Writing Rust Code for Arduino UNO: A Comprehensive Tutorial

Unlocking the Power of Rust on Arduino UNO

Introduction

Rust is a modern programming language that emphasizes safety and performance. By using Rust for your Arduino UNO projects, you can leverage its powerful features to write efficient and reliable code. This tutorial will guide you through the process of writing Rust code for the Arduino UNO, from setting up your environment to deploying your first Rust-based sketch.

Prerequisites

Before we dive into the tutorial, make sure you have the following:

- Arduino UNO board
- USB cable to connect the Arduino UNO to your computer
- A computer running Windows, macOS, or Linux
- Basic knowledge of programming
- Rust programming language installed (you can get it from rust-lang.org)

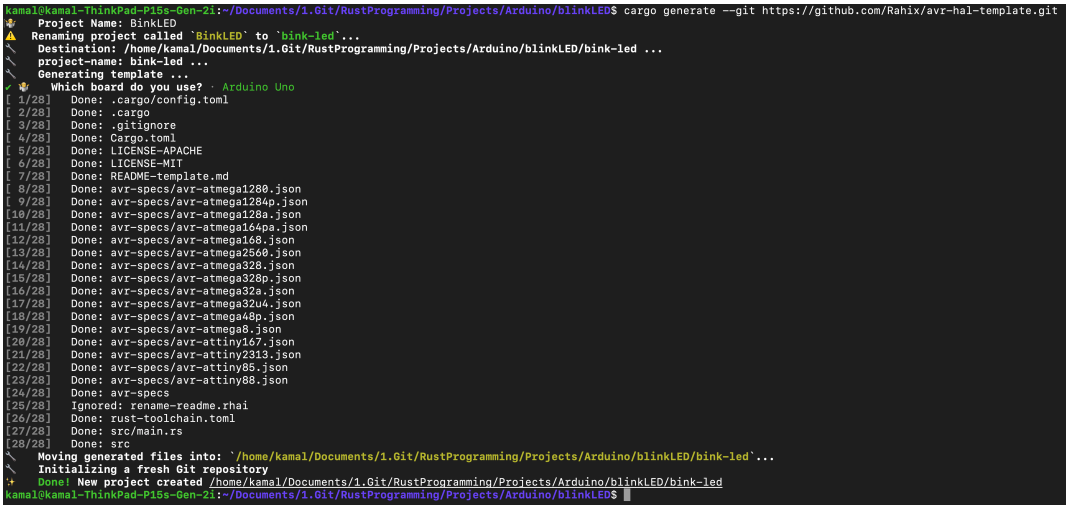
Setting Up Your Environment

1. Install Cargo Generate

```
cargo install cargo-generate --lockedcargo generate --version
```

2. Create a New Arduino Uno Project

```
cargo generate --git https://github.com/Rahix/avr-hal-template.git
```



3. Build the Project

```
cargo build --release
```

```
   Compiling nb v1.1.0
   Compiling vcell v0.1.3
   Compiling cfg-if v1.0.0
   Compiling bare-metal v1.0.0
   Compiling embedded-hal v1.0.0
   Compiling critical-section v1.2.0
   Compiling void v1.0.2
   Compiling ufmt-write v0.1.0
   Compiling avr-device v0.7.0
   Compiling nb v0.1.3
   Compiling unwrap-infallible v0.1.5
   Compiling embedded-storage v0.2.0
   Compiling panic-halt v0.2.0
   Compiling embedded-hal v0.2.7
   Compiling ufmt v0.2.0
   Compiling avr-hal-generic v0.1.0 (https://github.com/rahix/avr-hal?rev=3c089795cadbbc7fa83f45958128689fee7ba1e4#3c089795)
   Compiling atmega-hal v0.1.0 (https://github.com/rahix/avr-hal?rev=3c089795cadbbc7fa83f45958128689fee7ba1e4#3c089795)
   Compiling arduino-hal v0.1.0 (https://github.com/rahix/avr-hal?rev=3c089795cadbbc7fa83f45958128689fee7ba1e4#3c089795)
   Compiling bink-led v0.1.0 (/home/kamal/Documents/1.Git/RustProgramming/Projects/Arduino/blinkLED/bink-led)
WARN rustc_codegen_ssa::back::link Linker does not support -no-pie command line option. Retrying without.
    Finished `dev` profile [optimized + debuginfo] target(s) in 26.44s
kamal@kamal-ThinkPad-P15s-Gen-2i:~/Documents/1.Git/RustProgramming/Projects/Arduino/blinkLED/bink-led$ cargo build
```

4. Find the .elf File

```
target/avr-atmega328p/release/
```

5. Convert to HEX Format

```
avr-objcopy -O ihex target/avr-atmega328p/release/blink-led.elf blink.hex
```

6. Upload to Arduino Uno

```
avrdude -c arduino -p atmega328p -P /dev/ttyACM0 -b 115200 -U flash:w:blink.hex:i
```