Table of Contents

Repository description

I recommend read PDF version of this readme with better formatting.

Repository contains implementation of classical Arduino examples in the Rust programming language. It uses avr-hal low-level API and and some examples work directly with MCU registers. Comments often contain references to ATMega328p Datasheet. Examples cover themes:

- pins - low, high, pull-up, ADC
- different modes of timers
- different modes of PWM
- SPI
- UART (write to PC)
- interrupts (timer, external)

Source of examples

This repository based on tutorials from Amperka wiki (rus). Amperka is Russian's store which sells Arduino kits and extensions.

But there is not full implementation of all examples and I changed some of them. I suppose that some examples is too simple or don't give new experience compare to previous.

How to run on a board

1. Make circuit using scheme `simulide/project_name.png`.

2. Plug your Arduino UNO into your PC
3. Set enviroment variable with tty of your board. In my setup is usually `/dev/ttyUSB0`:

   bash

   ```
   RAVEDUDE_PORT=/dev/ttyUSB0
   ```

   fish

   ```
   set -x RAVEDUDE_PORT /dev/ttyUSB0
   ```

4. Run cargo `cargo run --bin <EXAMLPLE_NAME>`, for example: `cargo run --bin 01_blink`. Command will:
   a. build example
   b. make HEX file (you can load it into SimulIDE)
   c. write an example to Arduino

fish shell autocompletes examples names.

`cargo run` also generates HEX files, which can be loaded into SimulIDE.

I have tested this script only on Linux (Archlinux). It should work on Windows, but you should modify `.cargo/config.toml` (see the instruction in the file). Get attention, after this you should be generate HEX files manually.

SimulIDE projects

SimulIDE is a simple real time electronic circuit simulator, intended for hobbyist or students to learn and experiment with simple electronic circuits and microcontrollers, supporting PIC, AVR and **Arduino**.

Each example has corresponding SimulIDE project (`.sim1`) in the folder `simulide`. Due to this, you can try them without any hardware!

You can download SimulIDE for free, if set price to $0. But I recommend you to make a donation to the author (yes, I did it), because SimulIDE is a really cool project for electronics hobbyist.

How to run a project in SimulIDE:

1. Open project in **SimulIDE-1.0-RC3** and higher.
2. Right click on Arduino
3. Click "Load firmware" and select <buildname.hex>, which you can find in the folder `target/avr-atmega328p/debug`. HEX file is generated by `cargo -run` (see

Some projects has simulated in slower mode. I did it for more clarity.

List of examples

Example	Description	AVR (or Rust) techniques	Arduino functions
01_blink	Blinking led - Hello world in the Arduino.	• Toggle led • delay milliseconds	`delay` `digitalWrite`
02_blink_fade	Led with different brightness	Fast PWM mode	`analogWrite`
03_pot_light	LED with controlled by pot brightness	ADC, using pot	`analogRead`
04_buzzer	Buzzer plays musical notes	Timer: • compare match mode • toggle pin d9 by timer	`tone`
05_night_light	LED on/off controlled by pot and photoconductive cell	ADC, using pot and photoconductive	`analogRead`
06_pulsar_bar	Smoothly change brighntess of LED bar	Fast-Mode PWM with a deep description.	`analogWrite` `millis`
07_running_bar	Sequentially on/off leds in a bar	• AVR-Rust specific - use pins in array. • Working with UART	`digitalWrite`

09_mixer	Changes speed of motor by buttons	pull-up pins	`pinMode PULLUP`
			`digitalRead`
			`digitalWrite`

10_led_toggle	On/off led by button	Nothing new compare to 09_mixer	
11_inc_dec_light	Change brightness of led with 2 buttons	External Interrupts (INT0, INT1) Issue with Fast PWM	`attachInterrupt`
13_seven_segment_counter	Change digit from 0 to 9 per seconds on 7 segments counter.	struct as indicator's model	
14_shift_register.rs	Like 13, but uses 8-bit serial to parallel shift register 74HC595	SPI	`shiftOut`
15_display	Work with LCD 16 symbols 2 row display (HD44780). Example shows an implementation of all commands from datasheet.	Modeling device with `struct` and `impl` (like OOP)	Implements library `LiquidCrystal`