Embedded Rust Workshop (Installation)

Prof. Bart Massey and Rustaceans Africa

Introduction to Rust

What is Rust

Rust is a systems programming language that combines high performance with memory safety. It uses ownership and borrowing rules to prevent bugs without a garbage collector. Designed for reliability and concurrency, it’s used in everything from embedded devices to web services.

Why Rust

Features of Rust

- **Memory Safety** – Prevents crashes and bugs by managing memory without a garbage collector.
- **Ownership System** – Controls how data is shared and used through strict rules.
- **Borrowing & References** – Lets you safely access data without copying it.
- **Zero-Cost Abstractions** – High-level features that don’t slow down performance.
- **Concurrency** – Makes writing safe multi-threaded code easier.
- **Pattern Matching** – Provides powerful ways to handle different cases in code.
- **Error Handling** – Uses Result and Option instead of exceptions for safer code.
- **Traits** – Define shared behavior that different types can implement.
- **Generics** – Write flexible code that works with many data types.
- **Strong Type System** – Catches mistakes at compile time instead of at runtime.
- **Cross-Platform** – Runs on many systems, from embedded devices to servers.
- **Cargo & Crates.io** – Built-in package manager and ecosystem for easy project management.

Installing Rust

To install rust, we use the following curl command (For Linux users and Mac users)

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

For Windows users go to the below link to see how to install Rust for Windows.

```
https://forge.rust-lang.org/infra/other-installation-methods.html
```

The Rust tools are installed in the ~/.cargo/bin directory. Make sure you add that to the PATH.

Then check by running

```
rustup --version
```

Once rust has been installed, with rustup cargo will also be installed.

Check it using

```
cargo --version
```