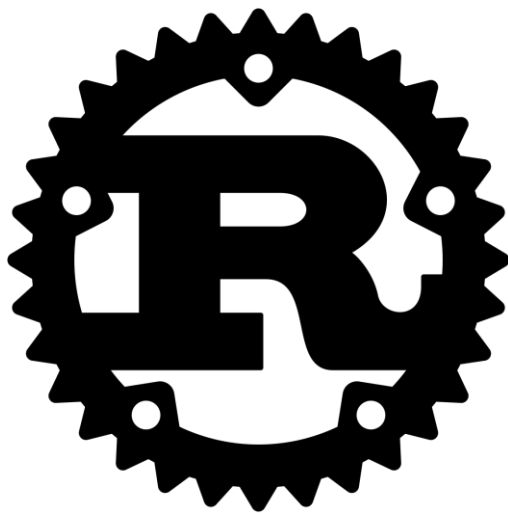


Assessing the risk of using RUST in a memory constrained system as compared to a C/C++ equivalent application

Team Members: Matthew Zhu, Ramiro Mattos, Shubhdeep Kaur, William Lee  
Sponsors: Jake Jansen & Zach Robbins

Background

- C used in embedded systems → memory-unsafe
- Past failures in C: AT&T network crash, the Morris Worm, Apple TLS Bug
- Rust = memory-safe alternative that enforces rules at compile time
- Goal: Evaluate Rust’s safety + performance for embedded use



\*Semester 1\*

Project Requirements

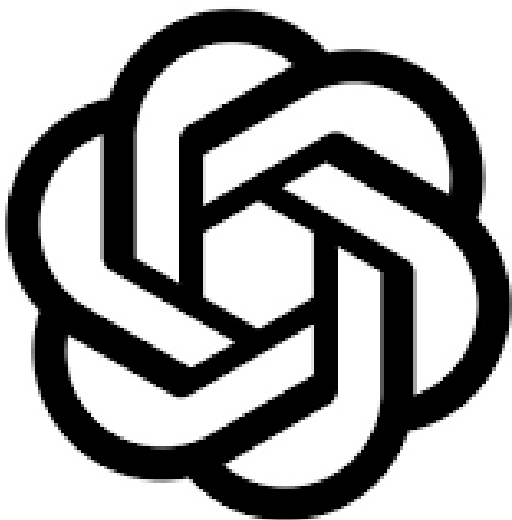
\*Semester 2\*

- Find C → Rust conversion tool
- Analyze & document the accuracy of translations

- Assessment of C (RTOS) vs Rust (RTIC)
- Measure: interrupt latency, memory use, multithreading performance

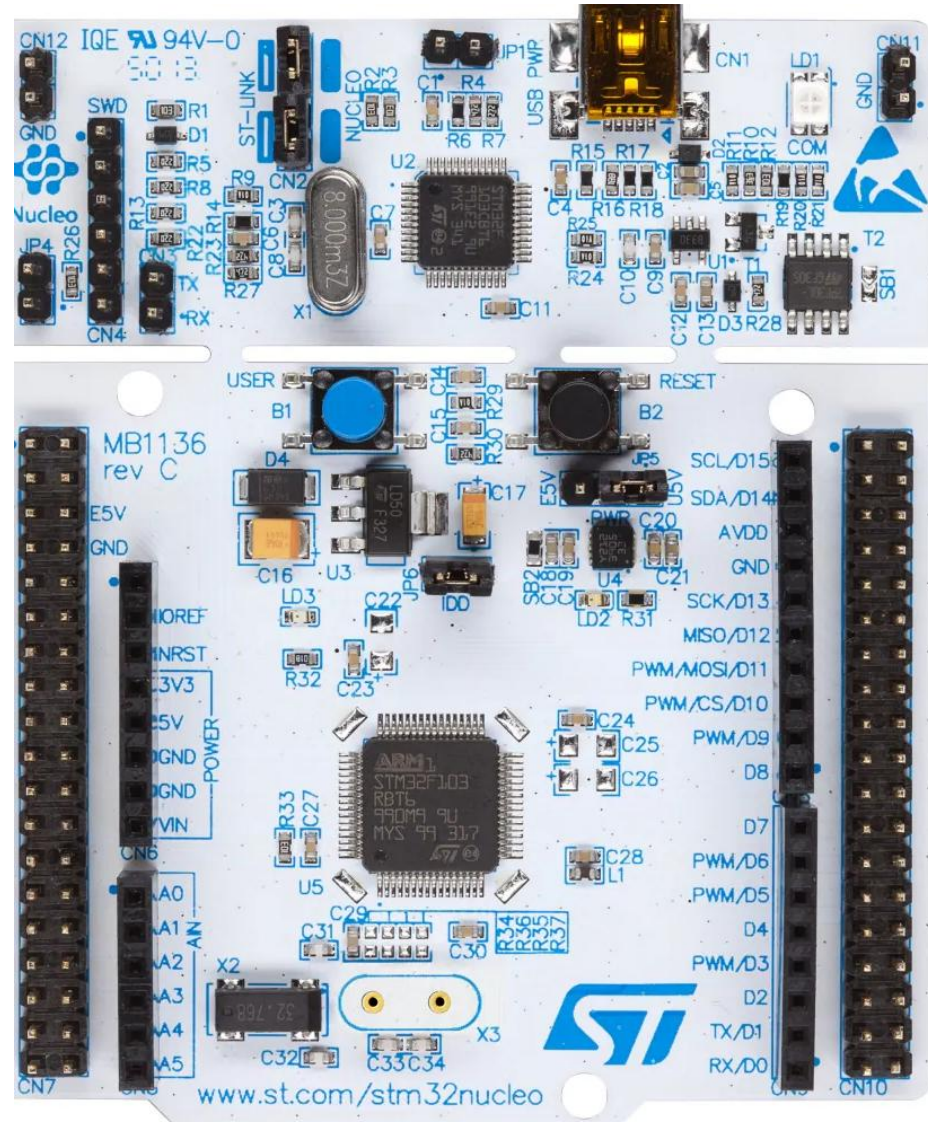
Rust Conversion

- Used **OpenAI ChatGPT** model
- Deterministic programs were not able to accurately convert C to Rust code



ChatGPT

Design Description & Justification



Performance/Memory Testing

- Hardware: **STM32 Nucleo board**
- Created applications in C and Rust with concurrent tasks
- Measured real-time performance, concurrency, and memory

Task 1

LED Blinking

Task 2

ADC Sampling

Task 3

SPI Data Sending

Task 4

UART sending

Outputs of C code and Rust code matched with the chosen translation tool

```
Hello, World!  
Sum: 8  
Factorial of 5 is 120  
Maximum value in array: 8
```

Output in C

```
Hello, World!  
Sum: 8  
Factorial of 5 is 120  
Maximum value in array: 8
```

Output in Rust

Results

Metric	C	Rust
Flash Memory	35.4kB	6.6kB
Static RAM	6.8kB	32 bytes
Interrupt GPIO toggle latency	3µs	0.5µs
Interrupt SPI latency	4µs	13µs
Interrupt low/high priority GPIO toggle latency	2µs/5µs	0.65µs/0.8µs
Multithreading GPIO toggle skew	100ms	14µs
Multithreading SPI/UART skew	3ms	1.4µs

\*Key Takeaway\*

Rust offers improved memory safety and performance over C in concurrent programming

Future Work

- Evaluate more peripherals (I2C, DMA, timers)
- Expanded multithreaded tests
- Investigate long-term maintainability and cost savings