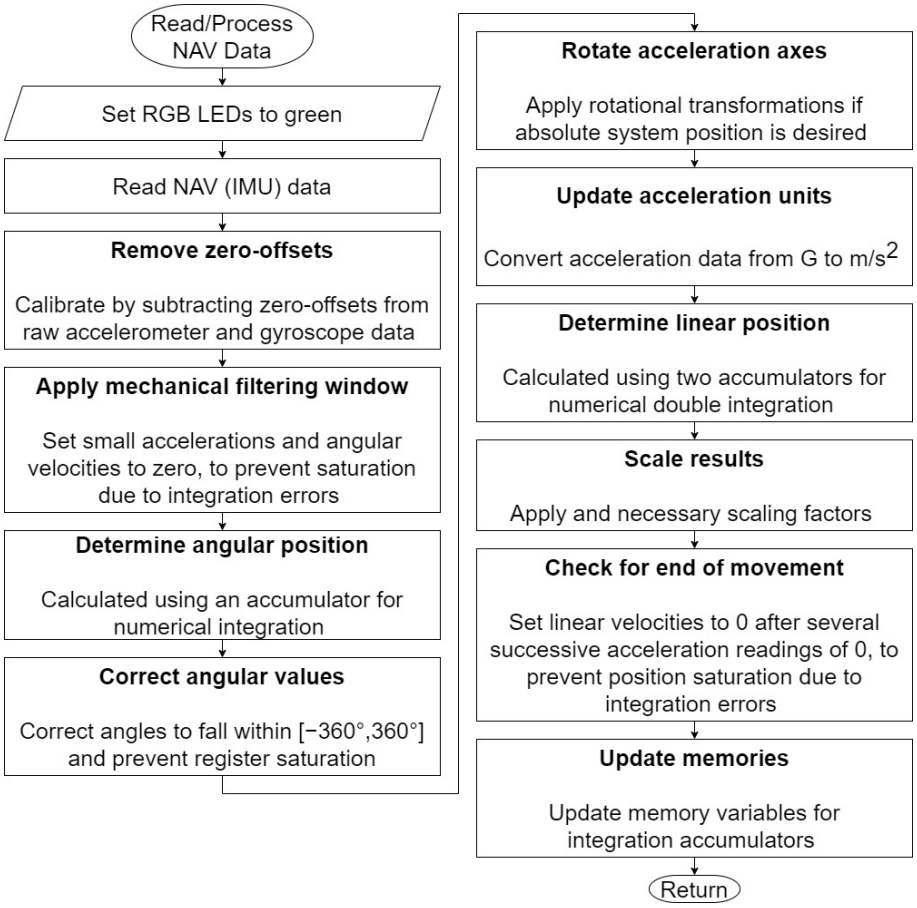

Figure 9. Flow-chart summarising the algorithm used to process incoming sensor data.

2.6. Data Capture and Human Machine Interface

To complement the on-board LED indicators, a user-friendly HMI is implemented on the host computer. The serial data transmitted by the embedded system may be viewed through a virtual serial port (configured for a baud rate of 9600 bps), or captured and displayed on a custom Matlab application. The latter plots the board location and orientation against the shaft position, using the algorithm summarised in Figure 10. This feature serves as a simple demonstration on how the developed prototype supports high-level data capture and display, and can be readily upgraded to represent frameworks that are more conducive to vehicular applications.

A call-back function is used to read incoming data whenever a termination character is detected on the serial port. Each incoming string is subsequently processed and used to control a real-time display. If the incoming message contains any data, it is extracted and used to update the plot in real-time. If not, incoming messages are simply displayed in the Matlab user terminal. Moreover, if a disable message is detected, suggesting that the on-board reset switch was flipped, the plot is reset in anticipation of a new set of datapoints. Additionally, the system halts execution if the user presses the Enter key, ensuring that the application does not become stuck in an infinite loop.

This application can be extended to support bi-directional control of the embedded system, by transmitting commands over UART to the embedded agent. With appropriate firmware development, the soft-core microcontroller can poll and decipher incoming UART commands. Alternatively, standard FPGA logic blocks may be employed to receive and interpret UART data. Moreover, networked operations across a set of embedded agents can be realised by simultaneously communicating with multiple embedded agents. Various online articles and tutorials exist to implement such upgrades, ensuring that the platform remains accessible to inexperienced ICPS users.

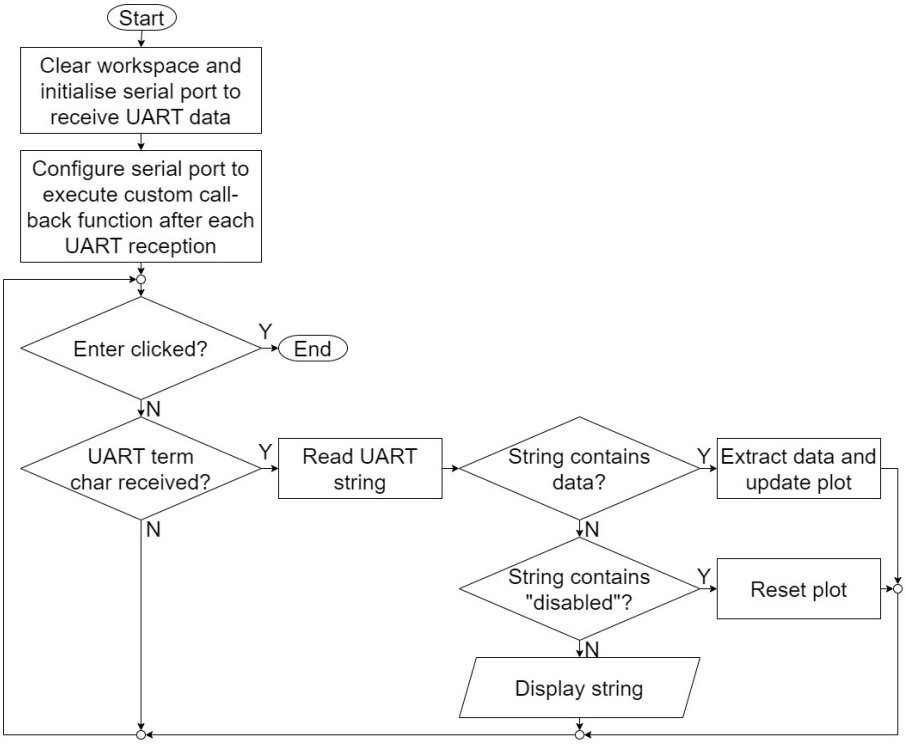

Figure 10. Flow-chart summarising the implemented Matlab application.

2.7. Implemented Prototype

The implemented prototype is shown in Figure 11, supported by several HDL, constraint, C and Matlab files available as an addendum to this work.

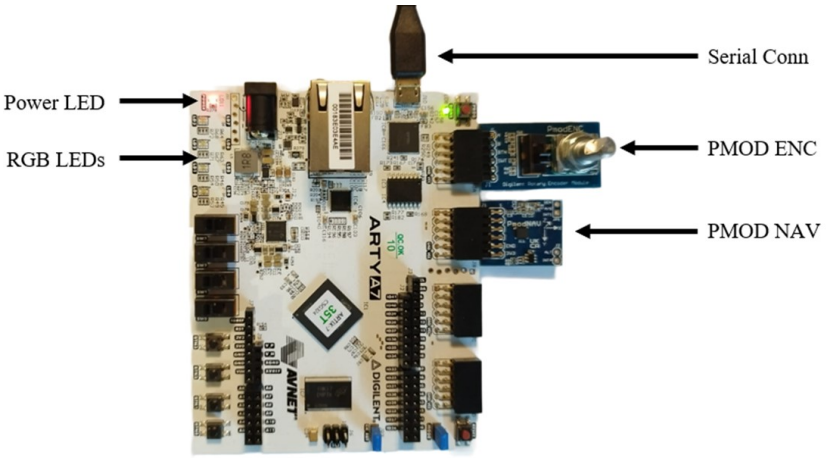

Figure 11. Layout of the implemented hardware prototype with an Arty A7 development board.

3. Testing, Results and Discussion

This section briefly reviews testing procedures for CPSs and FPGA-based embedded systems, before describing the testing approach used throughout this work. It subsequently presents the results obtained when validating and characterising the implemented prototype. Finally, potential design improvements and future modifications are highlighted, including techniques to improve system security, efficiency, accuracy and reliability. Moreover, the flexibility of the developed platform is reviewed to assess its suitability as a development platform for ICPSs and intelligent vehicular algorithms.

3.1. Testing Procedure

Any intelligent embedded system or CPS must be rigorously verified, tested and validated to ensure it meets project requirements. Zhou et al. [80] systematically review testing methodologies and testbeds for CPSs, and conclude that the heterogeneity and complexity of CPSs pose considerable challenges to existing testing procedures. Nonetheless, the V-model [81] has been widely adopted as an industry standard for software development, and can be extended to outline a sequential testing procedure for simple ICPSs. This involves individual testing of system elements against their respective requirements, followed by integration testing of multiple elements to assess the system as a whole. Moreover, the final system must be validated in the expected operation environment, under reasonable extreme conditions (such as radiation, temperature and humidity levels), and considering all edge cases and boundary conditions, when possible. While this may be impossible when considering random physical processes and stochastic intelligent algorithms, every effort must be made to validate system performance in all anticipated operating scenarios, particularly in safety-critical applications. Several intelligent algorithms have also been proposed for automatic optimisation and testing of CPSs [82].

In applications like spacecraft design, physical testing is not always possible, such that simulations are often used for the initial system validation. In particular, FPGA-based embedded designs often use behavioural, post-synthesis and post-implementation simulations to sequentially verify the logical, synthesized and implemented designs through a well-defined test bench, respectively. Additionally, timing, memory and layout constraints must be met to ensure a reliable implementation.

Zheng et al. [83] review the perception of researchers on CPS verification and validation techniques, and conclude that existing formal methodologies are insufficient to support the development of robust ICPSs. While simulations or controlled environments may help mimic the physical process associated with a CPS, these often fall short in accurately representing all the factors affecting the system in its true operating environment. When possible, in fact, in-field tests are preferable to fully validate system operation. The ASTREA project undertaken by the University of Malta, for instance, aims to provide a low-cost pico-satellite platform to directly test the reliability of electronic components and embedded systems in space, rather than relying on a series of controlled laboratory tests. Similarly, intelligent vehicles will benefit from physical tests that replicate the physical ITS environment. Moreover, safety-critical systems such as autonomous vehicles should adhere to relevant safety standards to guarantee system reliability.

Rigorous testing and characterisation was deemed unnecessary for the implemented system, since its main purpose is to showcase the flexibility of a platform for CPS research. Moreover, test-bench simulations for FPGA validation were considered unnecessary since standard Xilinx IP cores were utilised throughout the hardware design. Nonetheless, all elements of the CPS were validated against the project requirements, as depicted in Figure 12. The implemented hardware was first validated by ensuring that the microcontroller could successfully communicate with all sensor and communication modules. This was confirmed by successfully reading the IDs of each sensor and transmitting them over UART to the host computer. All embedded firmware and data processing algorithms were subsequently validated by confirming that the system operated correctly in all four modes of operation. Additionally, the high-level data processing and display algorithms were validated by ensuring correct processing and display of incoming UART messages. Finally, all components were tested within the integrated system, and a preliminary characterisation of the implemented use-case was performed. The primary objective of this research, however, remains to demonstrate that a simple commercially available FPGA platform can be utilised for educational and research purposes in ICPSs.

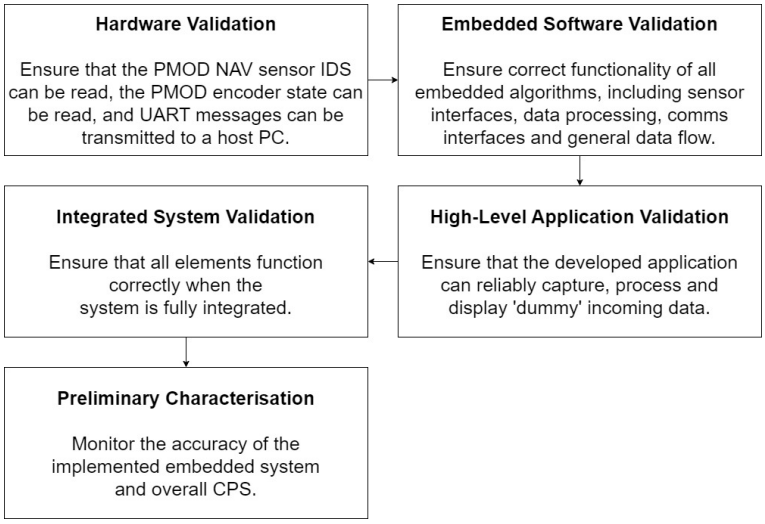

Figure 12. Overview of the adopted testing procedure.

3.2. System Validation

During system testing, both the individual components and unified CPS exhibited correct operation. The implemented prototype successfully integrated an IMU, switch and rotary encoder in an FPGA-based embedded system. Moreover, a softcore microcontroller was successfully implemented and used to collect, process and manipulate IMU data to determine the device position and orientation. This information, together with the position of the encoder shaft, was successfully transmitted over a serial UART interface on shaft rotation, captured by a Matlab application, and displayed through a series of real-time plots. Examples showing the observed plots when rotating the encoder shaft and moving the device along all six degrees of freedom (DOFs) are displayed in Figures 13 and 14.

The implemented HMI was also observed to function correctly, with the on-board LEDs appropriately indicating the mode of system operation, as shown in Figure 15. The embedded system further operated correctly when interfaced with an open-source virtual serial terminal, as demonstrated in Figure 16. Moreover, the custom application successfully displayed all system status messages, reset the displayed plots on a user reset, and halted execution upon clicking the Enter key, as shown in Figure 17.

These results affirm the successful design and implementation of a robust and flexible ICPS development platform. The prototype also demonstrates strong scalability and extensibility, allowing for seamless integration of additional components and modules as the complexity of ICPSs increases. With its modular design and flexible architecture, researchers and developers can easily incorporate new sensors, actuators, or communication interfaces to address specific application requirements. Specifically, all components of a simple CPS were successfully developed using an FPGA development board interfaced with COTS components. These included a sensor, visual LED display, rotary encoder, embedded processor with custom firmware, user-activated switches, and a data and communication link with a supervisory controller. A high-level application was also successfully deployed to collect, process and display data received from the embedded agent.

These claims substantiate the platform's capability to efficiently implement customised ICPSs without the need for specialised hardware or software expertise. Consequently, this platform holds significant value for educational purposes by simplifying the complexity of real-world heterogeneous ICPSs and bridging the gap between ICPS theory and practical implementation. Notably, sensors and communication interfaces can be readily substituted with alternative COTS or customised options to fulfil specific application requirements. Similarly, custom embedded firmware, FPGA logic, and high-level applications can be developed for different research scenarios, ensuring that the platform remains relevant for cutting-edge ICPS research. Furthermore, the CPS can incorporate customised communication and cybersecurity protocols, including intelligent algorithms.

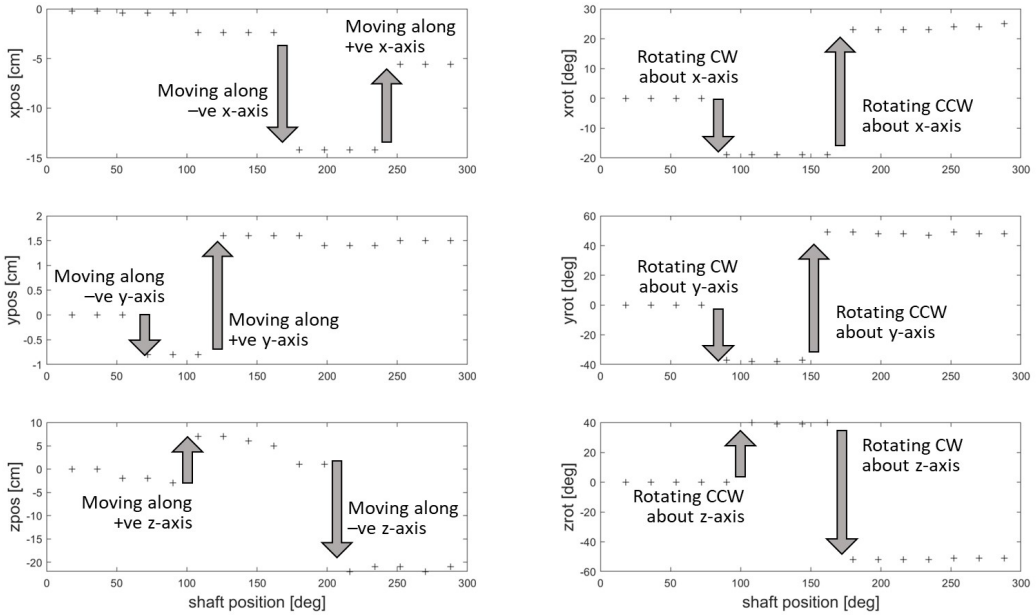

Figure 13. Summary of plots from six independent validation tests, showing correct system operation when moving or rotating the device along all 6 DOFs while rotating the encoder shaft.

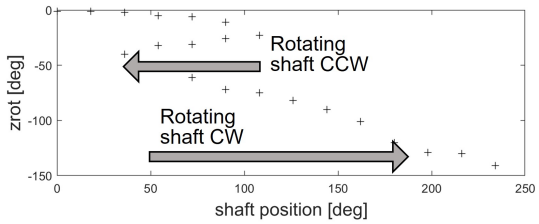

Figure 14. Plot showing correct system operation when rotating the encoder shaft in both directions, while rotating the entire board clockwise about the z-axis.

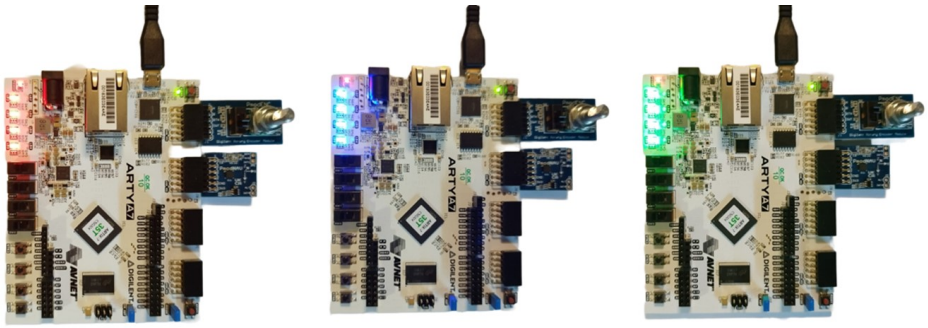

Figure 15. Correct LED display in disabled (**left**), calibration (**middle**) and enabled (**right**) modes.

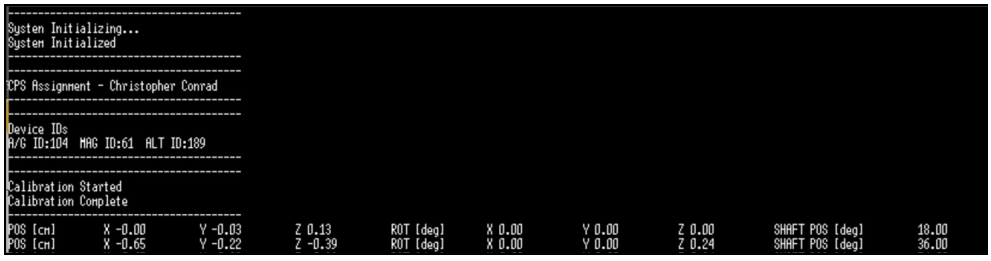

Figure 16. Example of the HMI when using a virtual serial port.

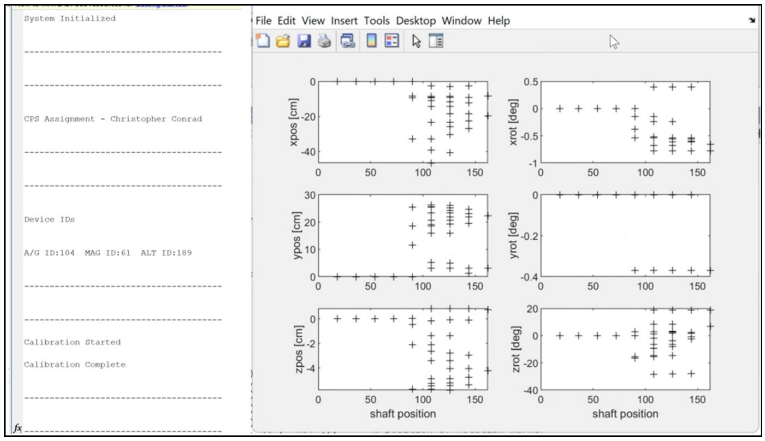

Figure 17. Example of the HMI when using the custom Matlab application.

3.3. Prototype Accuracy

Characterising the IMU-based system falls beyond the scope of this work, which solely aims to demonstrate that an FPGA-based system can be used to develop a complete ICPS. Notably, the prototype cannot be reliably characterised without an appropriate test setup, owing to a large degree of human error. Specifically, it is impossible to guarantee that device motion is perfectly aligned along a particular axis. Nonetheless, crude measurements demonstrated reasonable accuracy along all six DOFs, as summarised in Figure 18. To limit inaccuracies, these readings were taken while slowly orienting the platform. This approach, however, is not conducive of high-speed vehicular applications, and solely aims to demonstrate correct operation of the implemented use-case. Angular readings were observed to be more accurate than position readings, as expected due to the additional integrator needed to extract positional data from the accelerometer readings. These observations further verified the correct system operation.

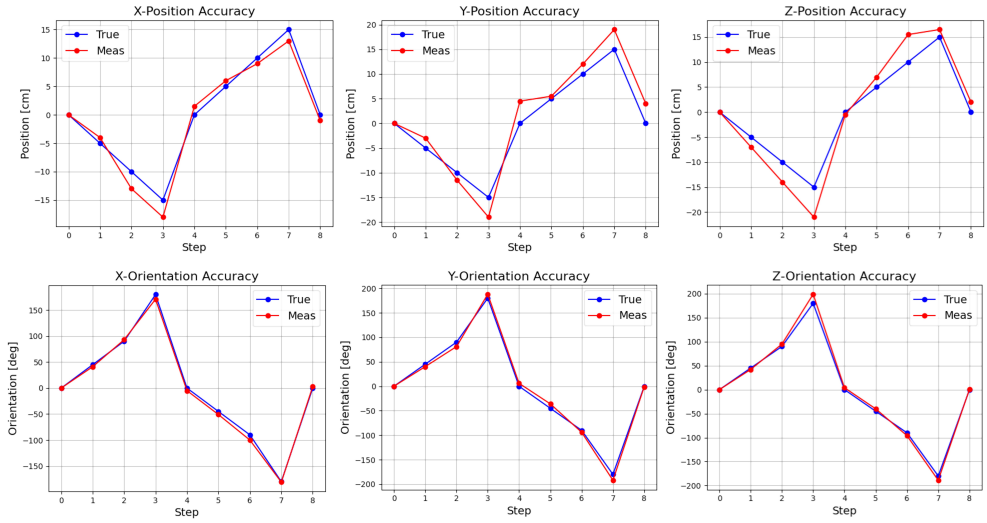

Figure 18. Preliminary showcase of prototype accuracy from a crude proof-of-concept test.

When conducting research on the proposed platform in an academic setting, it is imperative to employ rigorous testing and characterisation methodologies to validate the hypothesis or research question under investigation. This becomes particularly critical in the context of autonomous vehicle applications, where stringent testing protocols are essential to ensure the safety of drivers, vehicle occupants, pedestrians, and other users of the transportation infrastructure. In fact, Naufal et al. [84] assert that without robust safety assessments and comprehensive test data, the realisation of autonomous vehicles and ITSs remains unattainable.

3.4. Accuracy Improvements

Inertial systems implemented in real ICPSs will likely require considerable accuracy improvements. While more expensive and accurate IMUs can be used, several processing techniques may also be employed to improve position and orientation accuracy:

- Thermal compensation algorithms may be included, using the temperature sensor on board the PMOD NAV to monitor the ambient system temperature;
- Software-based timing algorithms can be replaced with interrupt-based hardware timers, thereby ensuring more accurate and consistent integration time steps;
- More complex algorithms may be used to fuse magnetometer, gyroscope and accelerometer data and obtain better position and orientation estimates. Several sensor fusion techniques have been proposed to handle this problem, often based on extended Kalman filtering [85]. Mahony and Madgwick filters [86] have also been shown to achieve good accuracy while incurring less computational overhead;
- IMU data can be fused with data obtained from other sensors [87];
- Intelligent embedded algorithms can be used to improve the estimation accuracy [88];
- If real-time data capture is not required, bi-directional digital filters can be implemented after data collection for better estimation accuracy.

3.5. Absolute 3D Positioning

While relative positioning in the device body frame was suitable for demonstrative purposes, absolute positioning in an external inertial frame is often required. An inertial system mounted on an UAS or self-driving car, for example, should support position and orientation calculations in a pre-defined external reference frame. This can be accomplished by rotating the acceleration axes using standard rotational matrices. Alternatively, quaternions offer a more robust representation of the 3D orientation of an object and are commonly used to translate IMU readings to a defined reference frame. Algorithms using this hyper-complex number system tend to be more computationally efficient, making them particularly suitable for embedded firmware within CPSs. Patel et al. [89], for example, use quaternions to deploy and evaluate IMU fusion algorithms for UAS navigation.

3.6. Efficiency

Embedded algorithms were implemented on a softcore microcontroller to demonstrate the flexibility of an FPGA-based platform. FPGA resources, however, could be used more efficiently to achieve better performance with lower power consumption. Custom VHDL scripts or IP cores, for instance, could be used to manage sensor interfaces and implement the data signal conditioning blocks in hardware. In fact, Seng et al. [90] systematically review embedded intelligence on FPGA systems and conclude that high computational power and low power consumption must be enabled through efficient FPGA designs.

3.7. Cybersecurity Considerations

Since the implemented prototype solely aimed to showcase the robust capabilities of the proposed platform, no cybersecurity features were designed or implemented. Nonetheless, encryption and decryption algorithms can be readily implemented and tested at both an embedded and supervisory level. Simple linear-feedback shift registers (LFSRs), for instance, may be used for pseudo-random number generation within a stream cypher [91]. More advanced block cypher and hash function-based encryption may also be implemented and tested for more advanced cybersecurity protocols. In fact, developing new and resilient cryptographic techniques remains an open research problem [92–94] and this platform offers a simple yet effective way to test such algorithms over a one or more data and communication links. This is especially important for autonomous vehicles and ICPSs within an ITS, where cyberattacks can have catastrophic and fatal results.

3.8. Flexibility for ICPS Experimentation

The primary objective of this research is to investigate the potential of using an FPGA development board with COTS or custom modules to facilitate the exploration, development, and testing of ICPSs and intelligent vehicular algorithms. The deployed use-case demonstrates that this approach offers a viable solution to advance ICPS research, foster ICPS education, and stimulate discourse surrounding the fundamental challenges associated with the deployment of ICPSs in autonomous vehicles and ITSs. As opposed to the platform developed by Zhang et al. [95], this solution explicitly considers the requirements of vehicular applications, including collaborative and networked ICPSs, stricter safety and performance criteria, and the need for better cybersecurity protocols. Additionally, the Raspberry Pi platform proposed by Garcia et al. [96] offers less flexibility than the developed FPGA-based embedded system, which enables dynamic hardware reconfiguration.

Isakovic et al. [97] suggest that custom hardware can reduce the costs of CPS platforms for research and education. Their approach, however, does not offer an open-source solution that is independent of specific hardware and software systems. In contrast, this work promotes the adoption of COTS development boards and modules to enable a broader audience to engage in ICPS experimentation and advance the emerging fields of autonomous vehicles and ITSs. Landolfi et al. [98] also propose a platform to support automotive CPS deployment, but focus on the marketplace architecture rather than emphasizing the educational and research potential of a technical ICPS experimentation platform.

Several simulation tools have also been developed to aid ICPS research and education. Schmittle et al. [99] and Iannino et al. [100], for instance, respectively propose simulation environments for UAS and factory systems. Nonetheless, the hardware-based experimentation platform prototyped in this work offers significant advantages over a purely software-based implementation. Simulation platforms often lack the ability to simulate low-level electronics, making it challenging to conduct accurate testing of intelligent embedded systems. In vehicular applications, ensuring proper power and thermal management at an embedded level is particularly crucial, highlighting the suitability of a hardware-based experimentation platform for ICPSs. Similarly, a simulated system may struggle to precisely model all the elements of the physical system in which an ICPS will be deployed.

Additionally, the proposed platform addresses the need for interdisciplinary collaboration in ICPS research. By providing a platform that integrates hardware and software components, researchers from diverse backgrounds, such as computer science, engineering, and transportation, can collaborate effectively to tackle the complex challenges of ICPSs in autonomous vehicles and ITSs. This interdisciplinary approach fosters a holistic understanding of ICPSs, promotes cross-pollination of ideas, and encourages the development of innovative solutions that consider both technical and societal aspects. Ultimately, it enables a more comprehensive and integrated approach towards building safe and efficient ITSs.

3.9. Limitations

While the FPGA-based platform presented in this research offers numerous advantages, it is important to acknowledge its limitations in addressing ICPS research questions. Primarily, cost-effective FPGA development boards are often limited in terms of logic elements, memory, and processing capabilities. This restricts the complexity and scale of the solutions that can be deployed within a single embedded agent. Although more powerful development boards are available, these will significantly increase the cost of such an experimentation tool. Additionally, the platform is limited by the hardware connections available on the ARTY A7 development board. Adding more connections or deploying custom sensors may require custom hardware design and development.

The platform may also face limitations in accurately simulating all aspects of a real-world environment. Physical factors, such as power management, thermal effects, and environmental conditions, may not be fully replicated in controlled hardware-based experimentation, potentially impacting the fidelity of the ICPS evaluation. Nonetheless, this solution remains a valid and versatile tool for initial exploratory research.

4. Conclusions

This work successfully demonstrates the potential of an FPGA-based embedded platform for exploring, developing, and testing ICPSs for autonomous vehicles and ITSs. The developed platform uses COTS components and a customisable FPGA development board to offer a cost-effective and accessible solution for ICPS research and education. This is showcased through a simple use-case that demonstrates the seamless integration of various sensors, communication modules, and HMIs within an embedded FPGA system. A high-level application further highlights the platform's ability to support real-time data collection, processing and display by a supervisory computer. The proposed solution explicitly addresses the specific requirements of intelligent vehicular applications, to encourage ICPS research and promote discussions on the fundamental issues surrounding their deployment in autonomous vehicles and ITSs. Moreover, the platform supports the deployment of low-level embedded firmware, intelligent algorithms, and custom sensor modules, ensuring that it remains relevant for state-of-the-art exploratory research.

This research offers a valuable resource for researchers, educators, and practitioners in the fields of ICPSs and ITSs. By fostering interdisciplinary collaboration, addressing domain-specific requirements, and supporting real-world experimentation, the platform opens up new avenues for advancing the understanding, development, and deployment of intelligent and efficient transportation systems of the future.

Future work will focus on enhancing and testing the capabilities of the platform, such as incorporating advanced ML algorithms for intelligent decision-making, deploying cybersecurity protocols over wired and wireless communication channels, and investigating techniques for ensuring the resilience and reliability of ICPSs in dynamic and unpredictable environments. Additionally, the applicability of this tool for CPS experimentation in different domains will be explored. Furthermore, empirical studies and user evaluations should be conducted to assess the effectiveness and usability of this tool in educational and research settings. This would provide valuable insights to optimise its research or educational potential. Finally, different use-cases will be deployed on the developed prototype, to better understand the capabilities and limitations of an FPGA-based embedded system for ICPS experimentation and exploration.

Author Contributions: Conceptualization, C.C. and S.A.-R.; methodology, C.C.; software, C.C.; validation, C.C.; formal analysis, C.C.; investigation, C.C.; resources, S.A.-R.; data curation, C.C.; writing—original draft preparation, C.C.; writing—review and editing, S.A.-R.; visualization, C.C.; supervision, S.A.-R. and A.T.; project administration, A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAM	Advanced air mobility
AI	Artificial intelligence
AMBA	Advanced microcontroller bus architecture
AXI	Advanced extensible interface
BRAM	Block random access memory
CLB	Configurable logic blocks
CNS	Communication, navigation and surveillance
COTS	Commercial off-the-shelf

CPS	Cyber-physical system
DAA	Detect and avoid
DOF	Degree of freedom
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
ESD	Electrostatic discharge
FPGA	Field programmable gate array
GNSS	Global navigation satellite systems
GPIO	General purpose input / output
HDL	Hardware description language
HMI	Human machine interface
IC	Integrated circuit
ICPS	Intelligent cyber-physical system
IMU	Inertial measurement unit
I/O	Input/output
IoT	Internet of things
IP	Intellectual property
ITS	Intelligent transportation system
I2C	Inter-integrated circuit
LED	Light emitting diode
LFSR	Linear-feedback shift registers
LiDAR	Light detection and ranging
LoRa	Long range
LTE	Long-term evolution
ML	Machine Learning
PCB	Printed circuit board
PMOD	Peripheral module interface
RGB	Red-green-blue
RISC	Reduced instruction set computing
SD	Secure digital
SoC	System on chip
SPI	Serial peripheral interface
UART	Universal asynchronous receiver transmitter
UAS	Unmanned aerial system
USB	Universal serial bus
UTM	UAS traffic management
V2C/N	Vehicle-to-cloud/network
V2I	Vehicle-to-infrastructure
V2P	Vehicle-to-pedestrian
V2V	Vehicle-to-vehicle
Wi-Fi	Wireless fidelity

References

1. Gao, Y.; Tian, F.; Li, J.; Fang, Z.; Al-Rubaye, S.; Song, W.; Yan, Y. Joint Optimization of Depth and Ego-Motion for Intelligent Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–14. [CrossRef]
2. Rani, S.; Kataria, A.; Chauhan, M.; Rattan, P.; Kumar, R.; Kumar Sivaraman, A. Security and Privacy Challenges in the Deployment of Cyber-Physical Systems in Smart City Applications: State-of-Art Work. *Mater. Today Proc.* **2022**, *62*, 4671–4676. [CrossRef]
3. Gunes, V.; Peter, S.; Givargis, T.; Vahid, F. A survey on concepts, applications, and challenges in Cyber-Physical Systems. *KSII Trans. Internet Inf. Syst.* **2014**, *8*, 134–159.
4. Fantin Irudaya Raj, E.; Appadurai, M. Internet of Things-Based Smart Transportation System for Smart Cities. In *Intelligent Systems for Social Good: Theory and Practice*; Springer Nature: Singapore, 2022; pp. 39–50. [CrossRef]
5. Mahrez, Z.; Sabir, E.; Badidi, E.; Saad, W.; Sadik, M. Smart Urban Mobility: When Mobility Systems Meet Smart Data. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6222–6239. [CrossRef]
6. Liu, S. The Business Case for Infrastructure-Vehicle Cooperative Autonomous Driving. *IEEE Eng. Manag. Rev.* **2022**, *50*, 189–194. [CrossRef]
7. Pan, M.; Li, Y.; Zhang, Z.L.; Luo, J. SCCS: Smart Cloud Commuting System With Shared Autonomous Vehicles. *IEEE Trans. Big Data* **2022**, *8*, 1301–1311. [CrossRef]

8. Shi, X.; Wei, H. An Embedded Vehicular Integrative Platform Based on CDMA. In Proceedings of the 2006 2nd IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications, Beijing, China, 13–16 August 2006; pp. 1–5. [CrossRef]
9. Rawat, D.B.; Bajracharya, C.; Yan, G. Towards intelligent transportation Cyber-Physical Systems: Real-time computing and communications perspectives. In Proceedings of the SoutheastCon 2015, Fort Lauderdale, FL, USA, 9–12 April 2015; pp. 1–6. [CrossRef]
10. Chen, L.; Li, Y.; Huang, C.; Xing, Y.; Tian, D.; Li, L.; Hu, Z.; Teng, S.; Lv, C.; Wang, J.; et al. Milestones in Autonomous Driving and Intelligent Vehicles—Part 1: Control, Computing System Design, Communication, HD Map, Testing, and Human Behaviors. *IEEE Trans. Syst. Man, Cybern. Syst.* **2023**, 1–17. [CrossRef]
11. Novak, A.; Novak Sedlackova, A.; Vochozka, M.; Popescu, G.H. Big Data-driven Governance of Smart Sustainable Intelligent Transportation Systems: Autonomous Driving Behaviors, Predictive Modeling Techniques, and Sensing and Computing Technologies. *Contemp. Readings Law Soc. Justice* **2022**, *14*, 100–117. [CrossRef]
12. Kliestik, T.; Musa, H.; Machova, V.; Rice, L. Remote Sensing Data Fusion Techniques, Autonomous Vehicle Driving Perception Algorithms, and Mobility Simulation Tools in Smart Transportation Systems. *Contemp. Readings Law Soc. Justice* **2022**, *14*, 137–152. [CrossRef]
13. Pan, G.; Alouini, M.S. Flying Car Transportation System: Advances, Techniques, and Challenges. *IEEE Access* **2021**, *9*, 24586–24603. [CrossRef]
14. Zhou, J.; Tian, D.; Sheng, Z.; Duan, X.; Shen, X. Joint Mobility, Communication and Computation Optimization for UAVs in Air-Ground Cooperative Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 2493–2507. [CrossRef]
15. Shrestha, R.; Bajracharya, R.; Kim, S. 6G Enabled Unmanned Aerial Vehicle Traffic Management: A Perspective. *IEEE Access* **2021**, *9*, 91119–91136. [CrossRef]
16. Javaid, S.; Saeed, N.; Qadir, Z.; Fahim, H.; He, B.; Song, H.; Bilal, M. Communication and Control in Collaborative UAVs: Recent Advances and Future Trends. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5719–5739. [CrossRef]
17. Wilson, A.N.; Kumar, A.; Jha, A.; Cenkeramaddi, L.R. Embedded Sensors, Communication Technologies, Computing Platforms and Machine Learning for UAVs: A Review. *IEEE Sens. J.* **2022**, *22*, 1807–1826. [CrossRef]
18. Prag, K.; Woolway, M.; Celik, T. Toward Data-Driven Optimal Control: A Systematic Review of the Landscape. *IEEE Access* **2022**, *10*, 32190–32212. [CrossRef]
19. Tantawy, A.; Abdelwahed, S.; Erradi, A. Cyber LOPA: An Integrated Approach for the Design of Dependable and Secure Cyber-Physical Systems. *IEEE Trans. Reliab.* **2022**, *71*, 1075–1091. [CrossRef]
20. De Klerk, M.L.; Saha, A.K. A Comprehensive Review of Advanced Traction Motor Control Techniques Suitable for Electric Vehicle Applications. *IEEE Access* **2021**, *9*, 125080–125108. [CrossRef]
21. Ding, D.; Han, Q.L.; Wang, Z.; Ge, X. A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2483–2499. [CrossRef]
22. Al-Dulaimi, A.; Al-Rubaye, S.; Cosmas, J. Adaptive congestion control for mobility in cognitive radio networks. In Proceedings of the 2011 Wireless Advanced, London, UK, 20–22 June 2011; pp. 273–277. [CrossRef]
23. Cogliati, D.; Falchetto, M.; Pau, D.; Roveri, M.; Viscardi, G. Intelligent cyber-physical systems for industry 4.0. In Proceedings of the 2018 First International Conference on Artificial Intelligence for Industries (AI4I), Laguna Hills, CA, USA, 26–28 September 2018; pp. 19–22.
24. Shuvo, M.M.H.; Islam, S.K.; Cheng, J.; Morshed, B.I. Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review. *Proc. IEEE* **2023**, *111*, 42–91. [CrossRef]
25. Pundir, A.; Singh, S.; Kumar, M.; Bafila, A.; Saxena, G.J. Cyber-Physical Systems Enabled Transport Networks in Smart Cities: Challenges and Enabling Technologies of the New Mobility Era. *IEEE Access* **2022**, *10*, 16350–16364. [CrossRef]
26. Xing, G.; Simplicio, M.; Pillai, A. *Autonomous Vehicles: Cyber-Physical Risk on a Massive Scale*; Technical Report; IEEE Transmitter: Manhattan, NY, USA, 2022.
27. Wilson, P. *Certifying AI for Safety Critical Aircraft Systems*; Technical Report; Acubed: Sunnyvale, CA, USA, 2020.
28. Qazi, S.; Khawaja, B.A.; Farooq, Q.U. IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends. *IEEE Access* **2022**, *10*, 21219–21235. [CrossRef]
29. Demigha, S. The impact of Big Data on AI. In Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 16–18 December 2020; pp. 1395–1400. [CrossRef]
30. Rodrigues, R. Legal and human rights issues of AI: Gaps, challenges and vulnerabilities. *J. Responsible Technol.* **2020**, *4*, 100005. [CrossRef]
31. Tran, C.N.N.; Tat, T.T.H.; Tam, V.W.Y.; Tran, D.H. Factors affecting intelligent transport systems towards a smart city: A critical review. *Int. J. Constr. Manag.* **2023**, *23*, 1982–1998. [CrossRef]
32. Khairullah, S.S.; Elks, C.R. Self-repairing hardware architecture for safety-critical cyber-physical-systems. *IET Cyber-Phys. Syst. Theory Appl.* **2019**, *5*, 92–99. [CrossRef]
33. Zhang, Y.; Qian, C.; Lv, J.; Liu, Y. Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Trans. Ind. Inform.* **2017**, *13*, 737–747. [CrossRef]
34. Dutt, N.; Jantsch, A.; Sarma, S. Self-aware cyber-physical systems-on-chip. In Proceedings of the 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2015; pp. 46–50. [CrossRef]

35. Shakeri, R.; Al-Garadi, M.A.; Badawy, A.; Mohamed, A.; Khattab, T.; Al-Ali, A.K.; Harras, K.A.; Guizani, M. Design Challenges of Multi-UAV Systems in Cyber-Physical Applications: A Comprehensive Survey and Future Directions. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 3340–3385. [CrossRef]
36. Alguliyev, R.; Imamverdiyev, Y.; Sukhostat, L. Cyber-physical systems and their security issues. *Comput. Ind.* **2018**, *100*, 212–223. [CrossRef]
37. Al-Mhiqani, M.N.; Ahmad, R.; Abidin, Z.Z.; Yassin, W.M.; Hassan, A.; Mohammad, A.N.; Clarke, N.L. A new taxonomy of insider threats; an initial step in understanding authorized attack. *Int. J. Inf. Syst. Manag.* **2018**, *1*, 343–359.
38. Kurniawan, J.H.; Chng, S.; Cheah, L. The Social Acceptance of Autonomous Vehicles. *IEEE Potentials* **2021**, *40*, 39–44. [CrossRef]
39. Pelau, C.; Dabija, D.C.; Ene, I. What makes an AI device human-like? The role of interaction quality, empathy and perceived psychological anthropomorphic characteristics in the acceptance of artificial intelligence in the service industry. *Comput. Hum. Behav.* **2021**, *122*, 106855. [CrossRef]
40. Denis, A.S.; Moskolai Ngossaha, J.; Ari, A. Cyber-Physical Urban Mobility Systems: Opportunities and Challenges in Developing Countries. *Int. J. Softw. Innov.* **2023**, *11*, 1–21. [CrossRef]
41. Xilinx. What Is an FPGA? Available online: https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html (accessed on 10 April 2023).
42. Boutros, A.; Betz, V. FPGA Architecture: Principles and Progression. *IEEE Circuits Syst. Mag.* **2021**, *21*, 4–29. [CrossRef]
43. Magyari, A.; Chen, Y. Review of state-of-the-art FPGA applications in IOT Networks. *Sensors* **2022**, *22*, 19. [CrossRef]
44. Chakraborty, A.; Kar, A.K. Swarm Intelligence: A Review of Algorithms. In *Nature-Inspired Computing and Optimization*; Springer: Berlin/Heidelberg, Germany, 2017. [CrossRef]
45. Steghöfer, J.P.; Kiefhaber, R.; Bee, K.; Bernard, Y.; Klejnowski, L.; Reif, W.; Ungerer, T.; André, E.; Hähner, J.; Müller-Schloer, C. Trustworthy Organic Computing Systems: Challenges and Perspectives. In *Autonomic and Trusted Computing, Proceedings of the 7th International Conference, ATC 2010, Xi'an, China, 26–29 October 2010*; Springer: Berlin/Heidelberg, Germany, 2010. [CrossRef]
46. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A review of challenges, solutions, and applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef]
47. Handayani, K.; Anugrah, P. Assessing the implications of net-zero emissions pathways: An analysis of the Indonesian power sector. In Proceedings of the 2021 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), Jakarta, Indonesia, 29–30 September 2021; pp. 270–275. [CrossRef]
48. Tomar, D.; Tomar, P. Integration of Cloud Computing and Big Data Technology for Smart Generation. In Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 11–12 January 2018; pp. 1–6. [CrossRef]
49. Al-Rubaye, S.; Rodriguez, J.; Fragonara, L.Z.; Theron, P.; Tsourdos, A. Unleash Narrowband Technologies for Industrial Internet of Things Services. *IEEE Netw.* **2019**, *33*, 16–22. [CrossRef]
50. Gopalakrishnan, S.K.; Al-Rubaye, S.; Inalhan, G. Adaptive UAV Swarm Mission Planning by Temporal Difference Learning. In Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 3–7 October 2021; pp. 1–10. [CrossRef]
51. Aloseel, A.; Al-Rubaye, S.; Zolotas, A.; Shaw, C. Attack-Detection Architectural Framework Based on Anomalous Patterns of System Performance and Resource Utilization—Part II. *IEEE Access* **2021**, *9*, 87611–87629. [CrossRef]
52. Mohamed, M.A.; Kardas, G.; Challenger, M. Model-Driven Engineering Tools and Languages for Cyber-Physical Systems–A Systematic Literature Review. *IEEE Access* **2021**, *9*, 48605–48630. [CrossRef]
53. Escobar, L.; Moyano, C.; Aguirre, G.; Guerra, G.; Allauca, L.; Loza, D. Multi-robot platform with features of Cyber-physical systems for education applications. In Proceedings of the 2020 IEEE ANDESCON, Quito, Ecuador, 13–16 October 2020; pp. 1–6. [CrossRef]
54. Lee, E.A. Cyber Physical Systems: Design Challenges. In Proceedings of the 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, USA, 5–7 May 2008; pp. 363–369. [CrossRef]
55. Khaitan, S.K.; McCalley, J.D. Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Syst. J.* **2015**, *9*, 350–365. [CrossRef]
56. Segura, M.; Poggi, T.; Barcena, R. A Generic Interface for x-in-the-Loop Simulations Based on Distributed Co-Simulation Protocol. *IEEE Access* **2023**, *11*, 5578–5595. [CrossRef]
57. Hubert, H.; Stabernack, B. Profiling-Based Hardware/Software Co-Exploration for the Design of Video Coding Architectures. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1680–1691. [CrossRef]
58. Luyan, W.; Zhangguo, S.; Long, C. The performance analysis for embedded systems using statistics methods. *Telkomnika Indones. J. Electr. Eng.* **2013**, *11*, 4099–4103. [CrossRef]
59. Bijjahalli, S.; Sabatini, R. A High-Integrity and Low-Cost Navigation System for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 356–369. [CrossRef]
60. Namuduri, K.; Fiebig, U.C.; Matolak, D.W.; Guvenc, I.; Hari, K.; Määttänen, H.L. Advanced Air Mobility: Research Directions for Communications, Navigation, and Surveillance. *IEEE Veh. Technol. Mag.* **2022**, *17*, 65–73. [CrossRef]
61. Schmitz, M.T.; Al-Hashimi, B.M.; Eles, P. *System-Level Design Techniques for Energy-Efficient Embedded Systems*, 1st ed.; Springer: New York, NY, USA, 2006. [CrossRef]

62. Swaminathan, N.; Reddy, S.R.P.; RajaShekara, K.; Haran, K.S. Flying Cars and eVTOLs—Technology Advancements, Powertrain Architectures, and Design. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 4105–4117. [CrossRef]
63. Panda, P.R.; Catthoor, F.; Dutt, N.D.; Danckaert, K.; Brockmeyer, E.; Kulkarni, C.; Vandercappelle, A.; Kjeldsberg, P.G. Data and memory optimization techniques for embedded systems. *ACM Trans. Des. Autom. Electron. Syst.* **2001**, *6*, 149–206. [CrossRef]
64. Timilsina, L.; Badr, P.R.; Hoang, P.H.; Ozkan, G.; Papari, B.; Edrington, C.S. Battery Degradation in Electric and Hybrid Electric Vehicles: A Survey Study. *IEEE Access* **2023**, *11*, 42431–42462. [CrossRef]
65. Zhang, D.; Zhou, X.; Cheng, E.; Wan, H.; Chen, Y. Investigation on Effects of HPM Pulse on UAV's Datalink. *IEEE Trans. Electromagn. Compat.* **2020**, *62*, 829–839. [CrossRef]
66. Arandhakar, S.; Jayaram, N.; Shankar, Y.R.; Gaurav.; Kishore, P.S.V.; Halder, S. Emerging Intelligent Bidirectional Charging Strategy Based on Recurrent Neural Network Accosting EMI and Temperature Effects for Electric Vehicle. *IEEE Access* **2022**, *10*, 121741–121761. [CrossRef]
67. Shen, Q.; Yue, C.; Goh, C.H.; Wang, D. Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3763–3772. [CrossRef]
68. Sharma, P.; Gillanders, J. Cybersecurity and Forensics in Connected Autonomous Vehicles: A Review of the State-of-the-Art. *IEEE Access* **2022**, *10*, 108979–108996. [CrossRef]
69. Kukkala, V.K.; Thiruloga, S.V.; Pasricha, S. Roadmap for Cybersecurity in Autonomous Vehicles. *IEEE Consum. Electron. Mag.* **2022**, *11*, 13–23. [CrossRef]
70. Khan, W.Z.; Khan, M.K.; Arshad, Q.u.A. Cybersecurity Digital Labels for Connected and Autonomous Vehicles. *IEEE Consum. Electron. Mag.* **2023**, *12*, 87–93. [CrossRef]
71. Choi, E.; Song, H.; Kang, S.; Choi, J.W. High-Speed, Low-Latency In-Vehicle Network Based on the Bus Topology for Autonomous Vehicles: Automotive Networking and Applications. *IEEE Veh. Technol. Mag.* **2022**, *17*, 74–84. [CrossRef]
72. Yassine, A.; Souweid, S. Time-to-Market and Product Performance Tradeoff Revisited. *IEEE Trans. Eng. Manag.* **2021**, 1–16. [CrossRef]
73. Tan, Z.; Dai, N.; Su, Y.; Zhang, R.; Li, Y.; Wu, D.; Li, S. Human–Machine Interaction in Intelligent and Connected Vehicles: A Review of Status Quo, Issues, and Opportunities. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 13954–13975. [CrossRef]
74. Mizutani, K.; Yamaguchi, H.; Urino, Y.; Koibuchi, M. OPTWEB: A Lightweight Fully Connected Inter-FPGA Network for Efficient Collectives. *IEEE Trans. Comput.* **2021**, *70*, 849–862. [CrossRef]
75. Khairullah, S.S. Realization of a 16-bit MIPS RISC pipeline processor. In Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–6. [CrossRef]
76. Monmasson, E.; Cirstea, M.N. FPGA Design Methodology for Industrial Control Systems—A Review. *IEEE Trans. Ind. Electron.* **2007**, *54*, 1824–1842. [CrossRef]
77. Chen, J.; Huang, S. Analysis and Comparison of UART, SPI and I2C. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023; pp. 272–276. [CrossRef]
78. Trinh, X.D.; Le, M.C.; Huy, T. IMU Calibration Methods and Orientation Estimation Using Extended Kalman Filters. In *AETA 2019—Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 541–551. [CrossRef]
79. NXP. AN3397—Implementing Positioning Algorithms Using Accelerometers—NXP, 2007. Available online: https://www.nxp.com/docs/en/application-note/AN3397.pdf (accessed on 1 April 2023).
80. Zhou, X.; Gou, X.; Huang, T.; Yang, S. Review on Testing of Cyber Physical Systems: Methods and Testbeds. *IEEE Access* **2018**, *6*, 52179–52194. [CrossRef]
81. Allouis, E.; Blake, R.; Gunes-Lasnet, S.; Jorden, T.; Maddison, B.; Schroeven-Deceuninck, H.; Stuttard, M.; Truss, P.; Ward, K.; Ward, R.; et al. A Facility for the Verification and Validation of Robotics and Autonomy for Planetary Exploration. In Proceedings of the 44th Annual Lunar and Planetary Science Conference, The Woodlands, TX, USA, 18–22 March 2013.
82. Deshmukh, J.V.; Horvat, M.; Jin, X.; Majumdar, R.; Prabhu, V.S. Testing cyber-physical systems through Bayesian optimization. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*, 1–18. [CrossRef]
83. Zheng, X.; Julien, C.; Kim, M.; Khurshid, S. Perceptions on the State of the Art in Verification and Validation in Cyber-Physical Systems. *IEEE Syst. J.* **2017**, *11*, 2614–2627. [CrossRef]
84. Naufal, J.K.; Camargo, J.B.; Vismari, L.F.; de Almeida, J.R.; Molina, C.; González, R.I.R.; Inam, R.; Fersman, E. A2CPS: A Vehicle-Centric Safety Conceptual Framework for Autonomous Transport Systems. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1925–1939. [CrossRef]
85. Kownacki, C. Optimization approach to adapt Kalman filters for the real-time application of accelerometer and gyroscope signals' filtering. *Digit. Signal Process.* **2011**, *21*, 131–140. [CrossRef]
86. Ludwig, S.A.; Burnham, K.D. Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 1236–1241. [CrossRef]
87. Zhang, P.; Gu, J.; Milios, E.E.; Huynh, P. Navigation with IMU/GPS/digital compass with unscented Kalman filter. In Proceedings of the IEEE International Conference Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; Volume 3, pp. 1497–1502.

88. Steinbrener, J.; Brommer, C.; Jantos, T.; Fornasier, A.; Weiss, S. Improved state propagation through AI-based pre-processing and down-sampling of high-speed inertial data. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 6084–6090.
89. Patel, U.N.; Faruque, I.A. Multi-IMU Based Alternate Navigation Frameworks: Performance & Comparison for UAS. *IEEE Access* **2022**, *10*, 17565–17577. [CrossRef]
90. Seng, K.P.; Lee, P.J.; Ang, L.M. Embedded Intelligence on FPGA: Survey, Applications and Challenges. *Electronics* **2021**, *10*, 895. [CrossRef]
91. Datta, D.; Datta, B.; Dutta, H.S. Design and implementation of multibit LFSR on FPGA to generate pseudorandom sequence number. In Proceedings of the 2017 Devices for Integrated Circuit (DevIC), Kalyani, India, 23–24 March 2017; pp. 346–349. [CrossRef]
92. Sharma, D.K.; Singh, N.C.; Noola, D.A.; Doss, A.N.; Sivakumar, J. A review on various cryptographic techniques & algorithms. *Mater. Today Proc.* **2022**, *51*, 104–109. [CrossRef]
93. Sohal, M.; Sharma, S. BDNA-a DNA inspired symmetric key cryptographic technique to secure cloud computing. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 1417–1425. [CrossRef]
94. Qian, Y.; Feng, Y.; Chen, H.H. Cryptographic techniques. In *Security in Wireless Communication Networks*; John Wiley and Sons, Ltd.: Chichester, UK, 2022; pp. 51–76.
95. Zhang, J.; Prabakar, K.; Hasandka, A.; Alam, S.M.S.; Jiang, Y.; Hodge, B.M.; Gao, D.W. Power and Communications Hardware-In-the-Loop CPS Architecture and Platform for DER Monitoring and Control Applications. In Proceedings of the 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aachen, Germany, 25–28 October 2021; pp. 170–174. [CrossRef]
96. García, M.V.; Pérez, F.; Calvo, I.; Morán, G. Building industrial CPS with the IEC 61499 standard on low-cost hardware platforms. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; pp. 1–4. [CrossRef]
97. Isakovic, H.; Ratasich, D.; Hirsch, C.; Platzer, M.; Wally, B.; Rausch, T.; Nickovic, D.; Krenn, W.; Kappel, G.; Dustdar, S.; et al. CPS/IoT Ecosystem: A platform for research and education. In *Cyber Physical Systems. Model-Based Design, Proceedings of the 8th International Workshop, CyPhy 2018, and 14th International Workshop, WESE 2018, Turin, Italy, 4–5 October 2018*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 206–213.
98. Landolfi, G.; Barni, A.; Menato, S.; Cavadini, F.A.; Rovere, D.; Dal Maso, G. Design of a multi-sided platform supporting CPS deployment in the automation market. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018; pp. 684–689. [CrossRef]
99. Schmittle, M.; Lukina, A.; Vacek, L.; Das, J.; Buskirk, C.P.; Rees, S.; Sztipanovits, J.; Grosu, R.; Kumar, V. OpenUAV: A UAV Testbed for the CPS and Robotics Community. In Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, 11–13 April 2018; pp. 130–139. [CrossRef]
100. Iannino, V.; Colla, V.; Denker, J.; Göttsche, M. A CPS-Based Simulation Platform for Long Production Factories. *Metals* **2019**, *9*, 1025. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Cranfield University

CERES Research Repository **https://dspace.lib.cranfield.ac.uk/**

School of Aerospace, Transport and Manufacturing (SATM) Staff publications (SATM)

Intelligent embedded systems platform for vehicular cyber-physical systems

Conrad, Christopher

2023-07-02

Attribution 4.0 International

Conrad C, Al-Rubaye S, Tsourdos A. (2023) Intelligent embedded systems platform for vehicular cyber-physical systems, Electronics, Volume 12, Issue 13, July 2023, Article Number 2908
https://doi.org/10.3390/electronics12132908

Downloaded from CERES Research Repository, Cranfield University