

RUST PROGRAMMING LANGUAGE AT BLUEWIND

Known for high performance and memory safety, Rust offers compilation efficiency like C. Ideal for reliable systems in automotive and industrial sectors and a promising modern language with built-in safety features.

RUST AT BLUEWIND

- Bluewind is part of the AURIX™ Rust Startup Ecosystem and is among the first to provide low-level Rust drivers for Infineon's AURIX™ Microcontroller.
- Our deep expertise in embedded programming, makes us well-equipped to leverage Rust's capabilities for future projects.
- We seamlessly integrate Rust with existing code for cost-effective and reliable embedded systems.
- We actively seek collaborations with universities and corporations to advance research initiatives.
- We employ Rust for crafting safety drivers in the Bluewind iMCAL Software Framework product.

```
node.setup_pins(Some {
    tx: PIN_TX_0_0_P20_8,
    rx: PIN_RX_0_0_P20_7,
});

node.setup_interrupt(&NodeInterruptConfig {
    interrupt_group: InterruptGroup::Rxf0n,
    interrupt: Interrupt::RxFifo0newMessage,
    line: InterruptLine::Line1,
    priority: Priority::try_from(2).unwrap(),
    tos: Tos::Cpu0,
});

Some(node.lock_configuration())
}

/// Initialize the STB pin for the CAN transceiver.
fn init_can_stb_pin() {
    let gpio20 = pac::P20.split();
    let mut stb = gpio20.p20_6.into_push_pull_output();
    stb.set_low();
}

#[export_name = "main"]
fn main() -> ! {
    info!("Start example: can_send");
    info!("Enable interrupts");
    unsafe {
        asm!("enable");
    }
}
```

WHY USING RUST AS A PROGRAMMING LANGUAGE?

- Excels in low-level hardware interaction, providing superior safety over C/C++.
- Strong support for interoperability with other languages in embedded systems.
- Simplifies deployment and maintenance of embedded code, even in functional safety applications.
- Modern software development environment with advanced tools enhancing team productivity.

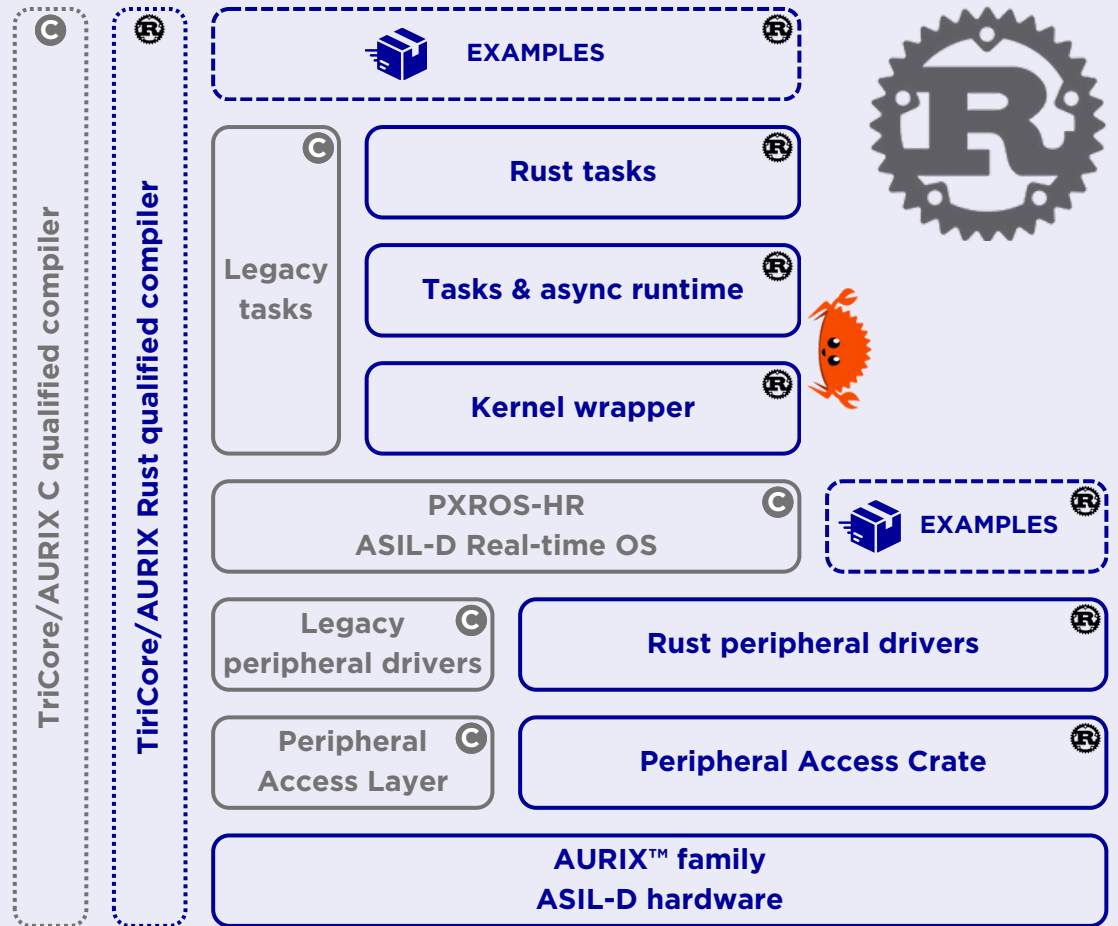


AURIX™ RUST STARTUP ECOSYSTEM

Bluewind offers low-level peripheral drivers for Infineon AURIX™ native in Rust, leveraging the Infineon Peripheral Access Crate (PAC) designed for AURIX™ TC375 peripherals and CSFRs.

Veeclle and HighTec provide a Rust asynchronous runtime and wrapper for HighTec's PXROS-HR multicore real-time OS, facilitating the coexistence of drivers, peripherals and algorithms in both existing "C" implementations and new Rust developments.

Application examples, curated by Veeclle and Bluewind, are also included in the AURIX™ Rust Startup Ecosystem, encompassing bare metal driver examples, driver instances employing PXROS-HR, and connectivity application demonstrations.



WANT TO GIVE IT A SHOT?

Bluewind Code Examples

QR Code leading to a GitHub page to download code examples for Bluewind drivers



Ecosystem Documentation

QR Code leading to a GitHub page to download code examples for the whole ecosystem.

