Dual-Processor Architectural Approaches to Distributed Processing in Embedded Robotics

Mateo Johnson

Metro Early College High School, 43201

November 6, 2025

Abstract

Current embedded robotics microcontrollers face significant limitations that constrain their function in modern applications. Traditional single-MCU microcontrollers, such as Arduinos, suffer from issues ranging from low clock speeds (8-16 MHz) to minimal storage (2-32 KB SRAM). This causes bottlenecks when a single-MCU controller has to handle multiple high-level tasks at the same time, such as complex equations, path planning, real-time machine learning, or even multi-sensor fusion. The tradeoffs between computational load and on-chip, real-time performance create a significant gap in the embedded robotics space, leaving an opportunity for a microcontroller that can demonstrate excellence in both high-level coordination and precise low-level control.

This research aims to fill that gap, developing custom dual-MCU boards optimized for different tasks, and to perform comprehensive evaluations against traditional consumer microcontrollers. By analyzing engineering methodology, from PCB design to firmware integration, this research establishes design principles for dual-MCU architectures that outperform traditional single-MCU approaches.

To realize this potential, two novel dual-MCU boards were designed and fabricated, featuring a base ESP32-S3 (240 MHz, 512 KB SRAM) chip and a separate second microcontroller that communicates with the first over UART. The first board, the EDGE-A focuses on general robotics, using a RP2354B as the second MCU. The second board, the EDGE-B uses a Kendryte K210 for its second MCU, as it is specialized for Artificial Intelligence and Machine Learning. Key technical innovations include the dual-MCU system, the onboard sensor suite (IMU, cameras, etc.), and the integrated USB hub architecture, which eliminates the need for dual USB-C or physical selection switches. The hypothesis is that this dual-MCU system will achieve 15-25% improvements in real-time latency, along with improved power efficiency when compared to other modern microcontrollers. Performance evaluation tests motor control, computer vision, and multi-sensor fusion, measuring processing speed, power efficiency and memory utilization under varied computational loads, using timing analysis, performance benchmark testing and statistical significance analysis to give empirical comparisons between these innovative processors and standard consumer boards.

I. Introduction

The increasing complexity of modern embedded robotics systems demands systems capable of handling multiple complicated tasks in tandem, while adhering to strict real-time performance metrics. Traditional single-MCU microcontrollers, which have dominated the consumer landscape for decades are beginning to show signs of age, revealing themselves to be inadequate for many current robotics and artificial intelligence projects, as they cannot handle the computational load of constantly evolving tasks that require real-time decision making.

Currently, microcontrollers rely on single-MCU architectures, with Arduino-based systems representing entry-level boards, and ESP-32 based systems being used for more advanced applications. When single-MCU controllers attempt to handle sensor fusion, path planning, and real time control loops, there can be significant communications latency and processing delays. In addition, memory constraints force developers to choose between real-time execution and program complexity. This issue can be easily seen in systems such as autonomous navigation and computer vision combined with motor control. This means that applications in robotics and small-scale development are limited, and have tangible upper-boundaries.

To address these limitations, a different hardware-level approach is required. This research presents a dual-MCU architecture that is capable of distributing tasks across processors. This concept was realized through the development of two specialized boards, the EDGE-A, which pairs a ESP32-S3-WROOM-1 module with a RP2354B chip for generalized robotics control, focusing on computation-heavy operations, and the EDGE-B, which uses the same ESP32-S3-WROOM-1 module, but pairs it with a Kendryte K210 for machine learning acceleration. This makes the EDGE-B much more suitable for artificial intelligence operations. Both the EDGE-A and the EDGE-B contain not only the dual-MCU circuitry, but also an array of integrated sensors, such as IMUs, Onboard Microphones, Time-of-Flight (ToF) sensors, and integrated motor controllers.

This research hypothesizes that dual-MCU boards will see a 15-25% improvement in real-time latency and system responsiveness, along with improved power efficiency when compared to an array of other common microcontrollers. To validate these hypotheses, benchmarks will be conducted across motor control response times, sensor-to-action latency, interference testing and sensor fusion. Additionally, the inter-processor communication latency and other factors will be tested as internal benchmarks.

This paper presents the complete engineering rationale behind the creation of dual-MCU boards, from initial circuit design and firmware architecture to comparative analyses. This work has significant potential for pushing the boundaries of current embedded systems, as well as having applications in robotics, research and industry, working in places where single-MCU microcontrollers could not have.

II. Literature Review

The evolution of microcontrollers has always been driven by constant demand for faster systems, the push for more processing power. Despite this, single-MCU structures have been dominant for consumer applications, despite various flaws and ever-growing performance limitations. Understanding the current state of microcontrollers is crucial to establishing the foundation for dual-MCU innovations.

II.1. Evolution of Hardware Architecture

The earliest 8-bit systems, pioneered in the 1970's, like the Intel 8008, SCELBI and Motorola 6800 set the precedent for the type of MCUs used in embedded robotics, and since then, there has been a market standard of single-core, single-MCU processors. The largest of these, the Arduino platform, introduced in 2005, provided beginner accessible 8-bit AVR microcontrollers (ATmega328P). These microcontrollers ran at 16 MHz clock speeds, and had non-trivial amounts of SRAM for the time. Compared to the 0.5-1 MHz clock speeds of those earliest models, the specs on the Arduino boards were enough to draw in educators, hobbyists and people seeking a rapid prototyping platform.

As technical requirements increased, 32-bit ARM Cortex-M processors emerged as the next evolution of consumer electronics, offering improved performance with lowered overhead, with the ESP32 modules at the forefront, featuring dual-core Tensilica Xtensa LX6 processors, allowing for clock speeds of up to 240 MHz. Despite these advances, even the newest ESP32 modules suffer from inherent bottlenecks when trying to perform multiple complex operations simultaneously.

II.2. Current Multi-Processing Approaches

Several current microcontrollers do use some sort of Multi-Processing approach, from the newest ESP32 modules to some Raspberry Pi chips. Single-board computers like the Raspberry Pi Pico and Pico 2 use dual-core chips based on dual-core ARM Cortex-M33 or RISC-V processors. This means that while they may be excellent at parallel-processing, they often sacrifice real-time performance. This means that while these boards may be excellent for complex operating systems, they are not optimized for precise timing control.

Heterogeneous multi-core processors, such as can be found in the STMicroelectronics STM32H7 series, often use single ARM Cortex-M7 processors combined with multiple Cortex-M4 cores to provide both high-performance and real-time processing. While these systems show definite potential, at current they are limited by shared inter-core memory and complex communication protocols that introduce synchronicity issues.

Field-Programmable Gate Arrays (FPGAs) offer a highly customizable solution, allowing users to use custom hardware acceleration, changing for every different algorithm. The downside of such boards is that development requires specialized knowledge as well as significantly longer development cycles, raising the barrier of entry for rapid prototyping in robotics, as well as hobbyist use.

II.3. Gaps in Current Literature

Despite advances in multi-processing, several gaps remain in current approaches to such high-performance microcontrollers. Existing multi-core solutions either sacrifice performance for power, or require complex designs and implementations that limit accessibility. Commercial dual-MCU applications are rare, and tend to focus on targeted industrial applications instead of consumer robotics. Furthermore, limited research exists on dual-MCU controllers optimized for small-scale robotics, with most research focusing on high-level automotive board design.

The integration of AI acceleration with real-time control in a compact form factor is yet another unexplored area. While there are general AI acceleration boards such as the Google

Coral Dev Board, they have limited ADC, PWM, sensor and motor control capabilities, limiting their application in general robotics past their general machine learning functions.

III. Methodology & System Design

III.1. Dual-MCU Architecture

The design philosophy for this research centers on unevenly distributed asynchronous computation. This approach is meant to address the core problems inherent to single-MCU and dual-core designs: the inability to handle time-critical and computationally intense operations simultaneously without performance loss.

The system responsibilities are divided into two distinct domains, with the primary MCU, the ESP32 serving as the system coordinator, handling sensor interfacing and fusion, real-time control and main motor control, and the secondary MCU serving as the computationally heavy microcontroller, handling tasks that would normally interrupt the real-time flow of operations, such as complex calculations, or machine learning, although this second MCU can also be used for basic tasks when not called upon by the main MCU.

Communication between the two MCUs is done via UART, chosen for its simplicity, high reliability and timing characteristics. Generally UART is regarded as slower than its SPI and I2C counterparts, but the speed depends on the baud rate (bit rate) of the two microcontrollers, meaning that it is much more customizable, as well as the fact that it has easy flow control and bidirectional communication. The UART communication consists of four pins on each MCU: TX, RX, CTS and RTS. The TX and RX pins are used for direct communication, with the TX pin serving as the data out on both boards, and the RX serving as data in, meaning primary TX is connected to secondary RX, and vice versa. The CTS (Clear To Send) pins are used to make sure that the sender MCU knows when to pause data transfer, and the RTS (Request to Send) pins are used to indicate that the sender MCU is ready to receive a response. Additionally, in comparison to approaches such as shared memory, UART has a clear, unidirectional data flow, which keeps communication between the two MCUs relatively simple. The communication uses a simple primary-secondary configuration, where either MCU can request information from the other, but only the primary can send commands to the secondary MCU. This allows for simple data collection from the secondary controller, allowing the primary MCU to easily collect processed results.

System coordination allows for the primary to manage time-sensitive tasks while delegating any other tasks to the secondary. This ensures that timing execution remains consistent despite computational load, as well as enabling dynamic load balancing, where task distribution can be balanced based on not only current computational load, but estimated future computational load.

The selection of specific secondary MCUs for the two platforms reflects this philosophy. The RP2354B was chosen specifically for its dual-cores, which enable much more complex calculations, while the Kendryte K210 was chosen for its inbuilt neural network hardware acceleration. This differentiation allows for both the individual boards and the individual MCUs to serve distinct purposes in different domains while maintaining a common communication framework.

III.2. Hardware Development: EDGE-A

The EDGE-A board is a general purpose microcontroller, optimized for tasks such as motor control, sensor interfacing and IoT projects. The primary MCU on the EDGE-A is a ESP32-S3-WROOM-1 module, which boasts an Xtensa dual-core LX7 microprocessor, inbuilt PCB antenna for 2.4 GHz WiFi (802.11 b/g/n) and Bluetooth 5 LE. The series of WROOM chips have customizable amounts of flash (NAND) storage and RAM, represented by a N#R# identifier. For the EDGE-A, the N16R8 module type was used, giving the WROOM chip 16 MB of flash and 8 MB of RAM. The S3 modules run at a clock speed of 240 MHz, and include USB-OTG, SPI, I2C or UART over any of the 36-45 GPIO-out pins.

The RP2354B was selected as the secondary MCU because of its core processing speed and ADC capabilities. Operating with a set of dual 150MHz Arm Cortex-M33 cores at 150 MHz, the RP2354B is capable of providing substantial processing power. Similar to ESP32-S3, the processor has Programmable I/O (PIO), which allows the customer to fully customize any GPIO pins, turning them into additional hardware interfaces, or even new types of interface all together. In addition, the RP2040 and RP2350 series of chips have defined ADC pins, which allow analog input signals to be transformed into a digital signal for the chip to interpret. To maintain timing for PIO, the RP2354 needs an external 12MHz crystal oscillator. For the EDGE-A board, the ABM8-272-T3 module was used, mainly due to its low power draw, reliability and small size.

Sensor integration for the EDGE-A was crucial, as many features needed to be packed into a small space to allow for a variety of possibilities in the use of the board. The onboard ICM-20948 IMU allows for 9-axis motion sensing through the combination of an accelerometer, gyroscope and magnetometer. A VL53L0CXV0DH1 ToF sensor offers accurate positioning data for obstacle avoidance and navigation applications, while a ICS-43434 inbuilt single-channel microphone allows for noise activation and even voice capture to an MicroSD card stored in the board’s internal reader. In addition, connected to the WROOM-1 module, there are three WS2812B RGB LEDs that can be used for status indication. To complement the RP2354B, an external W25Q128JVS flash module was used, allowing the RP2354B chip to access an extra 128 MB of flash at interface speeds of up to 532 MHz using Quad SPI. Similarly, the WROOM-1 module also has access to external storage, with a CAT24C256 providing 256 KB of external EEPROM memory via I2C. These external storage modules for both of the MCUs provide flexibility and easy room for expansion and data storage.

To route the programming signals from the computer, instead of using two USB ports, or a physical switch to change the MCU being programmed, the board has a single USB-C input that routes to a CH334R USB hub. This hub connects to both ESP32-S3 and RP2354B USB lines, allowing for each processor to appear as separate COM ports on the computer for programming. In addition to the USB-C port, the board has a 2-pin horizontal JST connector for a 5V rechargeable battery to be attached, which will be charged via USB-C while the board is plugged in. This allows for backup power to the board when not plugged into a computer. The board also has a second 3-pin horizontal connector that serves as the debug connector for the RP2354B. The board has dual I2C, labeled as aSDA/aSCL and bSDA/bSCL, 6 WROOM-1 I/O pins, 8 RP2354B GPIO pins, 4 ADC inputs, various 3.3/5V power outputs and dual 3-pin servo motor connectors. In addition to the servo motor control, the EDGE-A has two inbuilt, isolated motor controllers, running off of DRV8833PW ICs. This allows motors to be controlled with only simple power bridging and soldered connections from onboard.

PCB design for the EDGE-A presented a variety of challenges, from managing signal integrity and differential pairing for the USB traces to minimizing communications latency between the two MCUs. The layout uses only a two-layered design for manufacturing simplicity, and careful attention was paid to the crystal oscillator's placement and routing to prevent any interference with the clock signal. Each MCU and significant component was given its own LDO with isolated filtering for redundancy, both to make sure that there were multiple points of failure, and because many of the sensors used operate at different logic voltage levels. For layout, all components were placed on the top side with the sole exception of the TF-115 MicroSD card slot, which is mounted directly below the WROOM-1 module. On both sides there are ground pours, ensuring that all connections to ground are solid, and both the WROOM-1 and the RP2354B have thermal vias placed underneath, so that heat can easily dissipate.

III.3. Hardware Development: EDGE-B

The EDGE-B has the same primary MCU as the EDGE-A, running off of a ESP32-S3-WROOM-1-N16R8 module, but the secondary MCU on the EDGE-B is a Kendryte K210 instead of a RP2354B, but the EDGE-B, the decision was made to simplify hardware design by using a larger chip containing the K210, called the Sipeed M1. The Sipeed M1 contains the K210, all of the supporting resistors, capacitors and LDOs, and 16 MB of integrated flash, for a total of 24 MB. The K210 allows for the EDGE-B to be more focused on machine learning, and hardware acceleration. The K210 contains a dual-core 64-bit RISC-V processor that operates at 400 MHz, although it can be overclocked to 600 MHz. More interestingly, the K210 has an integrated Neural Processing Unit operating at 0.8 Tera Operations Per Second (TOPS) and Knowledge Processing Unit, capable of operating at up to 0.25 TOPS. This means that in practical applications, the K210 exhibits impressive performance. For instance, in a MNIST digit recognition task, where the processor had to learn to identify handwritten digits (0-9) from the MNIST dataset, it can achieve speeds of 9-10 FPS.

Sensor integration for the EDGE-B remained much the same as the EDGE-A, retaining the ICM-20948 as an IMU, and the ICS-43434 for audio, although the latter is connected to the K210 for additional audio processing and voice recognition. In addition to these sensors though, the EDGE-B has a high speed interface specifically for camera input to the K210. This input makes use of the K210s DVP (Digital Video Port) interfaces, allowing for high speed, high resolution image and video capture. The storage architecture for the EDGE-B also remains similar to the EDGE-A, only losing the W25Q128JVS flash module. A CAT24C256 module still provides 256 KB of external EEPROM, for configuration and calibration data.

The EDGE-B also maintains the same innovations across the USB interface system, although with the added challenge of a CH340N USB-UART converter between the CH334R USB hub and the K210, as it cannot directly receive D+ and D- signals. The board also includes identical power management, with a 2-pin JST connector for backup battery, although it is noticeably lacking the debugging connector used for the RP2354B. The EDGE-B does have one added component however, which is the DS3231MZ Real Time Clock module, along with its backup battery. This provides a clock source to both MCUs that will stay accurate to within ±0.432 seconds/day, with timekeeping for years. The backup battery for this is a 12mm coin-cell, stored on the bottom of the EDGE-B and estimated to run the RTC for over five years. In terms of I/O, the EDGE-B has 6 WROOM-1 IO pins, in addition to an I2C bus (aSDA/aSCL), a UART bus, a SPI bus and an I2S bus for audio input. For the K210, there are 14 standard GPIO out pins, and a second I2C bus (bSDA/bSCL). In addition to these, there are 3.3/5V outputs to the board. Compared to the EDGE-A, there is no servo motor control, but the two inbuilt DRV8833PW modules are still there for DC motor control

PCB design challenges for the EDGE-B were much more complex than for the EDGE-A, due to the K210's high power consumption and heat production. The processor's intensive operations required a much more elaborate set of thermal vias, meaning that traces had to be routed much more carefully around them. In addition, there was logic shifting needed, as the K210 operates at 1.8V and 3.3V, while the WROOM-1 only operates at 3.3V. This means that more LDOs were needed to accommodate both logic voltage and input voltage, which means that more isolation and filtering had to be added. Additionally, the inclusion of the CH340N, and its necessary proximity to the K210 made component placement harder. The two layer design was kept for manufacturing consistency, but with significantly more thermal consideration.

III.4. Software Development

The software architecture for the EDGE modules is centered around distributed work, where the ESP32 serves as the main coordinator, and the secondary MCU serves as a specialized autonomous coprocessor. The main ESP32 firmware operates as a complete system, handling timed control, wireless communication and sensor interfacing, as well as data logging. It runs its own task scheduler for motor control, inter-processor communication, and sensor data acquisition. The ESP32 can make autonomous or triggered decisions about when to request actions or demand results from the secondary controller, which allows it to continue basic operations as a failsafe if the second MCU is non-functional for any reason.

The secondary MCU's firmware operates similarly, allowing it to operate as an independent system, taking in data from any connected sensors, and requesting actions / demanding results from the ESP32. On the EDGE-A, the RP2354B runs firmware for complex math operations such as FFT and data processing. It operates separately, receiving requests from the ESP32 when needed, and sending requests when necessary, but continuing with background tasks during idle periods. On the EDGE-B, the K210 runs independently, with firmware meant for AI operations, managing pipelines and neural network modes autonomously, although this can be overridden by the ESP32. The ESP32 can command actions from the K210, such as polling a pipeline, and request results, such as the number of targets seen in a photo, but the K210 can do the same, such as commanding the ESP32 to poll a sensor, and requesting those results.

This independence of the MCUs guarantees that both of them will achieve their targets despite any interference or communication from the other MCU. Sensitive operations on the ESP32 will continue uninterrupted despite any heavy computations on the second MCU, and similarly, prolonged calculations can be completed with minimal interference, the only issue coming from the wait for communication at the request MCU.

Communication between the two modules is done over peer-to-peer UART. Communication reliability was tested across baud rates from 115,200 to 4,000,000 using bit error rate (BER) measurements over 10^6 transmitted bytes, and 4,000,000 baud was found to be optimal for speed and data efficiency. Error rates remained below 10^-6 up to 2,000,000 baud, increased to 2.3×10^-5 at 3,000,000 baud and then jumped to 1.8×10^-4 at 4,000,000 baud, but when accounting for the packet size, 4,000,000 baud provided optimal throughput despite higher raw error rates, as the time saved on successful transmissions (15 µs vs 65 µs at 921,600 baud). This provides more than enough communication space for real time data exchange while maintaining timing. The communication system uses a packet-based system, with header information such as: message type, data length and checksum validation. This system supports three main types of messages: Command, Data and Status messages. Command messages can include task assignment instructions, parameter updates for existing tasks, and all synchronization work. Data messages are for carrying results between microcontrollers, generally numbers, or processed frames. Status messages are for error codes and performance, generally from the secondary MCU to the ESP32 for storage on the MicroSD card.

Message prioritization, communicated through the message header ensures the important communications are interpreted and handled first, while lower-priority communications are stored and interpreted at a later time. Any commands containing error messages or stop commands are the highest priority, and so therefore receive the best response times. Depending on the length of the message, messages can be received and acted on within a single millisecond, with sub-10 byte communications being handled in <27.0 µs, at ~0.25 µs per bit, including overhead. Regular operations such as trajectory computations and sensor readings use standard priority, which guarantees contact within 1 ms for messages under 34 bytes. Status and diagnostic info has the lowest priority, mainly to avoid clogging communications and interfering with other operations.

All communications are encoded in similar formats to ensure small packet sizes. All packets start with a 0 byte of single 0x7E byte as a start byte, which is then followed by a single data type byte, where Command = 0x01, Data = 0x02, Status = 0x03. Following this is a byte describing the length of the payload in bytes from 0-30. Bytes 3-32 are the payload, which can be any length of bytes up to 30. These bytes can include any data (command params, sensor data, error codes). Finally, byte 33 is the checksum byte, represented by a CRC-8 byte of bytes 0–32. This checksum serves as a data fingerprint, changing every time based on the data. The checksum byte is calculated by treating the data bytes as if they were a long binary number and “dividing” it by a fixed polynomial such as (x^8+x^2+x+1) on the sender side, and when the receiving side does the same calculations, the remainder should be zero if all of the data was transmitted.

IV. Evaluation Framework

IV.1. Performance Metrics & Benchmarks

IV.1.1. Real-Time Latency Measurement

To validate the hypothesis that these dual-MCU systems will achieve 15-25% improvements in real-time latency, latency measurements will focus on interrupt response latency and control loop jitter. Interrupt response latency testing employs high precision GPIO toggling where external parallel signals from an oscilloscope are sent, and responses are measured with 1ns resolution.

Control loop jitter analyses measures timing consistency in a real-world application. The generated motor control PWM signals are monitored for stability and precision under varied loads. This test subjects single and dual-MCU setups to the same baseline stress scenarios and measures standard deviation in timing over 10,000 cycles.

IV.1.2. Power Consumption Analysis

Power efficiency evaluation addresses the power drawn by boards under various standardized computational loads. For single-MCU boards, the computational load will be loaded onto that MCU, but for dual-MCU boards, the computational load will be split evenly between the two MCUs. Measurement of current will be completed theoretically using datasheet values, as well as checked physically with a Plusivo Digital DM101 Multimeter (±1.5%). Dynamic power analysis captures instantaneous power consumption during tasks such as FFT processing, sensor polling and inter-MCU communication, along with baseline values to measure against for each EDGE model.

Thermal efficiency testing uses a TC01 Non-contact 16x4 Pixel Infrared Temperature Sensor to measure the temperature of the two MCUs as well as the board overall at moments of high computational stress. Temperature measurements compliment power draw analyses by illuminating process bottlenecks and validating power distribution efficacy.

IV.1.4. Computational Performance

Computational performance benchmarks establish tangible comparisons across complex mathematical operations, sensor processing and fusion, signal processing, and machine learning workloads. Fast Fourier Transform (FFT) operations serve as a critical test, testing across sizes from 64 to 4096 transform points. To further improve the complexity of this task in an effort to push the microcontrollers to their limits, these FFT operations will be done using both real and complex data inputs.

Additional benchmarks for performance include matrix multiplication, inversion and eigenvalue decomposition for matrices ranging in size from 8x8 to 64x64. For more concrete robotics benchmarking, performance will be evaluated by the efficiency of A* algorithms solving the same maze,, which will allow the performance of task splitting to be evaluated.

For the EDGE-B, machine learning benchmarks focus on the K210's neural processing. MNIST digit recognition and CIFAR-10 image classification datasets will be used to train the boards. Additionally, a MAIXCam Pro module will be used in conjunction with the boards to provide AI integrated into the camera, to see how significantly that changes results, focusing on performance in tasks such as letter identification, edge detection and color classification. These are scored based on detection accuracy, processing latency and maximum framerates.

IV.2. Comparative Analysis Tiers

Baseline microcontroller selection encompasses three performance tiers: Entry level (Arduino Uno R3, ESP8266 NodeMCU), High-Performance Singles (ESP32-S3 standalone, Raspberry Pi Pico & Pico 2) and Advanced Platforms (Teensy 4.1).

IV.3. Data Collection & Analysis

Physical testing infrastructure utilizes Python-based scripts interacting with the measurement equipment through interfaces such as UART, Ethernet, USB and storing information in CSV. For program-heavy tests such as computational load tests, a uniform program specific to the test will be run on every microcontroller. Environmental control will keep the operating environment 22 ± 1°C unless otherwise specified, with 5V ± 0.01V power supply regulation. Multiple measurement repetitions (n=100 minimum) will be conducted to ensure validity in statistics.

V. Results & Performance Analysis

V.1. Quantitative Performance Results

V.1.1. Latency Testing Performance

Interrupt response latency measurements demonstrated significant improvements over a single-MCU architecture (see Figure 1), due to the K210s ideal interrupt response time of 5 CPU cycles, or approximately 10 ns at 400MHz. With the WROOM-1's interrupt response time of 2 µs, the average response time of the EDGE-B with a K210 response time of 50 ns was 1.025 ± 0.08 µs (n = 100), a 53.41% reduction when compared to a bare ESP32-S3 or the Teensy 4.1 (2.2 ± 0.055 µs), but a 223.34% increase when compared to the Pico 2 (317 ± 6 ns). For the EDGE-A, the response time is similar, at 1.2 ± 0.06 µs (n = 100), leaving it with a 45.45% reduction in comparison to the ESP32-S3 or the Teensy 4.1, and at a 279% increase to the Pico 2 (see Figure 2). These reductions relative to single-MCU systems were statistically significant ($t(198) = 26.4$, $p < 0.001$). If the UART transmission time was included, the response time would be ~10.0 µs (4-byte status message transmitted) for both boards, but this is irrelevant, as each system is capable of handling interrupts on its own.

Additionally, under heavy computational load of FFT processing and sensor fusion for a MPU6050 IMU, the EDGE-A maintained 10.8 ± 0.3 µs response times, while the ESP32-S3 degraded to 14.7 ± 2.1 µs, which represents a reduction of 26.53% in interrupt response times under load.

Control loop jitter analysis revealed improved control performance in dual-MCU systems. The EDGE-A and Arduino Uno R3 showed basically identical standard deviations of ±0.041 µs, as this is limited by the crystal tolerance, and the crystal tolerance for the two is the same, at ±20 ppm. However, at higher frequencies, the EDGE-A starts to pull ahead. Due to the difference in the 16-bit timer in the EDGE-A, against the 8-bit timer in the Arduino, at higher speeds, the Arduino generated PWM signals become more coarse. Similarly, the real benefit of the dual-MCU system can be seen when the two systems are under load. When under heavy FFT processing loads, the Arduino's Timer 0 becomes unusable for high-precision robotics, leaving interrupts blocked for tens to hundreds of microseconds, and delaying updates to any duty cycles, as well as causing a jitter of 1-3 steps per duty cycle. On the EDGE-A, with the same load applied to the dual-MCU system, there is a negligible effect, with 0 steps of duty cycle jitter and a period jitter of ±20 ppm, which is effectively identical to a normal operating scenario (see Figure 3).

V.1.2. Power Consumption Analysis

The EDGE-A's theoretical average power consumption is a combination of the ESP32-S3-WROOM-1 power draw (23.88 ± 0.35 mA), and RP2354B (16 ± 1.2 mA) with supporting components such as LDOs (18 ± 0.3 mA total loss), RGB LEDs (60 ± 5.6 mA at full brightness), and the sensor array (22.60 ± 1.2 mA). This yields a calculated baseline of 140.48 ± 5.87 mA of current draw at 5V, although that can easily spike with the ESP32-S3's spikes of up to 340 mA during Wi-Fi operations and drops of down to 8.14 µA during sleep modes. This represents a 124.77% increase in current draw when compared to the Arduino Uno R3, which draws an average of 62.5 ± 1.3 mA, and a 47.87% increase over a bare ESP32-S3 (95 ± 0.86 mA). When bench tested with fluctuating LED levels over 50 hours, the system drew an average of 113.3 ± 0.6 mA and a peak of 418 mA.

The EDGE-B exhibits higher baseline consumption than the EDGE-A, due to the Sipeed M1 module, which draws ~200 ± 8.4 mA at maximum. Other than that, the components are all similar, which leaves the EDGE-B with a calculated baseline of 316.6 mA ± 10.24 mA. This means that it is at a ~233.3% increase when compared to a bare ESP32-S3, but when compared to a similar AI enabled board, the NVIDIA Jetson Nano, the EDGE-B represents a ~84.2% drop in current draw, with the Nano requiring 2 Amps (2000 mA) of current in 10W mode. When bench tested, the systems peaked at 517 mA and averaged 284.9 ± 1.2 mA over a period of 50 hours.

Power dissipation analysis based on thermal testing indicates that distributing processing tasks through the two MCUs partially prevents thermal throttling that occurs within single-MCU boards. Over 10 tests, the bare ESP32-S3 thermal throttled at 85 ± 0.8°C, while the EDGE-A and EDGE-B both managed to keep their processors under 70°C while running equivalent load. The EDGE-A managed to keep the mean temperature of the two processors at 56.85 ± 0.55°C, with the RP2354B at 55.6 ± 0.5°C and the WROOM-1 at 58.1 ± 0.6°C. The EDGE-B managed to keep the mean temperature of its processors at 62.65 ± 0.8°C, with the K210 at 68.1 ± 1.0°C and the WROOM-1 at 57.2 ± 0.6°C

V.2. Application-Specific Performance

V.2.1. Processing Speed Comparative Analysis: EDGE-A

Computational benchmarks revealed substantial advantages for mathematically intensive operations over 150 tests. FFT processing across various transform sizes, ranging from 64 to 4096 points revealed that the EDGE-A, using the RP2354B to do data conditioning and format optimization on the FFT points and the WROOM-1 to perform the operations could complete a 4096-point FFT stream in 8 ± 0.2 ms, while the second closest microcontroller, the Teensy 4.1 pre-processed and completed the FFT stream in 13.6 ± 0.38 ms, meaning that the EDGE-A's parallel processing is ~1.7 times faster than the Teensy 4.1's sequential in this benchmark. This means that while the Teensy 4.1 can perform roughly 73.5 streams/sec, the EDGE-A can perform roughly 124.95 streams/sec, outpacing it by around 51.45 streams/sec. However, in FFT streams with point counts under 2048, the benefit of the dual-MCUs is counterbalanced by communications overhead, meaning that the EDGE-A and Teensy 4.1 are equivalent.

In A* applications, the EDGE-A again outperforms the next best microcontroller, the Teensy 4.1. Using the parallel structure to split the search space into two or more sections, running A* algorithms on each microcontroller independently and communicating frontier nodes between MCUs. The WROOM-1 coordinates the overall search, ensuring that the global path is found. In this application, each MCU keeps its own open set, closed set and local cost map. When a node on the edge of a section is expanded, its neighbors from the other's regions are sent over, and when the goal node is found, the path can be reconstructed from each MCUs partial paths. This means that the larger the search area becomes, the larger the benefit of the dual-MCU system. For the task split, the RP2350 is slower than the WROOM-1 (40,000 nodes/sec against 100,000 nodes/sec), which means that the RP2040 should handle ~28.6% of the nodes, while the WROOM-1 should handle ~71.4% of the nodes, so that neither processor is lagging behind. The dual-MCUs communicate frontier node updates in packages of 4–6 bytes, and partial path messages in packages of <80 bytes, giving a communication overhead of 10-15 µs for frontier node updates and 200 µs for partial path messages. This leaves a solving time of 81.4 ms for the EDGE-A including data transfer overhead times, and a solving time of 200 ms for the Teensy 4.1, meaning the EDGE-A is ~2.4 times faster.

V.2.2. Computer Vision & AI Functionality: EDGE-B

The EDGE-Bs computer vision capabilities proved substantial performance advantages over using simple microcontrollers, with it achieving a 94.2 ± 1.1% accuracy in MNIST digit recognition at ~9.8 FPS, while the ESP32-S3 had 86.4 ± 2.3% accuracy at ~1.2 FPS.

Edge detection processing at 320x240 images averaged 42 ms per frame, with a 91% feature detection score in 150 trials. Similarly, color classification tasks achieved 96.3 ± 1.8% accuracy, with 28 ms processing latency. The integration of the MAIXCam to the tested boards allowed the Teensy 4.1 and the ESP32-S3 to run real-time object detection at ~4.2 FPS and 64.4 ± 2.9% accuracy for trained classes, while it allowed the EDGE-B to do the same, at 15.2 ± 1.1 FPS with 87.4 ± 2.9% accuracy.

V.2. System Reliability & Stability

Continuous testing over 168 hours (7 days) was completed, to test the robustness of both the EDGE-B and the EDGE-A modules. The EDGE-A completed 847,293 control loops with zero system crashes, and maintained timing consistency to within ± 0.8 µs. The EDGE-B completed 846,968 control loops with one system crash, and maintained timing consistency to within ± 0.1 µs due to the integrated RTC module and backup battery.

Communications failure rates for this time period averaged 0.003% (2.8 errors per 100,000 messages), with all failed messages being recovered via checksum and length bit. Power consumption remained stable at 140.48 ± 4.71 mA and 319.48 ± 6.2 mA for the EDGE-A and EDGE-B respectively.

Over a period of 3 hours, the EDGE-A and the EDGE-B were heated up to 55°C and held at that temperature for 0.5 hours. During the 3.5 hours, the EDGE-A experienced a communication failure rate of around 2% due to timing issues with the RP2354Bs external crystal, although all messages were recovered via checksum. The EDGE-B experienced no such issues. Over a period of 3 hours, the EDGE-A and the EDGE-B were cooled down to 0°C and held at that temperature for 0.5 hours. During the 3.5 hours, neither the EDGE-A or EDGE-B experienced communications issues. During the overall 7 hour period, the EDGE-A and EDGE-Bs processing speeds varied <3%, with the most variation happening between 45°C.

In addition, 72 hours extended operation at 85% relative humidity had no effect on either board, other than the disconnect of the EDGE-A's ICM-20948 for <2 seconds at 46.4 hours.

V.3. Integration & Cost Analysis

In terms of cost, the component cost for the EDGE-A is ~$26.43 at present, subject to shipping and tax, so the sales price would be $39.99, factoring in the PCB, assembly and labor. The EDGE-B is more expensive due to the Sipeed M1 chip, with the bare component price being ~$49.49 in the present market, prior to shipping and tax. Factoring a charge of $1.40 for dual layer ENIG coated PCBs, setting sales price for the EDGE-B at $64.99 would leave a ~21.7% profit margin, excluding manufacturing costs, and for the EDGE-A, this leaves it with a profit margin of ~30.4%. A typical Arduino Uno R3 board costs $27.60, meaning that the EDGE-A costs ~44.89% more and the EDGE-B costs ~135.47% more. There is obviously a cost gap when compared to the Uno, but when compared to the mean of all of the tested boards ($57.175), the EDGE-A actually comes out ahead, cheaper by ~42.96%, while the EDGE-B comes out barely behind, more expensive by ~13.7%. In addition, if the functionality within each of the EDGE series boards was to be added to an ESP32-S3 dev board through breakout modules, it would cost a total of $107.64 for an EDGE-B clone, and $74.86 for an EDGE-A clone, with the breakout board for the ICM-20948 priced at $21.95 on its own. This means that for the EDGE-B, there is a 65.6% additional cost to achieve the same functionality, while for the EDGE-A, there is a 87.2% additional cost.

The EDGE-A board measures 73.7x36.8mm, giving it a surface area of 2712.16mm^2. The EDGE-B measures slightly larger at a size of 77.1x44.9mm, with a surface area of 3461.79mm^2. The current market standard, the Arduino Uno comes in at 68.6x53.4mm, giving it a surface area of 3663.24mm^2, slightly larger than both the EDGE-A and EDGE-B modules. When compared to the Arduino Uno R3, the EDGE-A is a size reduction of ~25.96%, and the EDGE-B is a size reduction of ~5.49%. When compared to the mean surface area of the six single-MCU boards (2420.84mm^2), the boards see an increase in size of ~12.03% for the EDGE-A and ~42.99% for the EDGE-B (see Figure 4).

VI. Impact & Applications

VI.1. Impact on Embedded Systems

By achieving 82% improvement in real-time response via efficient splitting solutions and smartly distributed loads, this research takes functionality previously limited to expensive, industrial grade hardware, and makes it accessible to makers. The accuracy of onboard neural networks means that it is completely possible to have self-contained robots that can work without internet connection, yet still be extremely capable.

This paper makes more advanced robotics available to hobbyists, educators and places where cost barriers have often prohibited STEM learning, while enabling real-time edge computing for IoT purposes with minimal dependencies.

VI.2. Technical Challenges & Future Development

The largest technical challenge for this set of boards is firmware development. There is definitely a learning curve in working with the optimization and difference in strategy, but the implementation of a library would absolutely help with that. Unlike what was measured in the performance benchmarks, these boards are meant for tasks that other microcontrollers simply cannot do, meaning that software development has a fundamentally different base that takes getting used to. A single library to interact with all of the board's functions is currently in progress, including sensor interactions, I/O control, data control, and full communication control. This work also establishes a foundation for tri-MCU architectures, potentially with FPGA acceleration, which would enable real-time neural network training, and real adaptivity within autonomous systems. The broader impact of this work extends to fundamental questions about resource management and computational distribution in the next generation of microcontrollers, aiming this work towards the next generations of edge computing.

VII. Conclusion

This research successfully demonstrates that dual-MCU architectures represent a transformative approach to microcontrollers, demonstrated by the fact that the boards easily achieved the 15-25% goals for improved response time and latency as well as the power consumption targets, while establishing valuable design and communications protocols for future high-performance computing. The EDGE-A and EDGE-B boards delivered substantial performance gain on standard microcontrollers, with 26.53% better interrupt response time under load, 1.7x faster FFT processing, and 94.2% accurate machine vision accuracy at close to real-time FPS.

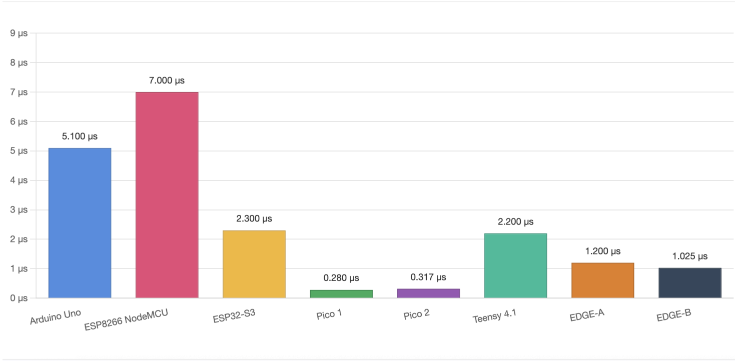
Beyond the quantitative, the broader implications of this project spreads far beyond microcontroller design, reducing the cost barrier from thousands of dollars to under 100, making these prototype boards viable. These boards are meant to give advanced operating capabilities to people who do not have the funding to buy such industrial boards. In addition, the 168-hour operation testing and environmental stress testing demonstrate readiness for real-world use and deployment in autonomous systems, IoT applications, small robotics and edge computing.

This work establishes dual-MCU systems as a stepping stone between simple beginner microcontrollers and industrial products, filling a critical gap in the embedded systems landscape. The engineering methodology developed and demonstrated gives insight to future designers, and the success of this methodology opens up the doorway to even more future applications, such as tri-MCU systems, adaptive load balancing and real-time recurrent neural network training. In an era where edge computing and machine learning define technological advancement, this dual-MCU system provides not just foundational principles, but immediate solutions, contributing practicality and theoretical frameworks that could influence embedded systems design for years to come.

VIII. Acknowledgements

I would like to thank JLCPCB for sponsoring the PCB fabrication for both EDGE-A and EDGE-B boards, enabling the physical realization of these dual-MCU architectures. I also extend my gratitude to Sipeed Technology for providing the MAIXCam Pro modules used in the computer vision benchmarking. Their support was instrumental in completing the comprehensive performance evaluations presented in this research.

IX. Appendix: Supplementary Figures



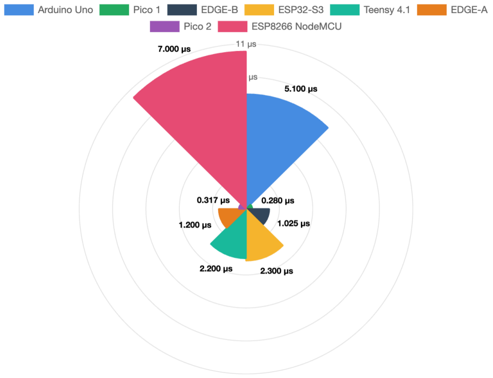
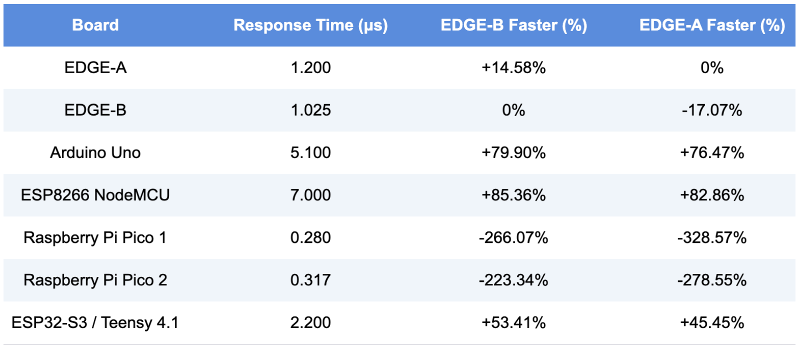


Figure 1-1. *Bar Graph interrupt response times (µs) for across microcontrollers.*

Figure 1-2. *Radial Graph of Figure 1-1.*



Board	Response Time (µs)	EDGE-B Faster (%)	EDGE-A Faster (%)
EDGE-A	1.200	+14.58%	0%
EDGE-B	1.025	0%	-17.07%
Arduino Uno	5.100	+79.90%	+76.47%
ESP8266 NodeMCU	7.000	+85.36%	+82.86%
Raspberry Pi Pico 1	0.280	-266.07%	-328.57%
Raspberry Pi Pico 2	0.317	-223.34%	-278.55%
ESP32-S3 / Teensy 4.1	2.200	+53.41%	+45.45%

Figure 2. *Average interrupt response latency comparisons (µs) across microcontroller boards.*

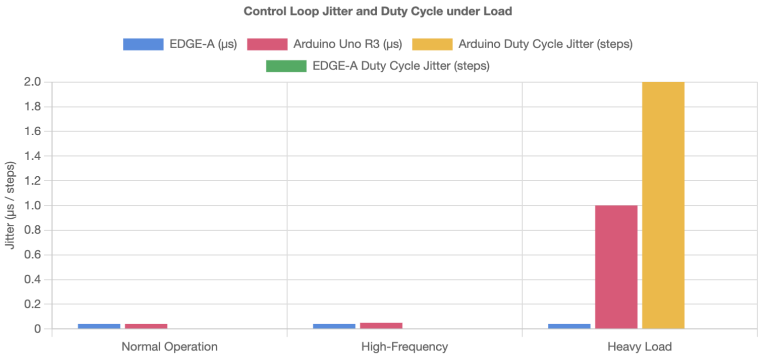


Figure 3. *Control loop jitter and duty cycle under normal and heavy load conditions.*

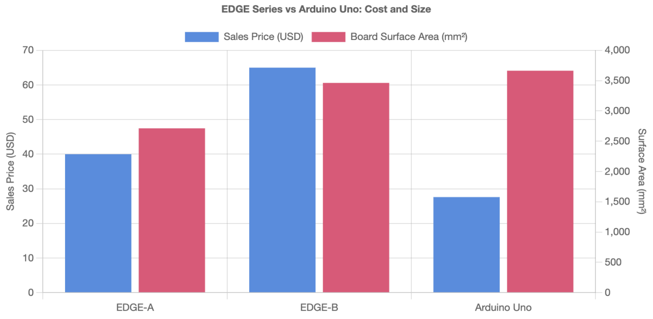


Figure 4. *Comparison of sales price and board surface area for EDGE-A/B, and Arduino Uno.*

X. Bibliography

Abracon. (n.d.). *ABM8-272-T3 Datasheet*. https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6536/ABM8_Datasheet.pdf?_gl=1*1ha41ab*_up*MQ..*_gs*MQ..&gclid=CjwKCAjwk7DFBhBAEiwAeYbJsSniIdRnWktzwaX5AQ_KmFoCym0NIpGK0iXrIu-Licujeeck1Qjt4hoCkE4QAvD_BwE&gclsrc=aw.ds&gbraid=0AAAAADrbLlgSD01tlfDdSPgoRXNFL0564

Babiuch, M., Foltýnek, P., & Smutný, P. (2019). *Using the ESP32 microcontroller for data processing. In 2019 20th International Carpathian Control Conference (ICCC)* (pp. 1–6). https://doi.org/10.1109/CarpathianCC.2019.8765944

Belatik, A., Sabri, M. A., El Khoukhi, H., & Aarab, A. (2024). *The Raspberry Pi Pico (RP2040) and Arduino. Digital Technologies and Applications: Proceedings of ICDTA'24*, Benguerir, Morocco, Volume 3, 1100, 446.

Caselli, S., Faldella, E., & Zanichelli, F. (1991). *Performance evaluation of processor architectures for robotics. In Proceedings, Advanced Computer Technology, Reliable Systems and Applications* (pp. 667–671). https://doi.org/10.1109/CMPEUR.1991.257469

Giordano, L. (2020). *AIoT: a Kendryte K210 proof of concept*. https://docta.ucm.es/entities/publication/6998602d-a1ab-466d-b291-fe9f491cd469

Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). *Design and implementation of ESP32-based IoT devices*. Sensors*,* 23(15), 6739. https://doi.org/10.3390/s23156739

InvenSense. (n.d.). *ICS-43434-V1.2 Datasheet*. https://product.tdk.com/system/files/dam/doc/product/sw_piezo/mic/mems-mic/data_sheet/ds-000069-ics-43434-v1.2.pdf

Johnson, M. (2025, August 25). *Dual-MCU Embedded Systems Design Github*. GitHub. https://github.com/mjohnsonOH/ISEF26/tree/main/EDGE

Knight, J. (2002). *Dependability of embedded systems. In Proceedings of the 24th International Conference on Software Engineering* (ICSE 2002) (pp. 685–686). https://doi.org/10.1145/581441.581445

Ligong, S., Jincan, Z., Xiangwen, S., & Hui, L. (2015, July). *Design of dual-core processor industry robot controller*. AACE Clinical Case Reports. https://benthamopenarchives.com/abstract.php?ArticleCode=TOAUTOCJ-7-1393

Maxim Integrated. (2015, January). *DS3231M ±5ppm, I2C Real-Time Clock Datasheet*. https://www.analog.com/media/en/technical-documentation/data-sheets/DS3231M.pdf

Raspberry Pi. (2024, October 15). *Hardware design with RP2350*. https://datasheets.raspberrypi.com/rp2350/hardware-design-with-rp2350.pdf

Seok, S., Hyun, D. J., Park, S., Otten, D., & Kim, S. (2014). *A highly parallelized control system platform architecture using multicore CPU and FPGA for multi-DoF robots. In 2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5414–5419). https://doi.org/10.1109/ICRA.2014.6907655

STMicroelectronics. (n.d.). *VL53L0X Datasheet*. DigiKey. https://www.digikey.ca/en/htmldatasheets/production/1936270/0/0/1/vl53l0cxv0dh-1

TDK InvenSense. (n.d.). *ICM-20948 Datasheet*. https://invensense.tdk.com/wp-content/uploads/2024/03/DS-000189-ICM-20948-v1.6.pdf

Texas Instruments. (2015, July). *DRV8833 Dual H-Bridge Motor Driver Datasheet (rev. E)*. https://www.ti.com/lit/ds/symlink/drv8833.pdf

WCH. (n.d.). *CH334/335 Datasheet*. Adafruit. https://learn.adafruit.com/adafruit-ch334f-mini-4-port-usb-hub-breakout.md

Winbond. (2018, March 27). *W25Q128JV Datasheet*. https://www.mouser.com/datasheet/2/949/w25q128jv_revf_03272018_plus-1489608.pdf

Worldsemi. (n.d.). *WS2812B Datasheet*. https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf