My Key Projects at tiSpace

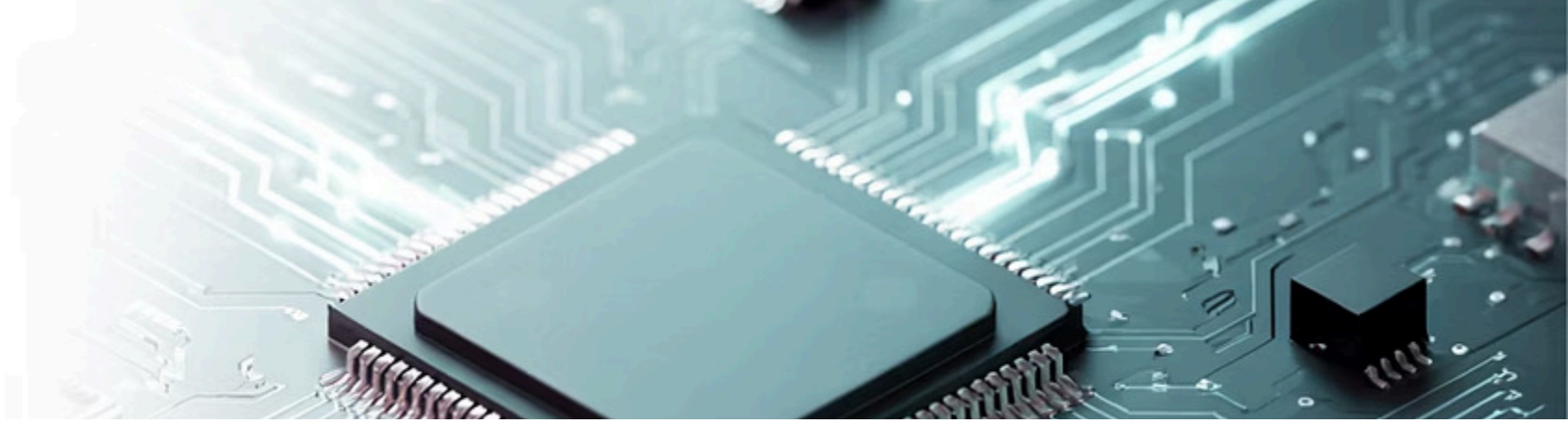
Demonstrating Expertise in Embedded Systems, Firmware Development, and Automation

Yi-Chien Chiang (江易謙)

Overview of Key Projects

Here's a glimpse into the diverse projects I've spearheaded at tiSpace, showcasing expertise in embedded systems, firmware development, and automation.

1	Strain Gauge Board Firmware	STM32F1, libopencm3, CAN bus	Reliable sensor data reporting for monitoring
2	Accelerometer Board Firmware	STM32F4, libopencm3, FreeRTOS, CAN bus	High-frequency acceleration data for dynamic systems
3	Thermocouple Hub Firmware	STM32F1, libopencm3, CAN bus	Multi-channel temperature monitoring with compensation
4	Altimeter for Parachute Deployment	STM32F4, libopencm3, FreeRTOS, CAN bus	Precise altitude-based control for safety-critical applications
5	Launch Box Software	C++ on Raspberry Pi 4, WS2812, Websockets	Intuitive launch control with real-time status indicators
6	Sensor Data Loss Investigation & Optimization	Firmware & server-side fixes (DMA, circular buffers)	Reduced data losses in subsystems and ground control
7	InfluxDB Management Page	C++ Crow library	Efficient data management for EGSE Server
8	ESS Auto Testing Design	Automated software for environmental tests	Streamlined validation for rocket subsystems under stress



Project 1 - Strain Gauge Board Firmware

This project involved developing critical firmware for a strain gauge board, implemented on an `STM32F1` microcontroller using `libopencm3` for bare-metal control. It provides robust, real-time strain measurement capabilities for various hardware setups.

1

ADS1118 Driver Development

Developed and thoroughly verified the `ADS1118` driver, enabling precise 16-bit analog-to-digital conversion for accurate strain readings.

2

System Flow Design & Calibration

Designed a comprehensive system flow, incorporating `Vinn` calibration during initialization and `R7` calibration with flash storage for persistent settings. This ensures consistent and reliable measurements.

3

CAN Bus Data Reporting

Implemented continuous 100Hz resistor value reporting and on-demand offset data reporting over the `CAN` bus, facilitating real-time monitoring and diagnostics.

4

Customizable Build Options

Created flexible `Makefile` options, allowing for the generation of binaries with customizable `CAN` IDs, enhancing deployment versatility.

5

Challenge & Outcome

Successfully addressed challenges in ensuring real-time accuracy, resulting in a highly robust monitoring solution for critical strain measurements in hardware.

Project 2 - Accelerometer Board Firmware

This project focused on developing robust firmware for an accelerometer board utilizing an `STM32F4` microcontroller, `libopencm3`, and `FreeRTOS` for efficient multitasking.

ADXL375 Driver

Implemented and verified the `ADXL375` driver, supporting SPI burst mode, FIFO buffering, and GPIO interrupts for reliable data acquisition.

High-Frequency Data Output

Achieved continuous 800Hz acceleration data reporting via `CAN bus`, enabling real-time monitoring of dynamic systems.

Axis Calibration

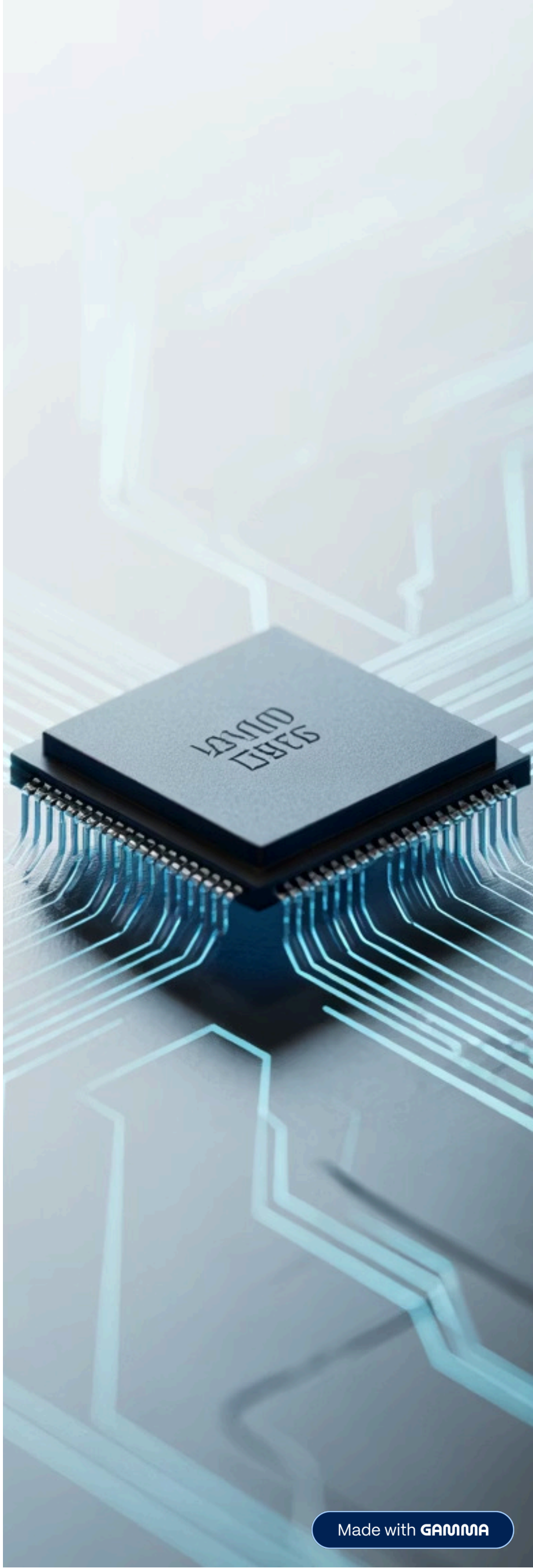
Designed comprehensive calibration functions for the X, Y, and Z axes, with data stored in flash for persistent and accurate measurements.

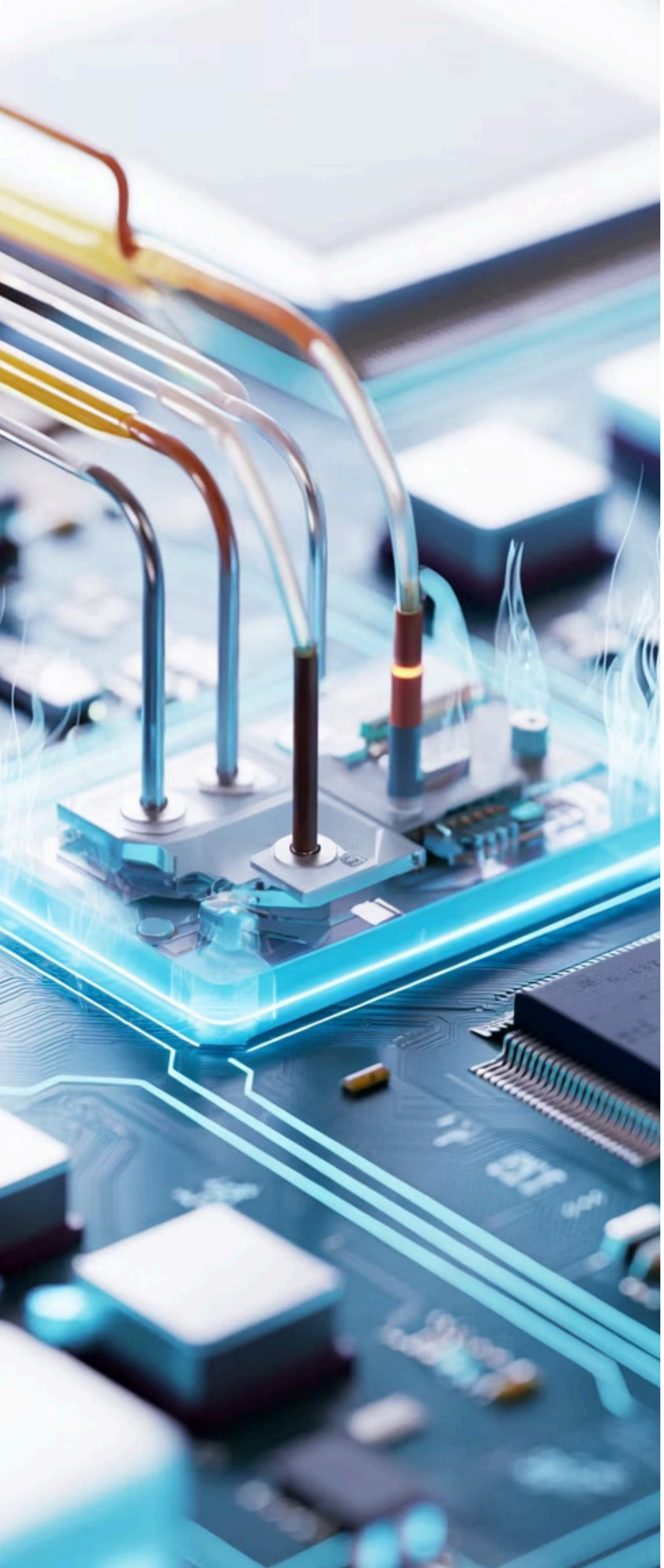
Calibration Software

Developed an accompanying calibration software/website to provide user-friendly adjustments and simplify configuration.

Impact

This development enabled high-precision motion tracking and vibration testing, crucial for dynamic environments and critical aerospace applications.





Project 3 - Thermocouple Hub Firmware

This project focused on developing robust firmware for a thermocouple hub, leveraging an `STM32F1` microcontroller and `libopencm3` to create a reliable multi-channel temperature monitoring system.

ADS1120 Driver

Developed and verified the `ADS1120` driver, enabling high-precision 24-bit analog-to-digital conversion for accurate temperature readings.

Cold Junction Compensation

Integrated Cold Junction Compensation (CJC) to ensure precise temperature measurements, even with varying ambient conditions.

CAN Bus Data Output

Configured continuous 10Hz data output for four thermocouples over the `CAN` bus, providing real-time thermal monitoring.

Impact

The successful implementation resulted in a reliable and accurate multi-channel temperature hub, essential for thermal monitoring in critical subsystems.

Project 4 - Altimeter for Parachute Deployment

This project involved developing critical firmware for an altimeter designed for parachute deployment, implemented on an `STM32F4` microcontroller with `libopencm3` and `FreeRTOS` for real-time operation.

DPS310 Driver Development

Developed and verified the `DPS310` driver, enabling accurate barometric pressure sensing for precise altitude determination.

PWM Servo Control

Implemented `PWM` control for precise servo actuation, triggering parachute deployment at critical, pre-defined altitude thresholds.

CAN Protocol Integration

Integrated `CAN` protocol for robust communication, allowing target pressure thresholds to be configured and stored persistently in flash memory.

Compensated Measurement

Developed algorithms for compensated measurement results, enhancing altitude accuracy despite environmental variations like temperature and humidity.

Impact

This development was critical for safety-sensitive applications requiring precise altitude-based control, ensuring reliable and accurate parachute deployment in dynamic scenarios.

Project 5 - Launch Box Software

This project involved developing critical C++ software for a launch box on a `Raspberry Pi 4`, integrating `WS2812` LEDs and a physical button for intuitive launch control and system monitoring.

Integrated Control Interface

Developed software to seamlessly integrate `WS2812` LEDs for visual feedback and a button for triggering launch sequences.

State of Health (SoH) Indicator

Implemented an `SoH` LED matrix, subscribing to EGSE Server telemetries via WebSockets to provide real-time system status.

Secure Launch Trigger

Configured the physical launch button to securely initiate the launch sequence via an `HTTP POST` request to the server.

Custom GPIO Driver

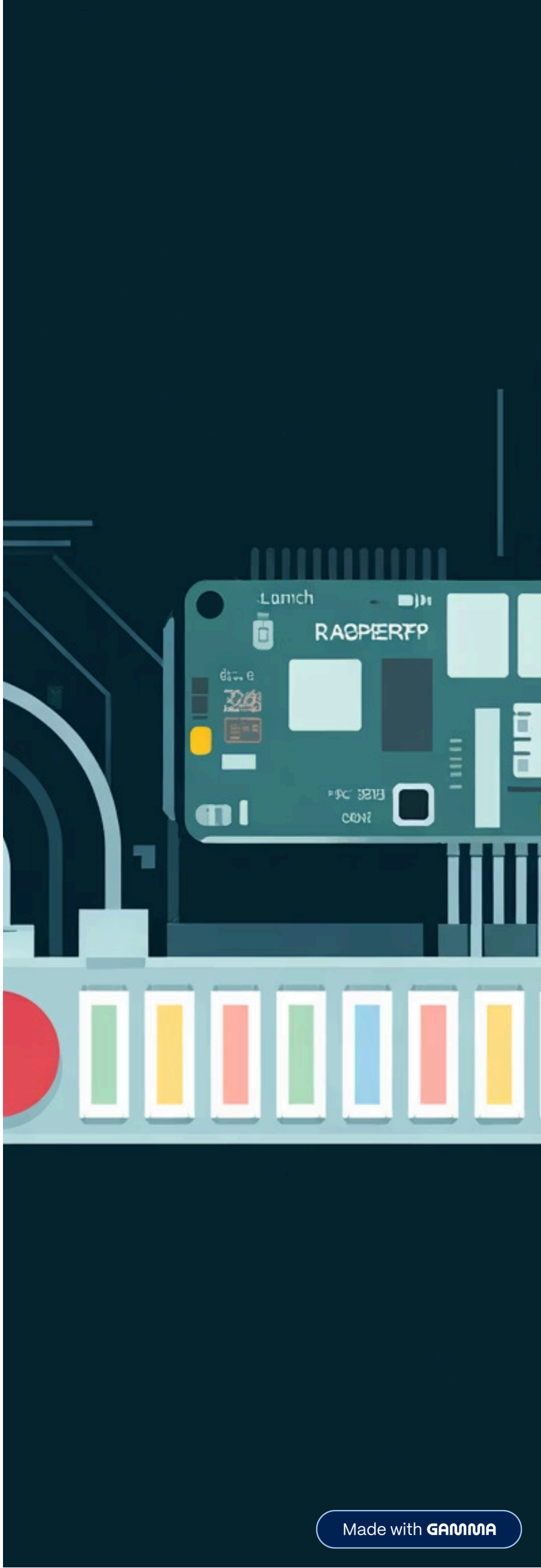
Developed a custom `Raspberry Pi C++ GPIO` driver and `HTTP Client` to manage hardware interactions and server communication efficiently.

Maintainable Codebase

Utilized a readable `SoH` list with macros, ensuring easy modifications and enhanced maintainability for future updates.

Result

This project delivered a user-friendly and reliable interface, streamlining ground support operations for rocket launch sequences.



Project 6 - Investigating Sensor Data Losses

This project focused on systematically analyzing and resolving critical sensor data loss issues across various subsystems and the ground control system, significantly enhancing overall system reliability.

Firmware Optimization

Replaced `FreeRTOS` queues with circular arrays in bay controller firmware, minimizing data copy time and boosting sensor data handling speed.

Efficient UART Transmission

Optimized `UART` transmission by implementing direct peripheral copying using `DMA`, ensuring faster and more reliable data transfer.

Server Stability Enhancements

Resolved `EGSE` Server InfluxDB storage issues by preventing rapid socket open/close cycles, enhancing data integrity and system robustness.

Impact

These targeted fixes and optimizations led to a significant reduction in data losses, resulting in vastly improved system reliability and the accuracy of critical telemetry data.

Project 7 - InfluxDB Management Page for EGSE Server

This project involved developing a web-based management interface for the EGSE Server's InfluxDB database, utilizing the `C++ Crow library` to provide robust data handling capabilities.

CSV Data Export

Implemented functionality to export selected data ranges to `CSV`, allowing users to specify timestamps for precise and efficient data positioning and analysis.

Data Manipulation Tools

Provided user-friendly tools for modifying and deleting individual data entries, ensuring high data integrity and administrative control within the database.

Outcome

This initiative significantly simplified database administration, streamlined data access, and enhanced the overall integrity of critical telemetry data for ground support operations.

Project 8 - ESS Auto Testing Design

This project involved designing and implementing automated software for Environmental Stress Screening (ESS) of critical rocket subsystems, ensuring robust performance under extreme conditions.

Validation Under Extreme Conditions

Designed and implemented validation protocols for rocket subsystems under rigorous thermal, vibration, and vacuum conditions.

Streamlined Testing Process

Developed software that significantly streamlined the testing process, reducing manual effort and increasing throughput for subsystem qualification.

Accelerated Flight Readiness

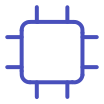
Accelerated the qualification timeline for critical components, ensuring rapid and rigorous preparation for mission deployment.

Impact

This automated testing framework drastically improved the efficiency and reliability of subsystem qualification, ensuring rapid and rigorous preparation for mission deployment and enhancing overall program timelines.

Key Skills Demonstrated

My projects at tiSpace have provided extensive experience across a range of critical engineering disciplines, from low-level embedded programming to robust system automation.



Embedded Firmware

Proficiency in developing real-time firmware for STM32 microcontrollers using libopencm3 and FreeRTOS.

Sensor Integration

Expertise in integrating and developing drivers for various sensors, including pressure, acceleration, and temperature.

Real-time Communication

Strong background in implementing robust communication protocols such as CAN bus, UART, and DMA for high-speed data transfer.

Software Development

Skilled in C++ software development, utilizing modern libraries like Crow and integrating with WebSockets and HTTP for distributed systems.

System Optimization

A proven ability to analyze, optimize, and debug complex embedded and software systems to enhance performance and reliability.

Aerospace Automation

Experience in designing and implementing automated testing and control solutions tailored for critical aerospace applications.

These diverse capabilities ensure a holistic approach to problem-solving, contributing to the development of highly reliable and performant systems.

Conclusion & Q&A

My journey through these projects at tiSpace has honed my skills in developing robust, high-performance systems for critical aerospace applications, covering embedded firmware, sensor integration, real-time communication, and comprehensive system optimization.

Summary of Experience

These projects showcase extensive hands-on experience in building reliable, high-performance embedded systems and software for demanding environments, emphasizing practical problem-solving.

Future Contributions

I am eager to leverage this comprehensive skill set, passion for innovation, and proven track record to tackle new engineering challenges at **URSROBOT**.

Thank you! Any Questions?