Sensors & Transducers, Vol. 268, Issue 1, April 2025, pp. 37-44

Sensors & Transducers

Published by IFSA Publishing, S. L., 2025

http://www.sensorsportal.com

A COTS-based Lightweight, Low-power and Versatile Companion Computer for Nano UAVs

Rintaro OKUDERA, Yixiao LI, Yutaka MATSUBARA and Hiroaki TAKADA

Graduate School of Informatics, Nagoya University, Nagoya 464-8601, Japan

E-mail: r_okudera@ertl.jp

Received:29 Jan. 2025 /Revised: 14 March 2025 /Accepted:21 April 2025 /Published:30 April 2025

Abstract: Recent advances in Unmanned Aerial Vehicles (UAVs) have driven a rapid expansion in their application domains, ranging from logistics and surveillance to disaster relief and entertainment. In particular, Nano UAVs (weight: under 100 g) offer significant advantages in cost, safety, and regulatory compliance. However, their limited payload capacity and the high-power consumption of conventional control systems severely constrain flight time. Traditional companion computers designed for computationally intensive tasks, such as image processing, often require 2-5 times more power and nearly double the weight of the onboard flight controller, further exacerbating these limitations. In this study, we present a novel companion computer based on Sony's Spresense platform that is lightweight, low-power, and highly versatile. Weighing only 7 grams and featuring a POSIX□compliant RTOS alongside a multi□core architecture, the proposed solution is optimally tailored for Nano UAV applications. Extensive Hardware□in□the□Loop (HITL) evaluations demonstrate that our Spresense□based system consumes only 1/18 the power of the Raspberry Pi 4 Model B while delivering comparable functionality. Moreover, real Nano UAV tests using the Nano Mind 110 (weight: 36 g) confirm that integrating our companion computer results in only a modest increase in overall power consumption, thus preserving flight time. Compared to current state□of□the□art approaches, our design effectively addresses challenges in availability, extensibility, and ease of development, offering a cost□effective and practical alternative for Nano UAV systems. Future work will extend these findings through further real□world validations, including advanced navigation, obstacle avoidance, and ROS□based applications, to confirm the robustness and scalability of our approach.

Keywords: Nano UAVs, Low-power, Companion computer, Versatile UAV applications, POSIX-compliant RTOS.

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) have rapidly expanded in their application scenarios. UAVs are increasingly being considered a novel means of transportation for both people and goods, and are now widely employed in logistics, disaster relief, and entertainment. In particular, Nano UAVs – those weighing under 100 grams – excel in terms of cost, ease of maintenance, regulatory compliance, and safety, making them suitable for both indoor and outdoor applications [1]. However, when deploying Nano UAVs in professional settings, several challenges remain.

(1) Flight Time

Large UAVs can extend their flight time by incorporating high-capacity batteries and engine power [2]. However, such solutions are not feasible for Nano UAVs due to their strict size and weight restrictions, since every gram allocated to onboard electronics or sensors directly competes with the available capacity for battery storage.

(2) Power Consumption

UAVs inherently require high power consumption for their propulsion systems – especially for the motors – which results in short continuous flight time. Research efforts have focused on lightweight designs, improved aerodynamic performance, motor enhancements, and refined control algorithms to address this issue [3]. Moreover, as the range of UAV applications expands, the computational power required is also increasing. For example, consider image processing: while early UAVs (e.g., around 2010 with the AR.Drone) typically used VGA-resolution cameras (approximately 300000 pixels), it is now common in 2025 to mount cameras with FHD or even higher resolutions. When applying a linear filter, the computational load for processing an 8K image (approximately 30000000 pixels) is roughly 100 times that for a VGA image. Although this example involves relatively simple processing, more complex tasks can demand even greater computational resources. Consequently, the computational performance required by UAVs is growing exponentially.

(3) System Configuration

Fig. 1 illustrates a simplified system configuration for typical UAVs, which generally consists of two main components:

- **Flight Controller:** It handles basic tasks by computing the UAV's attitude from sensors (e.g., IMU, GNSS, pressure sensor, etc.) and controlling the motors that drive the propellers. Its applications typically run on RTOS (Real-time Operating System) based firmware such as PX4 from the open source Dronecode ecosystem [4];
- **Companion Computer:** It manages advanced tasks (image processing, SLAM, path planning, etc.) using data from cameras and depth sensors. Most of the existing solutions are based on GPOS (General-purpose Operating System) like Linux.

Typically, flight controllers are built around Cortex M microcontrollers, whereas companion computers use platforms such as Raspberry Pi 4 or NVIDIA Jetson. Compared to the flight controller, the companion computer generally consumes 2–5 times more power and generates significant heat. In addition, it is roughly twice as heavy, and the need for a heat sink further increases its overall weight. These factors combine to reduce flight time.

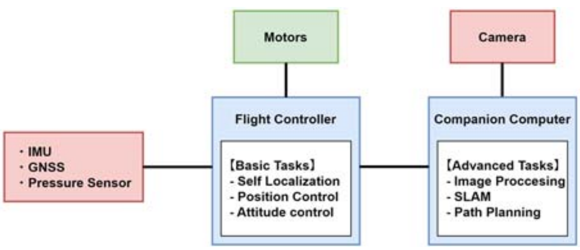

Fig. 1. System Configuration of Typical UAVs.

(4) Diversification of Functional Requirements

Initially, companion computers were primarily used for processing camera images. With the expansion of UAV applications, their roles have diversified to include autonomous flight control, AI processing, and swarm control. Since the choice of companion computers must align with the required functions, there is no one-size-fits-all solution. Moreover, more advanced functionalities typically come with increased power consumption and weight, further reducing flight time. For additional examples, please refer to the survey by Foisal et al. [5].

From the above discussion, reducing power consumption and enhancing the versatility of companion computers emerge as critical challenges. In this study, we present a new companion computer suitable for Nano UAVs built on PX4. We have implemented the necessary communication architecture and utilized multi-core task processing to achieve both energy efficiency and high performance. The effectiveness is evaluated by comparing our proposed approach with Raspberry Pi 4. This paper is an extension of work originally presented in DAUS' (International Conference on Drones and Unmanned Systems) 2025 [6]. In this paper, we present new evaluation results including experiments on real Nano UAV and more literature reviews to further validate our approach.

This section has provided an overview of our study and its contributions. Section 2 reviews related studies and products, Section 3 details the proposed Spresense-based companion computer and its implementation, Section 4 presents our HITL evaluation experiments, Section 5 presents our real-world evaluation experiments, and Section 6 positions our work within the context of existing studies before concluding in Section 7.

2. Related Work

This section describes state-of-the-art studies and products related to companion computers for Nano UAVs.

2.1. FPGA-based Approaches

Cheng et al. [7] developed a UAV control system using an FPGA as a companion computer for Bitcraze's Crazyflie 2.1 [8]. The system utilizes AMD's XC6SLX9 FPGA [9], which features 5720 slice LUTs, 11440 flip-flops, and 32 blocks of 18 Kbit Block RAM. The primary advantage of using an FPGA is its extremely low power consumption – over 100 times lower than that of a comparable embedded GPU. However, due to inherent limitations in implementing complex arithmetic circuits on FPGAs, these devices are less versatile for executing complex computations.

2.2. SoC-based Approaches

GAP8 [10] is a commercial product from GreenWaves Technologies that complies with the

Crazyflie-AIdeck [11] standard and is designed to offload mission control tasks from Crazyflie. GAP8 integrates a Ri5cy core as its host CPU, is equipped with 1.5 MB of on-chip SRAM, and features a parallel-programmable cluster of eight additional Ri5cy cores. The Ri5cy is a 4-stage pipeline core based on RV32 and compliant with the custom XpulpV2 RISC-V ISA, which includes extensions for DSP and machine learning applications (supporting 16/8-bit SIMD operations and hardware loops). In GAP8, only the Fabric Controller Core can access peripheral devices, while the remaining cores serve solely as accelerators for computation. This limitation restricts scalability and versatility for complex IoT applications. Moreover, although these systems support an RTOS, they do not support Linux [12], making it difficult to leverage existing Linux-based software assets.

In contrast, Shaheen [13] supports both RTOS and Linux, enabling the use of extensive software stacks such as ROS. It integrates a fully programmable parallel 8-core RV32 cluster accelerator. The RV32 cores in Shaheen are derived from Ri5cyNN cores and support mixed precision, achieving up to 8.5 times faster performance than Kraken. Additionally, it incorporates an RV64 core featuring advanced virtualization and security functions, along with up to 512 MB of main memory. However, Shaheen is only a prototype and has not been released commercially. Furthermore, while the SoC's power consumption is discussed, the overall board power consumption and scalability remain unclear, leaving its effectiveness as a companion computer for Nano UAVs uncertain.

3. Proposed Method

3.1. Overview

Companion computers always face trade-offs among versatility, power efficiency, and weight. The main objective of our approach is to develop a companion computer that enhances the performance of Nano UAVs by effectively balancing these three factors. Our proposed companion computer is based on Sony's Spresense, a commercial off-the-shelf (COTS) microcontroller board developed by Sony [14] (see Fig. 2 for an overview of the board's design and key components). Its compact and lightweight design makes it particularly well-suited for Nano UAV applications compared to larger solutions (e.g., Raspberry Pi 4: 85 × 56 mm, 46 g).

Spresense is equipped with a variety of useful peripherals, including an integrated Global Navigation Satellite System (GNSS) module for accurate navigation and localization. This high degree of extensibility enables users to develop a wide range of feature-rich UAV applications. Moreover, many official and third-party add-on boards are available to provide additional functionalities, such as high-resolution cameras, extra sensors, Wi-Fi, and LTE connectivity.

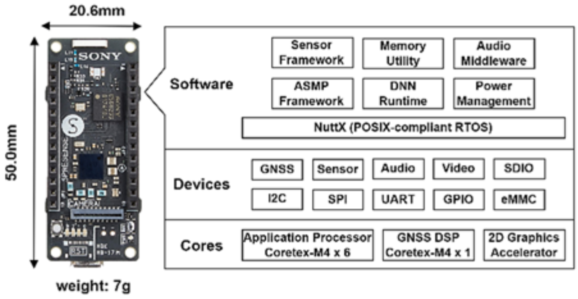

Fig. 2. Overview of Sony Spresense Board.

The application processor on Spresense features six Arm Cortex‑M4 cores, offering robust parallel processing capabilities for demanding tasks such as image processing and Deep Neural Network (DNN) inference. A dedicated Arm Cortex‑M4 DSP handles GNSS processing, further enhancing overall performance. The board's SDK is based on NuttX, a POSIX‑compliant RTOS that provides both high real‑time performance and efficient development productivity.

The unique combination of performance, power efficiency, and extensibility provided by Spresense has proven its value in space robotics applications, including lunar exploration robots [15]. In this study, we implement functionalities that enable Spresense to serve as a companion computer within the Dronecode open‑source UAV development ecosystem [4]. This approach not only facilitates efficient and rapid prototyping of Nano UAVs but also benefits from a supportive community that continuously contributes new features and improvements.

3.2. Comparison with Candidate Boards

We have also investigated other popular COTS boards with size small enough to be equipped on a Nano UAV. Table 1 shows a comparison of the Raspberry Pi Zero WH [16], Arduino Nano 3.0 [17] and Spresense on power consumption, performance and capabilities. For Raspberry Pi Zero WH, the power usage is measured with Wi-Fi and Bluetooth turned off. As the table indicates, Spresense provides computational performance comparable to the Raspberry Pi Zero WH while consuming only 1/16 of its power and less than 1/3 of the power of the Arduino Nano 3.0. Thus, we determined that Spresense was the most suitable COTS board to serve as a companion computer for Nano UAVs.

3.3. Communication Architecture Implementation

PX4 is the firmware for flight controllers in Dronecode ecosystem. Fig. 3 depicts the communication architecture between the flight controller and our companion computer. By default, PX4 and Spresense share no common protocol. Thus, we ported the MAVLink protocol [18] to Spresense due to its lightweight and reliable messaging capabilities. MAVLink enables seamless data exchange between PX4 and Spresense through a serial interface. This also allows users or developers to bridge existing ROS2 applications on Spresense using micro-ROS via MAVLink [19].

PX4 applications cannot directly use MAVLink for communication. Instead, they are required to use the publisher-subscriber middleware uORB provided by PX4. To fully leverage the functionalities of Spresense, we have also extended the PX4 firmware to enable the translation between custom MAVLink messages and uORB messages. This enables developers to offload flight controller functions – such as GNSS data acquisition – to Spresense.

Existing high-level APIs such as MAVSDK [20] offer powerful UAV control features but are not optimized for microcontrollers like Spresense [21]. We addressed this by developing custom UAV control APIs based on the MAVLink protocol. Through analysis of MAVLink message flows in existing systems, we tailored them to Spresense, creating lightweight APIs optimized for resource-limited environments while maintaining compatibility with PX4.

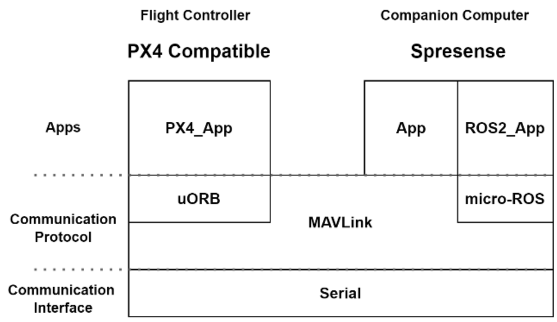

Fig. 3. Communication Architecture between the Flight Controller and the Companion Computer.

Table 1. Comparison of COTS boards with tiny form factor.

	Raspberry Pi Zero WH [16]	Arduino Nano 3.0 [17]	Spresense [13]
Power (No load) [mW]	500	100	30
DMIPS	1250	20	1200
Size	65.0 mm × 30.0 mm	43.2 mm × 18.5 mm	50.0 mm × 20.6 mm
Weight	7 grams	9 grams	7 grams
Capabilities	Wi-Fi/Bluetooth, Display output, Camera interface	-	GNSS receiver, Hi Resolution Audio I/O, Camera interface

3.4. Multi-core Task Processing

Spresense operates at 156 MHz per core, which is significantly lower than the 1.8 GHz found in Raspberry Pi 4. This lower clock speed contributes to its ultra□low power consumption. To overcome this limitation, we exploit Spresense's multiple cores to concurrently manage several tasks, such as GNSS positioning and image processing, within the confines of its limited computational resources. The Spresense SDK supports both Symmetric Multi□Processing (SMP) and Asymmetric Multi□Processing (AMP); in our implementation, we employed SMP to efficiently distribute tasks across available cores. For example, one core is dedicated to flight controller communication while others handle GNSS processing or image processing. As all cores share the same operating system, inter□core communication is straightforward. Fig. 4 demonstrates the task distribution strategy, including how unused cores are put into sleep mode to conserve power without sacrificing performance.

4. HITL Evaluation

We evaluated the performance, power efficiency, and versatility of our proposed companion computer using a Hardware-in-the-Loop (HITL) setup. All tests employed Pixhawk 6× [22] as a flight controller.

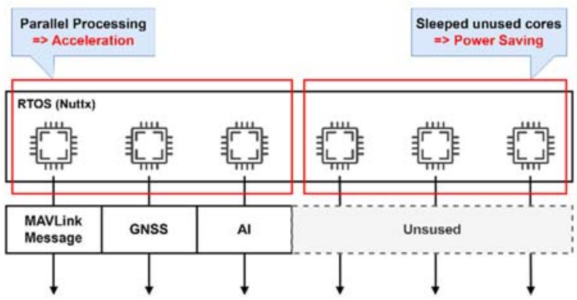

Fig. 4. Task Distribution for Multi-Core Utilization.

4.1. Communication Latency

We measured the Round-Trip Time (RTT) to assess the communication latency among Spresense, the Raspberry Pi 4 Model B, and a PC. The PC utilized a USB serial converter (DSD Tech SH-U09C5), whereas the other platforms used direct serial communication.

Fig. 5 illustrates the sequence diagram used for measuring RTT. In this test, RTT is defined as the time elapsed from sending data from the companion computer to the flight controller and receiving the returned data from the flight controller. All UARTs were configured at 921600 baud, 8N1, with no flow control. The flight controller transmitted acknowledge (ACK) messages at 50 Hz, while each companion computer sent synchronous (SYN) messages at frequencies of 1 Hz, 10 Hz, and 100 Hz and processed ACKs at 1000 Hz.

Fig. 6(A) shows the RTT distributions for the PC, the Raspberry Pi 4 Model B, and Spresense. At 10 Hz, the PC exhibited an average RTT of 35.05 ms (standard deviation (SD) = 3.97 ms), the Raspberry Pi 4 Model B 30.86 ms (SD = 1.72 ms), and Spresense 20.85 ms (SD = 0.21 ms). These results indicate that Spresense achieves a 32 % reduction in average RTT and a markedly lower variability compared to the Raspberry Pi 4 Model B, while consuming only 1/18 of its power. Based on these findings, we conclude that Spresense offers a more stable and power-efficient performance for time-critical UAV operations. Fig. 6(B) shows the RTT distributions on Spresense when varying the baud rate. Four baud rates were tested: 57600, 460800, 921600, and 2000000, while keeping all other conditions constant. The results indicate that for SYN frequencies of 1 Hz and 10 Hz, no RTT losses occur regardless of the baud rate. However, at SYN frequencies of 100 Hz and 1000 Hz, RTT losses begin to emerge. At a baud rate of 57600, communication fails when the SYN frequency is set to 100 Hz or 1000 Hz. Moreover, ACK reception losses are observed at these higher SYN frequencies regardless of the baud rate. Therefore, it is recommended that the baud rate be set between 460800 and 2000000 for reliable communication.

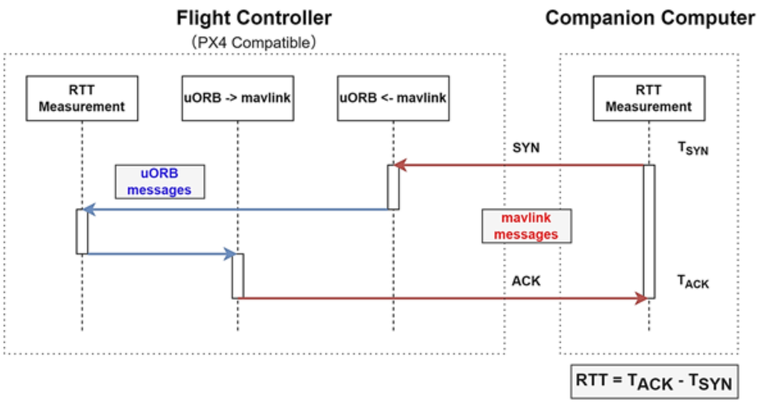

Fig. 5. Sequence Diagram of RTT (Round Trip Time).

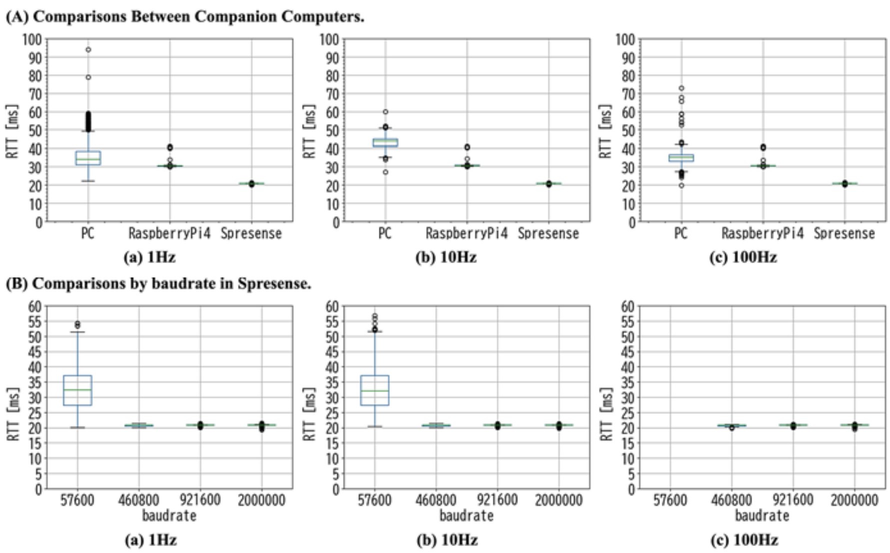

Fig. 6. RTT Results (A: Device Comparison, B: Baud Rate Variation).

4.2. Versatility and Power Consumption

We evaluated the versatility and power consumption of the developed companion computer by creating three applications – an Operation App, a GNSS App, and an AI App – on both Spresense and the Raspberry Pi 4 Model B.

Operation App: sends basic flight commands (takeoff, altitude hold, landing).

GNSS App: acquires real-time location data and sends it to the flight controller via MAVLink.

AI App: performs real-time image recognition by processing camera inputs with a pre-trained machine learning model.

Table 2 shows the measured power consumption in our HITL environment. Spresense consistently consumed about 1/18 the power of the Raspberry Pi 4 Model B while retaining comparable functionality.

Table 2. Power Consumption Comparison.

Application	Raspberry Pi 4 Model B [mW]	Spresense[mW]
Standby	3088	40
Operation App	3178	50
GNSS App	3219	88
AI App	3549	510

5. Real UAV Evaluation and Analysis

We validated the effectiveness of our proposed method using a real Nano UAV, the Nano Mind 110 [23]. Nano Mind 110 is a Nano UAV featuring a 110 mm diagonal body weight of 36 g, and a maximum take-off weight of 80 g. It is equipped with the PX4-compatible flight controller, MindRacer [24].

5.1. Experimental Environment

Fig. 7 illustrates the experimental setup for real□world testing. Instead of a 3.7 V lithium polymer battery, we used a DC stabilized power supply set to 4.2 V to emulate a fully charged battery. A weighing scale was employed to estimate thrust by monitoring changes in its readings. Additionally, to minimize the effects of ground effect [25], all experiments were conducted in an environment free of obstacles beneath the UAV.

We evaluated the relationship between power consumption and thrust for the Nano Mind 110. Fig. 8 shows the measurement results. In our experiment using a DC power supply, we were only able to measure thrust up to 25 g.

5.2. Power Consumption Evaluation

However, given that the torque and power consumption of a DC motor are linearly related, we extrapolated the linear approximation curve to estimate the relationship up to the maximum thrust of 80 g. The derived linear fit is given by:

$$\mathrm{Power[mW]} = 386 \cdot \mathrm{Thrust[g]} + 1811 \quad (1)$$

To verify the validity of this approximation, note that the Nano Mind 110 has a maximum payload of 80 g. Assuming it is equipped with four motors, and referencing the specifications of motors of the same size [26], the maximum power consumption is calculated as:

$$35360\mathrm{[mW]} = 4 \cdot 3.4\ \mathrm{[V]} \cdot 2600\mathrm{[mA]} \quad (2)$$

From the approximation curve, the estimated power consumption at a thrust of 80 g is

$$32691\mathrm{[mW]} = 386 \cdot 80\mathrm{[g]} + 1811 \quad (3)$$

Since these two values are close, the estimation of the relationship between thrust and power consumption is considered valid. Moreover, since the drive system consumes almost no power at 0 g thrust, the baseline power consumption of 1811 mW can be attributed to the flight controller.

5.3. Power Consumption Analysis

Based on Equation (1), we estimated the power consumption of the Nano Mind 110 in a hovering state. With a body weight of 36 g, the hover power consumption is calculated as:

$$15707\mathrm{[mW]} = 386 \cdot 36 + 1811 \quad (4)$$

Next, consider equipping the Nano Mind 110 with Spresense as a companion computer while running an AI application. In this configuration, Spresense consumes 510 mW and adds 7 g to the weight, resulting in a total weight of 43 g. The hover power consumption is then estimated as:

$$18919\mathrm{[mW]} = 386 \cdot 43\mathrm{[g]} + 1811 + 510 \quad (5)$$

Thus, integrating Spresense increases power consumption by approximately 20 %. In contrast, when equipping the Nano Mind 110 with the Raspberry Pi 4 Model B running a samapplication, the Raspberry Pi 4 Model B consumes 3549 mW and weighs about 65 g, yielding a total weight of 104 g. The estimated hover power consumption becomes:

$$45504\mathrm{[mW]} = 386 \cdot 104\mathrm{[g]} + 1811 + 3549 \quad (6)$$

This represents an increase of approximately 190 % relative to the base Nano Mind 110. Moreover, the total weight of 104 g exceeds the maximum take-off weight of 80 g, rendering the Raspberry Pi 4 Model B unsuitable for this application.

Given that the flight time of the Nano Mind 110 is approximately 7 minutes, the hover flight time is estimated to be around 5.8 minutes with Spresense and only about 2.4 minutes with Raspberry Pi 4 Model B. These results indicate that integrating Spresense results in a much smaller increase in power consumption – and consequently a less severe impact on flight time – compared to using Raspberry Pi 4 Model B.

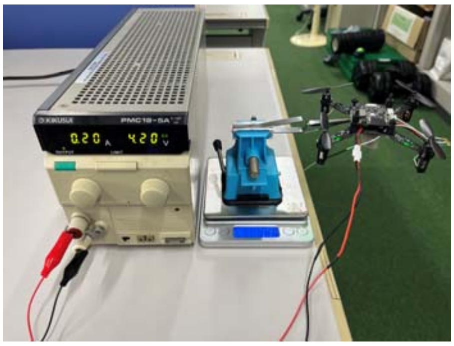

Fig. 7. Experimental Setup for Real□world Testing.

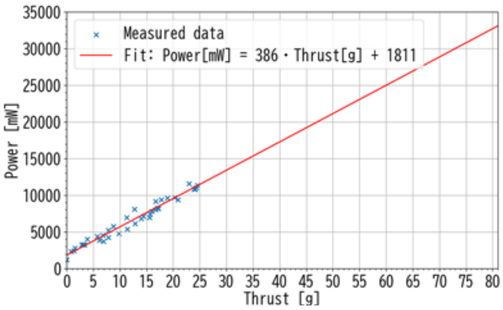

Fig. 8. Relationship between Thrust and Power Consumption.

6. Comparison with State-of-the-art

Table 3 shows a comparison between our proposed method and three state-of-the-art related studies or products.

J. Cheng et al. [7]: Propose an FPGA-based companion computer notable for its low power consumption. However, due to inherent circuit constraints in FPGAs, performing complex computations is challenging, and the system remains in the research stage, making practical deployment difficult.

AI-deck [9]: COTS microcontroller board developed by Bitcraze. AI-deck adapts GAP8 processer by GreenWaves Technologies. GAP8 itself supports the POSIX-compatible RTOS NuttX; however, AI-deck is limited to PULP OS and FreeRTOS, complicating the reuse of Linux-based development assets. Furthermore, in GAP8, only one core – the Fabric Controller Core – can access peripheral devices, while the other cores function solely as accelerators for computation. This design significantly limits its extensibility and versatility for complex IoT applications.

Shaheen [11]: Supports Linux in addition to an RTOS. Nonetheless, Shaheen is only a prototype and has not been released as a commercial product. Furthermore, while the S oC's power consumption is mentioned, the overall board power consumption and extensibility remain unclear, making its effectiveness as a companion computer for Nano UAVs uncertain.

Table 3. Comparison with State-of-the-art Companion Computers.

	J. Cheng et al. [7]	**AI-deck [9]**	**Shaheen [11]**	**Spresense (Ours) [13]**
Availability	Prototype	Product	Prototype	Product
Category	FPGA	SoC	SoC	SoC
Application Processor	-	RI5CY	CVA6	6× Cortex-M4
Accelerator	XC6SLX9	8× RI5CY	8× FLEX-V	Cortex-M4 2D Graphic Accelerator
Max Freq. [MHz]	-	250	500 – 600	156
Memory	-	1.5 MB SRAM + 8-64 MB HyperBUS	1 MB SRAM + 32-512 MB HyperBUS	1.5 MB SRAM + 8 MB Flash
OS	-	RTOS	RTOS + GPOS	RTOS
Power [mW]	≦ 50	60	200	40

Our Companion Computer: Overcomes all the issues found in these studies. Since Spresense is marketed by Sony, it is readily available. It supports NuttX thus allowing reuse of Linux-based development assets. Moreover, official and third-party add-on boards (e.g., for cameras, Wi-Fi, and LTE, etc.) are available to extend its functionality. In terms of power consumption, it achieves levels comparable to J. Cheng et al.'s FPGA-based companion computer, which boasts the lowest power consumption among the compared systems. Additionally, Spresense offers a significant cost advantage, priced at $65.00 compared to AI-deck's $240.00, making it an ideal choice for budget-sensitive applications without sacrificing performance or extensibility.

7. Conclusion

In this study, we developed a lightweight, low-power, and versatile companion computer for Nano UAVs based on the Spresense platform, offering balanced performance, straightforward development with NuttX, and robust extensibility. HITL evaluations show that, compared to Raspberry Pi 4 Model B, Spresense achieves approximately 30 % reduction in round-trip time (with the standard deviation reduced to 1/10) while consuming only 1/18 of the power.

Moreover, the real□UAV evaluations confirmed that integrating Spresense into the Nano Mind 110 results in only a modest increase in power consumption and hover power requirements – ensuring a relatively minor impact on flight time. In contrast, employing Raspberry Pi 4 Model B would dramatically increase power consumption and weight, rendering it impractical for Nano UAV applications.

Notably, our proposed method demonstrates a unique integration of availability, extensibility, and ease of development – features that previous studies could not simultaneously achieve. Future work will focus on real-world validation – indoor navigation, outdoor obstacle avoidance, and ROS 2–based applications using micro-ROS – to confirm its robustness and scalability.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 24KK0184.

References

[1]. A. Kushleyev, D. Mellinger, C. Powers, V. Kumar, Towards a swarm of agile micro quadrotors, *Autonomous Robots*, Vol. 35, Issue 4, 2013, pp. 287-300.

[2]. Yamaha Motor, Hybrid Drone Engine, https://global.yamaha-motor.com/np/hybrid-drone/

[3]. P. K. Chittoor, B. Chokkalingam, L. Mihet-Popa, A review on UAV wireless charging: fundamentals, applications, charging techniques and standards, *IEEE Access*, Vol. 9, 2021, pp. 69235-69266.

[4]. Dronecode Project, https://dronecode.org/

[5]. F. Ahmed, M. Jenihhin, A survey on UAV computing platforms: a hardware reliability perspective, *Sensors*, Vol. 22, Issue 16, 2022, 6286.

[6]. R. Okudera, Y. Li, Y. Matsubara, H. Takada, A lightweight, low-power and versatile companion computer for nano UAVs, in *Proceedings of the 1st International Conference on Drones and Unmanned Systems (DAUS'25)*, 2025, pp. 292-295.

[7]. J. Cheng, Y. Yamaguchi, A study of FPGA-based micro-drone controller, https://www.ieice.org/publications/conference-FIT-DVDs/FIT2019/data/pdf/CC-003.pdf (in Japanese)

[8]. Bitcraze, Crazyflie 2.1 Product Page, https://www.bitcraze.io/products/old-products/crazyflie-2-1/

[9]. AMD, Spartan-6 Tech Specs, https://www.amd.com/en/products/adaptive-socs-and-fpgas/fpga/spartan-6.html

[10]. GreenWaves Technologies, GAP8 Low Power Processor, https://greenwaves-technologies.com/low-power-processor/

[11]. Bitcraze, AI-Deck Product Page, https://www.bitcraze.io/products/ai-deck/

[12]. PULP Platform, https://pulp-platform.org/

[13]. L. Valente, A. Nadalini, A. H. C. Veeran, M. Sinigaglia, et al., A heterogeneous RISC-V based SoC for secure nano-UAV navigation, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 71, Issue 5, 2024, pp. 2266-2279.

[14]. Sony Developer World, Spresense, https://developer.sony.com/spresense

[15]. Takara Tomy, SORA-Q Product Page, https://www.takaratomy.co.jp/products/sora-q/

[16]. Raspberry Pi Zero W Product Page, https://www.raspberrypi.com/products/raspberry-pi-zero-w/

[17]. Arduino Docs, Nano Hardware Page, https://docs.arduino.cc/hardware/nano/

[18]. MAVLink Developer Guide, https://mavlink.io/en/

[19]. Micro-ROS, https://micro.ros.org/

[20]. MAVSDK Documentation, https://mavsdk.mavlink.io/main/en/index.html

[21]. MAVSDK Documentation, FAQ, https://mavsdk.mavlink.io/main/en/faq.html

[22]. PX4 Autopilot User Guide, Holybro Pixhawk 6X, https://docs.px4.io/main/en/flight_controller/pixhawk6x.html

[23]. PX4 Autopilot User Guide, Nano Mind 110, https://docs.px4.io/main/en/complete_vehicles_mc/NanoMind110.html

[24]. PX4 Autopilot User Guide, MindRacer, https://docs.px4.io/main/en/flight_controller/mindracer.html#mindracer-hardware

[25]. A. Matus-Vargas, G. Rodriguez-Gomez, J. Martinez-Carranza, Ground effect on rotorcraft unmanned aerial vehicles: a review, *Intelligent Service Robotics*, Vol. 14, Issue 1, 2021, pp. 99-118.

[26]. Bitcraze Store, 4×7×20 mm DC motor pack, https://store.bitcraze.io/products/4-x-7x20-mm-dc-motor-pack-for-crazyflie-2-x-thrust-upgrade