LULEÅ
UNIVERSITY
OF TECHNOLOGY

STM32 Nucleo-F411RE
Board Support Crate for Embedded Rust

D7018E - Special Studies in Embedded Systems

Johannes Sjölund
johsjl-1@student.ltu.se

January 16, 2018

LULEÅ UNIVERSITY OF TECHNOLOGY

STM32 Nucleo-F411RE

- Development board for the Arm Cortex-M4 microcontroller STM32F411RE by STMicroelectronics.
- Supports Arduino header expansion boards, as well as STM32 Nucleo Morpho headers.
- Power, buttons, LEDs, programmer IC, USB-to-Serial converter.

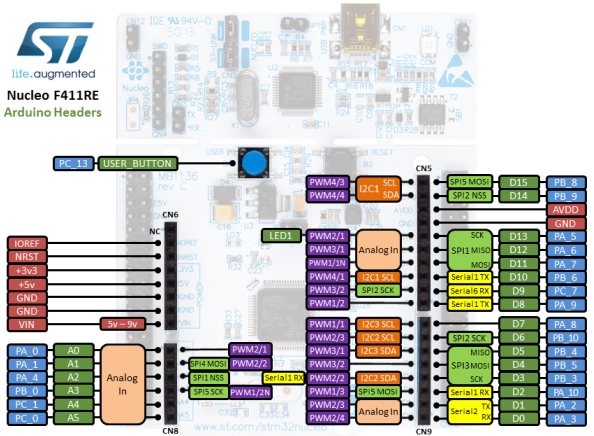


Figure: Map of hardware peripheral resources on the Nucleo-F411RE.

[1] http://www.st.com/en/evaluation-tools/nucleo-f411re.html

LULEÅ UNIVERSITY OF TECHNOLOGY

Real-time systems programming

Our programs must:

1. Respond to many different external events, all in a timely fashion and never miss anything.
   - Button clicks
   - Communication
   - The passing of time
2. Be correct and never enter an undefined state.
3. Not waste power.

Problem:
How does a real-time system read and write mutable resources when different processes can access them at any time?

- **Data race** conditions can be solved by mutual exclusion on critical sections.
- Mutual exclusion can lead to **deadlock** if system uses resource holding, circular wait and no preemption.

LULEÅ UNIVERSITY OF TECHNOLOGY

RTFM

Real-Time For the Masses (RTFM)

Programming model for robust real-time systems. [2]

- Based on a reactive programming model involving **tasks** and **resources**.
- Resource management provided by the **Stack Resource Policy** (SRP).
- Uses the underlying interrupt hardware for static priority scheduling of tasks (the BASEPRI register).
- Guarantees **deadlock-lock free** execution.

[2] P Lindgren, M Lindner, A Lindner. **RTFM-core: Language and Implementation.** LTU, 2015

LULEÅ UNIVERSITY OF TECHNOLOGY

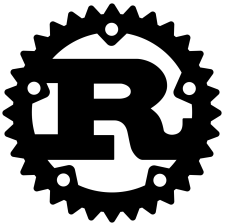
Rust

What is special about it?

Designed for highly concurrent and highly safe systems, in the context of memory safety.

- Does not permit **dangling pointers**, or **data races** in safe code.
- Unsafe code can still be used, but must be tagged as `unsafe{}`.

Does not perform automated **garbage collection** like Go, Java or .NET, where resources like heap memory used by unused variables are freed periodically. Rust frees them when variables gets out of scope according to its **ownership** and **borrowing** system.



[3]

[3] `https://www.rust-lang.org/en-US/`

LULEÅ UNIVERSITY OF TECHNOLOGY

cortex-m-rtfm

Rust crate for RTFM on Arm Cortex-M

Rust crate (code library) which combines the best of Rust and RTFM, written by Jorge Aparicio and Per Lindgren.

Designed around a concept of **tasks** and **resources**, where tasks

- are **event triggered**,
- are assigned **constant priorities**,
- must not contain endless loops. [4]

[4] `https://docs.rs/cortex-m-rtfm/0.3.1/cortex_m_rtfm/`

LULEÅ UNIVERSITY OF TECHNOLOGY

Programming embedded Rust

The **cortex-m-rtfm** crate is great, but how to make use of it for a specific microcontroller?

Manufacturers like STMicroelectronics and NXP Semiconductors provide **SVD files** (XML) which describes the hardware features, memory layout, and register functions of the device.

The SVD file can be used by **svd2rust**[5] to generate Rust code for peripheral access on the microcontroller.

`https://gitlab.henriktjader.com/pln/STM32F40x`

[5] `https://docs.rs/svd2rust/0.12.0/svd2rust/`

LULEÅ UNIVERSITY OF TECHNOLOGY

Board support crates

We can use the code from **svd2rust** to access the microcontroller registers, but how do we know what to do with them? What is `stm32f40x::dma2::s0ndtr::ndt`?

Read the **datasheet** [6] and **reference manual** [7].

To speed up development, a board support crate can provide a hardware abstraction layer, and examples of how to use the devices.

- STM32F3DISCOVERY: `https://github.com/japaric/f3`
- Blue-pill: `https://github.com/japaric/blue-pill`
- **Nucleo-F411RE**: `https://github.com/jsjolund/f4`

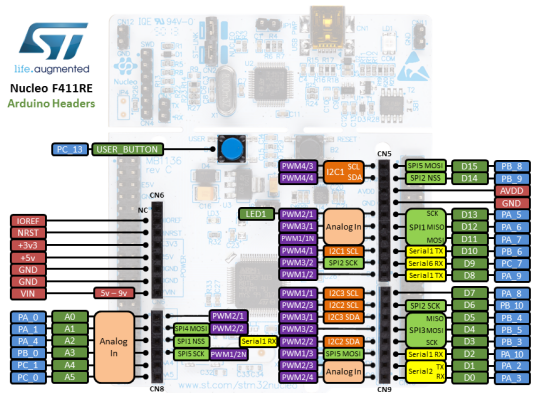
[6] `http://www.st.com/resource/en/datasheet/stm32f411re.pdf`
[7] `http://www.st.com/resource/en/reference_manual/dm00119316.pdf`

LULEÅ UNIVERSITY OF TECHNOLOGY

STM32 Nucleo-F411RE Board Support Crate

Provides an abstraction layer for STM32F40x

- GPIO
- ADC
- Communication
  - I^{2}C, SPI
  - UART (serial) over USB using ST-Link-v2
- Timers
  - Microsecond counter
  - PWM generation
  - Input capture
- DMA
- Clocking (16-100 MHz)



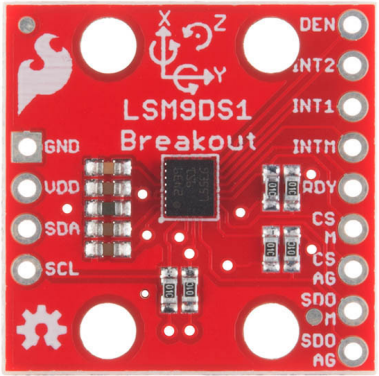
LULEÅ UNIVERSITY OF TECHNOLOGY

Usage /examples/imu.rs

The inertial measurement unit LSM9DS1 can communicate through SPI and report

- Acceleration (accelerometer)
- Rotational velocity (gyro)
- Magnetic field strength (magnetometer)

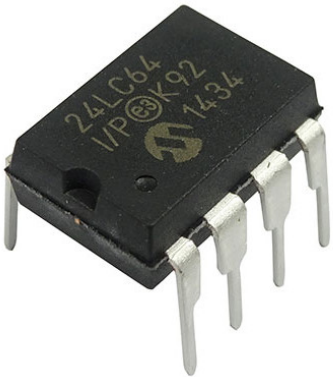
Sensor fusion algorithms such as **Madgwick AHRS** can take these measurements and calculate current attitude/orientation relative to Earth's horizontal plane.



[8] http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

Usage /examples/eeprom.rs

To store persistent data such as user settings, we may want to use an external EEPROM. The Microchip 24LC64 can communicate through the I^2C protocol for reading and writing.



9

[9] http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

LULEÅ UNIVERSITY OF TECHNOLOGY

Other examples

- /examples/adc1.rs - Reading ADC using a timer and DMA.
- /examples/usart2-dma.rs - Serial communication over USB using DMA.
- /examples/button.rs - Toggling the LED when user interrupts with button.
- /examples/pwm1.rs - Output PWM signal using timer.
- /examples/capture4.rs - Measure PWM signal frequency using timer.

LULEÅ UNIVERSITY OF TECHNOLOGY

Limitations/Future work

- Different peripheral **pin mappings** are available but not supported.
- Should be converted to the **singletons** framework.
- Add features like
  - Timer encoder support,
  - Injected mode ADC,
  - Real-time clock,
  - USB OTG,
  - Secure digital card support,
  - More DMA.
- Lots and lots of **usability** improvements.

LULEÅ UNIVERSITY OF TECHNOLOGY

Questions?

Bug reports and pull requests are always welcome.