



DesignNews

Writing Microcontroller Drivers in Rust

DAY 1: Introduction to Rust

Sponsored by

DigiKey



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

THE SPEAKER



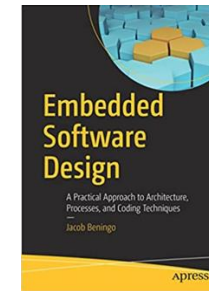
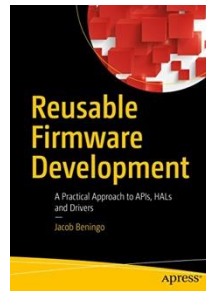
Jacob Beningo

Jacob@beningo.com

Beningo Embedded Group – CEO / Founder

Focus: Embedded Software Consulting and Training

Help teams deliver higher-quality embedded software faster. We specialize in creating and promoting embedded software excellence in businesses around the world.



Blogs for:

- DesignNews.com
- Embedded.com
- EmbeddedRelated.com
- MLRelated.com

Visit www.beningo.com to learn more

01 Embedded Programming Languages

- “You can't allow tradition to get in the way of innovation. There's a need to respect the past, but it's a mistake to revere your past.”

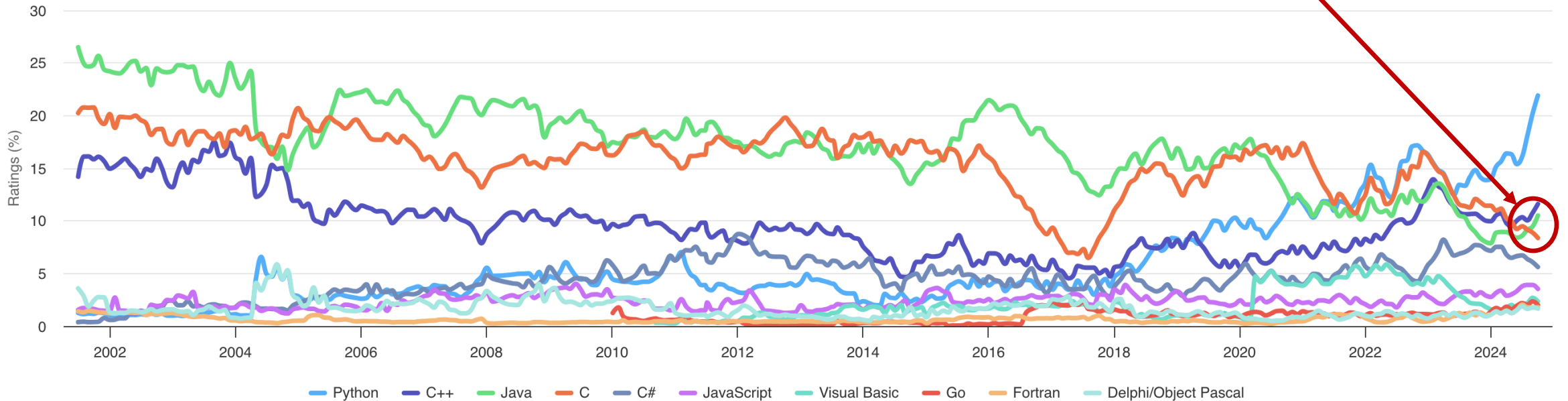
- Bob Iger

Embedded Programming Languages

General Language Popularity

TIOBE Programming Community Index

Source: www.tiobe.com



C++ has surpassed C!

Embedded Programming Languages

Embedded Software Languages

Most Popular Embedded

- C (60 - 70%)
- C++ (20% - 25%)
- Python (<5%)
- Assembly
- Other

Note: 13-14% of Rust Developers are developing bare-metal embedded systems! [Source](#)

Oct 2024	Oct 2023	Change	Programming Language	Ratings	Change
1	1		Python	21.90%	+7.08%
2	3	^	C++	11.60%	+0.93%
3	4	^	Java	10.51%	+1.59%
4	2	v	C	8.38%	-3.70%
5	5		C#	5.62%	-2.09%
6	6		JavaScript	3.54%	+0.64%
7	7		Visual Basic	2.35%	+0.22%
8	11	^	Go	2.02%	+0.65%
9	16	^^	Fortran	1.80%	+0.78%
10	13	^	Delphi/Object Pascal	1.68%	+0.38%
13	20	^^	Rust	1.45%	+0.53%
16	10	v	Assembly language	1.13%	-0.51%

Embedded Programming Languages

What is Rust?

Rust is a modern systems programming language focused on safety, speed, and concurrency.

- Developed by Mozilla Research, with the first stable release in 2015
- Employs a rich type system and ownership model to guarantee memory and thread safety
- Uses a unique system of ownership with rules that the compiler checks at compile time to achieve memory safety without garbage collection
- Advanced features like ownership, types, and borrowing prevent data races at compile time
- Strives for performance equal to C in terms of speed and control over system resources
- Rust's build system and package manager make it easy to manage dependencies through crates and Cargo

Can still leverage existing C code and libraries!

Audience POLL Question

What is your primary development language?

- a) C
- b) C++
- c) Python / MicroPython
- d) Rust
- e) Other

•• Rust in Embedded Systems

02

Rust in Embedded Systems

Why choose Rust for Embedded?

Advantages:

- Memory Safety
- Concurrency Safety
- Zero Overhead Abstractions
- Cross-Platform Development
- Modern Tooling
- Growing ecosystem
- Deterministic resource cleanup
- Compile-time error checking
- Interoperability

Disadvantages:

- Steep learning curve
- Smaller talent pool
- Limited library support for some targets
- Longer compile times
- Evolving language and ecosystem
- Verbose error handling
- Limited support for very low-level dev
- Lack of IDE support similar to C/C++
- Cross compilation complexity

Rust in Embedded Systems

Memory Safety – A C Example

```
#include <stdio.h>
#include <stdint.h>

int main (void){
    printf ("Hello, world!\n");
    printf ("Let's overflow the buffer!\r\n");

    uint32_t array[5] = {0, 0, 0, 0, 0};

    for(int index = 0; index < 6; index++) {
        printf("Index %d: %d\r\n", index, array[index]);
    }

    return 0;
}
```

Buffer Overflows

A buffer overflow is a type of software vulnerability that occurs when more data is written to a buffer in memory than the buffer can store. This can cause the data stored in adjacent memory locations to be overwritten, leading to unintended and potentially harmful consequences.

Buffer overflows can cause crashes, corrupt data, or even allow attackers to execute arbitrary code on the system.

Rust in Embedded Systems

Memory Safety - A Rust Example

```
fn main() {  
    println!("Hello World!");  
    println!("Let's overflow the buffer!");  
  
    let array: [u32; 5] = [0;5];  
  
    for index in 0..array.len() + 1 {  
        println!("Index {}: {} ", index, array[index])  
    }  
}
```

Buffer Overflow Detection

- **Bounds Checking:** Data can never be written outside a buffer's boundaries
- **Ownership and Borrowing:** Prevents use-after-free and double-free bugs.
- **Safe Rust:** Language subset that makes it easier to write memory safe code.
- **Mutability:** by default, variables are not mutable and must be declared so.

Audience POLL Question

What timeframe are you looking to use Rust for embedded?

- a) Now
- b) 1 – 3 months
- c) 3 – 6 months
- d) 6 – 12 months
- e) 12+ months

•• The Rust Toolchain

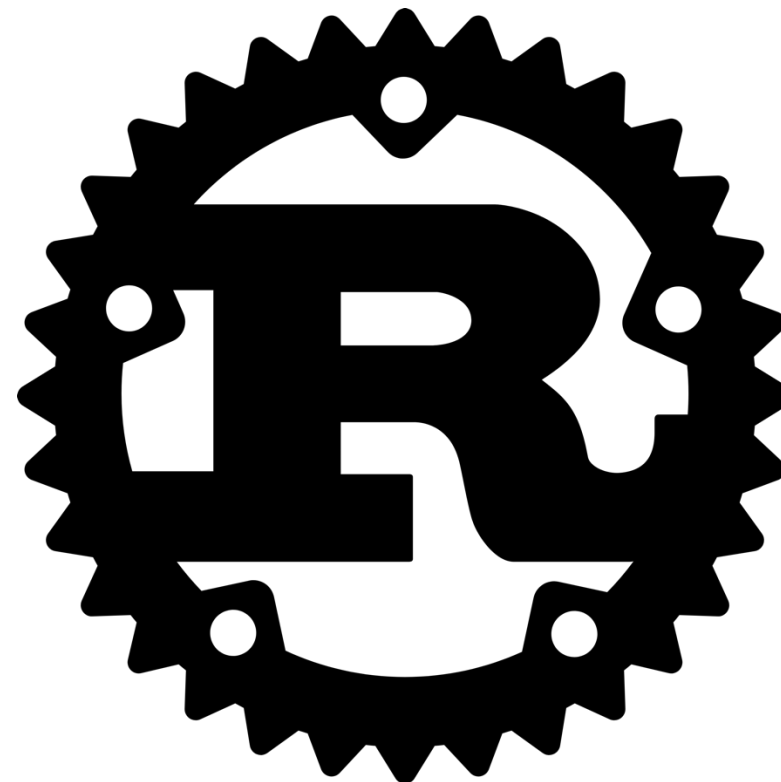
03

The Rust toolchain

What is needed to build Rust applications?

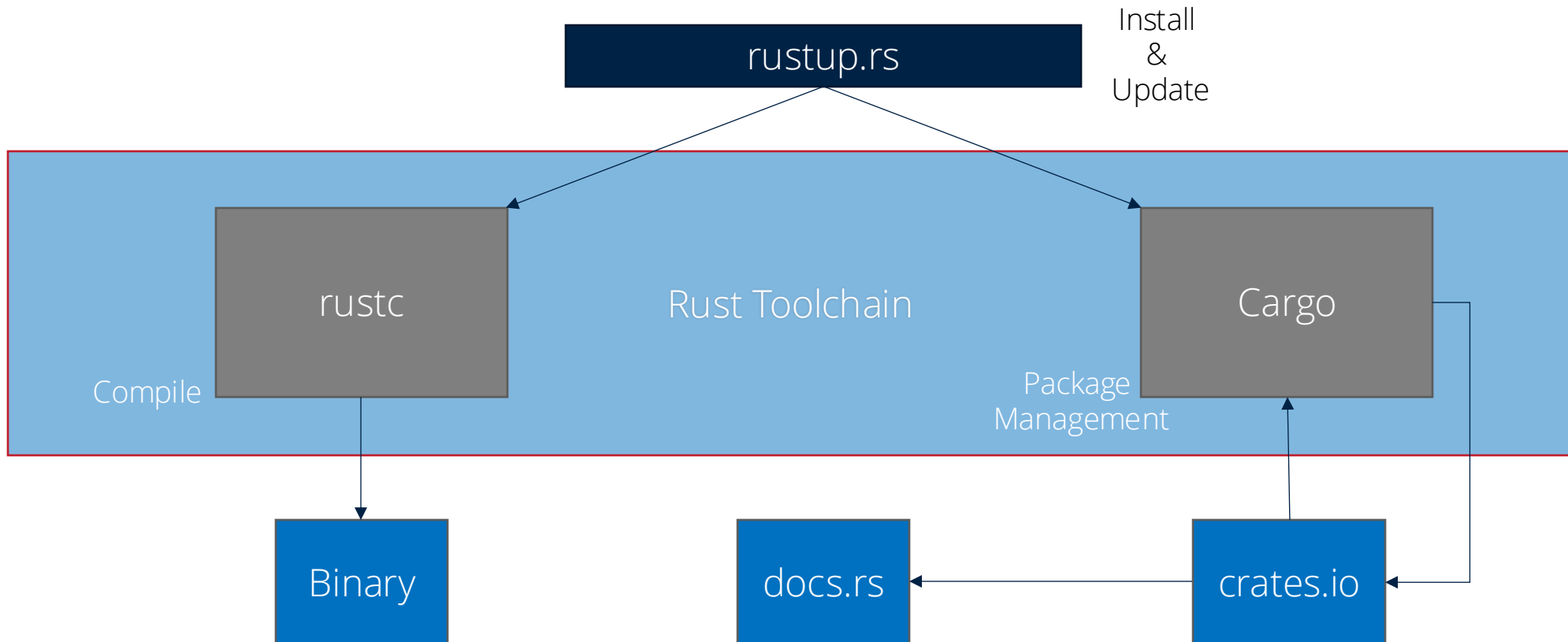
- **Rustup** is a command-line tool for managing Rust versions and associated tools. It simplifies installing and managing different versions of Rust and associated toolchains.
- **Cargo** is the Rust package manager. It handles dependencies, compiles packages, runs tests and generates documentation.
- **Rustc** is the Rust compiler, which converts Rust code into executable binaries.

<https://www.rust-lang.org/learn/get-started>



The Rust toolchain

Rust Toolchain Overview



The Rust toolchain

Rustup

Visit <https://rustup.rs/>

- Follow install instructions

```
beningo@Jacobs-MacBook-Pro ~ % rustc -V
rustc 1.65.0 (897e37553 2022-11-02)
beningo@Jacobs-MacBook-Pro ~ %
```

To install Rust, if you are running Unix,
run the following in your terminal, then follow the onscreen
instructions.

```
$ curl --proto 'https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```



If you are running Windows 64-bit,
download and run
rustup-init.exe
then follow the onscreen instructions.

If you are running Windows 32-bit,
download and run
rustup-init.exe
then follow the onscreen instructions.

Audience POLL Question

How much experience do you have with Rust?

- a) None
- b) beginner
- c) intermediate
- d) Expert

•• Next Steps

04

Embedded Rust Docker Container

- https://mailchi.mp/beningo/embedded_rust_docker_container
- Rust Toolchain
- Embedded Tools

Beningo Rust Docker Container



Additional Resources

Please consider the resources below:

- [Jacob's Blogs](#)
- [Jacob's CEC courses](#)
- [Embedded Software Academy](#)
- Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>

www.beningo.com



Consulting

Coaching

Training



DesignNews

Thank You

Sponsored by

DigiKey

