



exida Automotive Symposium 2022

October 13-14, 2022

Daniel Silverstone (Codethink Ltd.)

Jonathan Moore (exida LLC.)

Session 0X: Title: Is Rust ready for safety related applications?



About the Presenter - Jonathan

Over 20 Years of automotive and robotics experience in systems engineering, failure mode avoidance and standards-based software development

Focused on supporting new and existing customers with their implementation of functional safety standards

Bachelor of Electrical and Electronic Systems Engineering from University of York and a Master of Science Electromagnetic Compatibility from University of York

Member of US and International Committee ISO 26262 2 Ed

–Part 6 Software

Sorry I can't be here with you all today.



Director Advanced Systems

jmoore@exida.com

+1 435 754 6280



Daniel Silverstone

Mail: daniel.silverstone@codethink.co.uk

2011 : Codethink Ltd

2006 : Simtec Electronics

2004 : Canonical Ltd

Projects / Special Interest

- Tooling and testing
- Requirements Engineering
- Systems Engineering
- Software Build and Integration
- Developer enablement
- Member of the Rust community

Can you afford to ignore Rust?

How hard will it be to use in Safety projects?

Let's take a look at what Rust offers, and what difficulties you need to be aware of.

What do we do with C today

What resources do we have to help with C?

- exida ctools - <https://github.com/exida/ctools>
- JSF / MISRA C rules - <https://www.stroustrup.com/JSF-AV-rules.pdf>
- MISRA Exemplar <https://github.com/jubnzv/MISRA-Example-Suite>
- AUTOSAR rules
https://www.autosar.org/fileadmin/user_upload/standards/adaptive/17-03/AUTOSAR_RS_CPP14Guidelines.pdf

What is the general suitability of the C language for safety?

- Suitable programming language to solve the problems in the Automotive space
- Coding guidelines
- Language subset
- Static analysis

What else do we do?

- All wrapped by architecture guidelines
- Application specific meta-programming language which generates nominally “good” C

- Brief history
- Status of the language ecosystem
- Rate of change
- Standardization and adoption
- Target (e.g. device) support
- Linux kernel support
- Documentation
- Language specifications
- Contributing to the ecosystem and getting support
- Governance and community

- Core processes
 - Discoverability
 - Documentation
 - Version Control
-
- Package management and build tooling
 - Test frameworks



What makes Rust special

Language

- Panics
- Canaries
- Borrow checker
- Lifetime

Tooling

- Clippy
- Run-time protections
- More helpful messages

What do beginners struggle with

- How does Rust work
- What is it
- Why come to Rust
- Wrapping, c2rust
- Tool support
 - Who has them, licences and safety status

- Compile time canaries are more valuable than runtime canaries, but they come with a cost
- Not everything written will work
- Linting tool says avoid valid rust (called clippy)
- Run-time canaries - control panics
- No amount of compiler and lint help can prevent semantic errors

- The Linux kernel now allows Rust, how does it coexist?
- What is safe, unsafe
- How do they live together
- ***unsafe*** keyword
- FFI and C
- Tooling to assist with all this.

- Built in testing mechanisms
 - Unit testing within codebases
 - Integration testing around codebases
 - Code documentation as tests
- Libraries to assist with further testing
 - Mocking
 - Benchmarking
 - Fuzzing
- Tooling to assist even further
 - MIRI
 - Tarpaulin
- Testing the tooling
 - For tool confidence exida recommend that tool users repeat for their use cases the same validation tests performed by the tool vendor
- ...



C2Rust conversion

- What is this tool
- How to set up
- Can we trust it
- What does it actually do

Repository - <https://github.com/immunant/c2rust>

Demo - <https://c2rust.com/>

- Community-provided tooling
- Licences
- Ferrocene: Rust for Critical Systems
 - <https://ferrous-systems.com/ferrocene/>
- Trusting openly provided tooling

Uncaught

Undefined behaviour

3.3%

Runtime canary

25.0%

Compile-fail

66.3%

[illegible]

Table 1 - Coding	Table 3 - Design Principles	Table 6 - Design	Table 7 - Unit testing
1a clippy	1a gets-out-of-way	1a needs tooling	1a human
1b unnecessary	1b human	1b semi-built-in	1b human
1c built-in	1c human	1c built-in	1c human
1d semi-built-in	1d semi-built-in	1d clippy	1d human
1e human	1e human	1e built-in	1e human
1f irrelevant	1f human	1f built-in	1f needs-tooling
1g semi-built-in	1g human	1g semi-built-in	1g needs-tooling (some done)
1h semi-built-in	1h system property	1h human	1h clippy but also MIRI
1i built-in	1i system property / semi-built-in	1i semi-built-in	1i clippy but also MIRI
		1j 3rd-party-tooling	1j supported human
			1k supported human
			1l supported human
			1m needs-tooling
			1n irrelevant

Is there an argument to use Rust

- What argument can we make for using Rust in Safety Applications?

Is it sufficiently ready for you?

- Only AutoSAR MCAL available for my target or my library is not in Rust and a wrapper not appropriate
 - My target not supported by Rust yet
 - Rust's tier 3 target support is not good enough
 - Particular library and/or wrapper needs
 - What do you get / don't you get
-
- What do you need
 - Will you contribute - be originator of that support
 - Cost of converting library or wrapping Rust - more or less risk than continuing to use C
 - What is your argument for using C?

Programming Rust, Fast Safe Systems Development Blandy, Orendorff & Tindall O'Reilly 2nd Ed June 2021

<https://www.oreilly.com/library/view/programming-rust-2nd/9781492052586/>

<https://ferrous-systems.com/ferrocene/>

<https://www.rust-lang.org/>

exida ctools <https://github.com/exida/ctools>

JSF / MISRA C rules - <https://www.stroustrup.com/JSF-AV-rules.pdf>

MISRA Exemplar <https://github.com/jubnzv/MISRA-Example-Suite>

AUTOSAR rules

https://www.autosar.org/fileadmin/user_upload/standards/adaptive/17-03/AUTOSAR_RS_CPP14Guidelines.pdf

MISRA <https://www.misra.org.uk/>

Software metrics <https://www.exida.com/Blog/software-metrics-iso-26262-iec-61508>

Codethink <https://www.codethink.co.uk/>

Exida www.exida.com <https://www.exida-eu.com/>

Speak with us if you'd like to see any code from the ctools analysis



excellence in dependable automation

Many Thanks for Your Attention

Daniel.Silverstone@codethink.co.uk
JMoore@exida.com