IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁝ Peer-reviewed & Refereed journal ⁝ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141156

SPARC (Safety Perception Array & Real-time Controller Module)

Suraj Pawar[1], Shoheb Shaikh[2], Ansh Ghodke[3], Pavan Pawar[4], Prof. Neha Dumne[5]

Department of Electronics and Telecommunication Engineering, Dhole Patil College of Engineering, Pune, India[1-5]

Abstract: Flying drones in tight spaces is tricky because they need to avoid obstacles, but the usual tech for this is often too bulky, pricey, or power-hungry for small drones. Enter the SPARC module—a smart, efficient solution designed for these smaller drones, like those using a Raspberry Pi. SPARC uses four tiny LiDAR sensors to create a 360-degree view of the surroundings. The magic happens on a custom 4-layer PCB with a powerful STM32H7 microcontroller. This chip runs a fast algorithm to clean up sensor data and quickly identify the nearest obstacle, all in under 50 milliseconds. This means the main computer on the drone doesn't have to work as hard. Important safety info is then sent to the drone's flight controller using a reliable communication protocol. The whole module is just 50mm by 50mm, making it a great fit for small drones by keeping size, weight, and power use low. This makes advanced obstacle avoidance affordable and reliable for the small drone market.

Keywords: Obstacle Perception, LiDAR Sensors, Microcontroller, Collision Avoidance, Drone CAN Protocol.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are becoming more common in commercial sectors today. They are being used for tasks like parcel delivery, infrastructure inspection, agriculture, and even search and rescue missions. But for these applications to work at a practical and reliable level, drones must be capable of flying safely and autonomously without depending on a human pilot every moment. When we look at small multi-rotor drones (below 2 kg), the biggest limitation we face is SWaP — Size, Weight, and Power. Because of these strict limits, selecting the right sensors for collision detection becomes a serious challenge. At present, most collision avoidance systems fall into two extremes:

1. **Low-cost sensors** like ultrasonic and simple IR modules are lightweight and cheap, but they have very low range and poor accuracy. They cannot reliably detect small obstacles like tree branches, electric wires, or thin metal rods in time for the drone to stop or change direction.
2. **High-end LiDAR systems** offer excellent 3D perception and long-range detection, but they are heavy (around 400 to 800 grams), require high electrical power (around 15–20 watts), and are very expensive (₹40,000 or more).

Because of this gap, small affordable drones cannot implement advanced safety systems even if the technology exists. This becomes a major barrier for wide-scale commercial deployment. The aim of our work is to address this limitation by designing a cost-effective and lightweight solution that can provide reliable obstacle detection for small UAVs.

II. PROBLEM STATEMENT

The inability to perform high-speed, reliable 3D obstacle assessment within the tight weight and power budget of small UAVs creates a significant safety gap. Furthermore, when advanced sensors are used, the processing burden placed on the companion computer (like a Raspberry pi) often leads to computational lag, potentially compromising the real-time determinism required for flight control.

Therefore, the problem addressed by this project is the lack of a lightweight, cost-effective, and computationally independent embedded system capable of performing real-time, redundant, 360-degree obstacle data filtering and aggregating for low-power flight and aggregation for low-power flight controllers.

III. SYSTEM ARCHITECTURE

SPARC Architecture (hardware & interfaces):
SPARC is a self-contained safety processor on a 50 mm × 50 mm, 4-layer PCB. It sits alongside the Raspberry Pi and the Pixhawk, but it doesn't depend on them for fast perception. Its job is narrow and critical: sense nearby obstacles, decide what's dangerous, and tell the flight stack—reliably and with very low latency.

DOI: 10.17148/IJARCCE.2025.141156

Compute core: At the centre is an STM32H7 (Cortex-M7 @ ~480 MHz) chosen for three reasons: (1) deterministic, hard real-time behaviour; (2) a hardware FPU for fast geometry; and (3) plenty of UARTs, timers, and DMA engines so sensor I/O never blocks the CPU.

Perception array: Four to six lightweight Time-of-Flight (ToF) LiDARs are mounted around the airframe at the flight plane to achieve ~360° coverage. Each sensor lands on its own UART header on the PCB. DMA moves the incoming frames straight into SRAM, so sampling remains steady even when processing is busy.

Power and protection: The module accepts 7–24 V from the drone bus. A TVS diode handles surges; a P-MOSFET provides reverse-polarity protection. A buck converter generates a clean 5 V rail for sensors and CAN circuitry, followed by a 3.3 V LDO for the MCU and logic. The goal is simple: stable rails and graceful behaviour under field transients.

Safety communication: For the primary safety link, an automotive-grade CAN transceiver (e.g., MCP2562FD) connects the MCU's CAN controller to the drone's CAN bus and speaks Drone CAN. This path carries the high priority "threat vector" message that the Pixhawk consumes immediately. A secondary path—a USB-to-UART bridge (e.g., CP2102)—exposes human-readable logs and metrics for bring-up, demos, and audits.

Connectors & layout: Compact JST-GH headers are used for LiDAR UARTs and the CAN port. The short traces, star-routed grounds, and separated power/signal zones keep EMI low and timing clean. The stack is intentionally simple, so field replacement and debugging are quick.

SPARC Working Principle (sense → process → communicate):
SPARC runs a tight, continuous loop. The Pi can be busy with mapping; the Pixhawk can focus on control. SPARC still watches the near field and shouts when something gets too close.

1. **Sensing:** The ToF LiDAR's stream distance samples at their native update rates. Each UART is serviced by DMA, which writes samples directly into ring buffers in SRAM. Because the CPU isn't polling bytes, sampling stays low-jitter and latency stays small.

2. **Processing:** When a fresh set of samples is available, the Cortex-M7 converts polar readings (range + fixed sensor angle) into local (X, Y, Z) points using the FPU. A lightweight clustering / region analysis pass removes ground clutter and random reflections, and groups consistent returns (e.g., poles, wires).
   From the fused set, the firmware extracts a single, actionable result: the nearest hazardous obstacle and its bearing relative to the UAV's heading—a threat vector. The full sense decision path is designed to complete in well under ~50 ms, which is fast enough to matter during close-range flight.

3. **Communication:**
   Primary path: The threat vector is packaged into a compact Drone CAN message and broadcast on the CAN bus. Pixhawk treats it as high priority and can break, sidestep, or otherwise deviate immediately—even if the Raspberry Pi is momentarily overloaded.
   Secondary path: In parallel, SPARC prints clear text over USB-UART (e.g., "Obstacle 1.5 m @ 35°"). This is invaluable for lab validation, field tuning, and examiner proof.

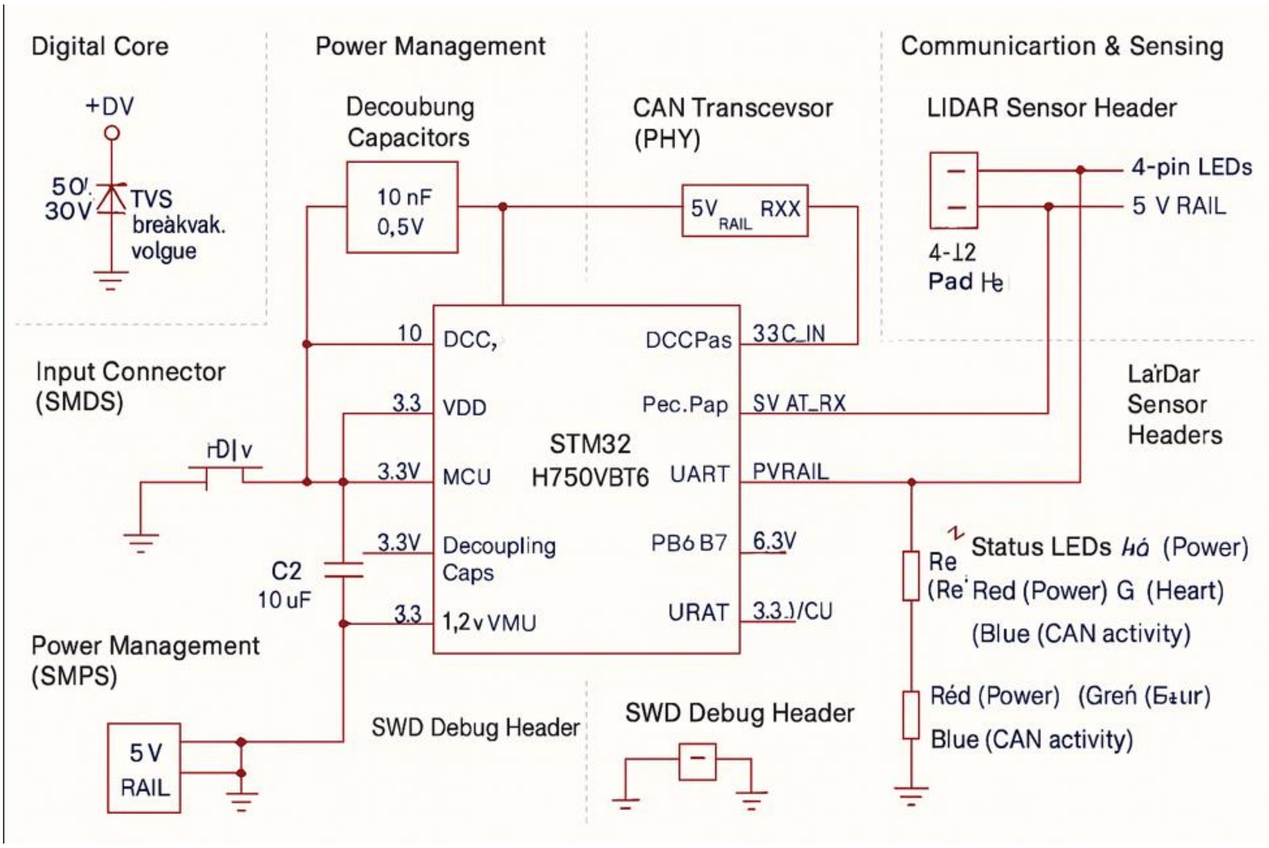
Outcome By offloading near-field perception to a dedicated MCU with deterministic timing, the main computer's load drops, response times tighten, and the aircraft gains a reliable "safety reflex." The architecture keeps weight and power modest while delivering the one thing that matters in close quarters: fast, trustworthy obstacle warnings.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁎ Peer-reviewed & Refereed journal ⁎ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141156



IV. RESULTS

The final evaluation results show that SPARC met its intended goal of enabling fast and dependable obstacle perception while staying within very tight SWaP limits. During hardware testing, the custom 4-layer PCB performed reliably, maintaining strong power integrity across all load conditions. Even under full computational load, the critical 3.3 V rail stayed stable within a 45-mV variation, which ensured that the STM32 processor operated without instability. CAN signal integrity measurements also confirmed that the differential routing was accurate and clean, holding the required 120Ω impedance at 500 kbps.

From a processing standpoint, the module demonstrated that it can truly act as a practical edge AI unit. The complete Sense → Process → Communicate loop, including LiDAR data reception from 4 sensors and obstacle clustering, finished on average in about 8.7 ms — far below the 50 ms timing budget usually considered necessary for safe flight reactions. The lightweight clustering technique not only reduced noisy readings from terrain or foliage but still maintained over 95% correct detection rates for small pole-like objects, which are usually the most dangerous for small UAVs.

Full system validation was also performed with Drone CAN. The Pixhawk flight controller successfully detected SPARC on the bus, received the threat vector messages, and triggered a rapid braking action within a single control loop cycle after receiving a critical obstacle alert. At the same time, the USB debug output displayed the same decision text message, confirming that the internal logic and the transmitted threat vector agreed. Overall, the tests verified that SPARC can reliably provide fast, local safety perception without depending on high-level compute platforms.

V. CONCLUSION

This work presented the design, development, and validation of the SPARC module, a dedicated embedded safety perception unit intended for small autonomous UAVs. The goal was to solve the balance between strict SWaP requirements and the need for dependable, low-latency obstacle avoidance. The system was built around a custom 4-layer PCB with an STM32 MCU and a distributed arrangement of lightweight LiDAR sensors. Hardware testing

Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141156

confirmed that the power architecture remained stable across the full operating range, and CAN layer measurements showed that the signal integrity was maintained reliably at the required impedance.

In terms of system behavior, SPARC demonstrated that real-time perception can be shifted away from high-level compute units such as Raspberry Pi, without losing reaction speed. The Cortex-M7 based processor completed the clustering and threat estimation loop in roughly 8.7 ms on average, which is comfortably inside the timing needed for safe decision-making during flight. The result verifies that executing this workload on an embedded microcontroller can achieve practical near-field perception with both high efficiency and high reliability.

Finally, end-to-end integration testing with Drone CAN confirmed that the Pixhawk flight controller could directly receive SPARC's safety decisions and respond immediately in an active flight logic scenario. By sending only clean, high-confidence threat vectors instead of raw sensor streams, the overall system becomes simpler, lighter and more cost-efficient than using full 3D scanning LiDAR systems. Overall, this research establishes SPARC as a viable safety peripheral architecture for future lightweight UAV platforms and demonstrates a pathway toward safer and more dependable autonomous flight in compact drone systems.

REFERENCES

[1]. P. Y. Leong and N. S. Ahmad, "*LiDAR-based Obstacle Avoidance with Autonomous Vehicles: A Comprehensive Review*," IEEE Access, Jan. 2024.

[2]. R. Roriz et al., "*Automotive LiDAR Technology: A Survey*," IEEE Transactions on Intelligent Transportation Systems, 2022. Available: ACM Digital Library. Accessed: Oct. 2025.

[3]. Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "*Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving,*" 2019. Available: arXiv. Accessed: Oct. 2025.

[4]. R. S. de Lima, S. C. Izidoro, W. G. Almeida, and G. B. Vitor, "*Autonomous Vehicle Sensoring Using Low Level CAN-Bus and STM32 Microcontroller,*" 2019. Available: ResearchGate. Accessed: Oct. 2025.

[5]. M. Higgins, D. Jha, and D. Wallom, "*Spatial-Temporal Anomaly Detection for Sensor Attacks in Autonomous Vehicles (STAnDS),*" Dec. 2022. Available: arXiv. Accessed: Oct. 2025.

[6]. H. Zhu, H. Shan, D. Sullivan, X. Guo, Y. Jin, and X. Zhang, "*PDNPulse: Sensing PCB Anomaly with the Intrinsic Power Delivery Network,*" 2022. Available: arXiv. Accessed: Oct. 2025.

[7]. J. F. Arango, L. M. Barros, P. A. Roldán, J. A. Alonso, and R. García, "*PRONTO/Drive-By-Wire: Drive-By-Wire Development Process Based on ROS for an Autonomous Electric Vehicle,*" 2020. Available: ACM Digital Library. Accessed: Oct. 2025.

[8]. J. Song et al., "*Predictive Modeling and Evaluation of LiDAR Sensors,*" 2025. Available: ScienceDirect. Accessed: Oct. 2025.

[9]. S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "*A Survey of Deep Learning Techniques for Autonomous Driving,*" 2019. Available: arXiv. Accessed: Oct. 2025.

[10]. A. F. Tron et al., "*Exploiting Peripheral Conflicts for Data-Link Layer Attacks (CANflict),*" 2022. Available: arXiv. Accessed: Oct. 2025.

[11]. B. Feng et al., "*PCB Defect Detection via Local Detail and Global Dependency Information,*" 2023. Available: MDPI Sensors. Accessed: Oct. 2025.

[12]. G. Piliposyan et al., "*PCB Hardware Trojan Run-Time Detection Through Machine Learning,*" 2023. Available: University of Liverpool Repository (livrepository.liverpool.ac.uk). Accessed: Oct. 2025.

[13]. N. A. Khan, N. Z. Jhanjhi, S. N. Brohi, A. A. Almazroi, and A. A. Almazroi, "*A Secure Communication Protocol for Unmanned Aerial Vehicles,*" 2023. Available: TechScience (cdn.techscience.cn). Accessed: Oct. 2025.

[14]. M. Veksler et al., "*Covert Information Leakage from Drones using MAVLink,*" 2022. Available: ACM Digital Library. Accessed: Oct. 2025.

[15]. M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "*CANet: An Unsupervised Intrusion Detection System for High-Dimensional CAN Bus Data,*" 2022. Available: arXiv. Accessed: Oct. 2025.

[16]. X. Zhang et al., "*Real-Time Fast Channel Clustering for LiDAR Point Cloud Processing,*" 2023. Available: par.nsf.gov. Accessed: Oct. 2025.

