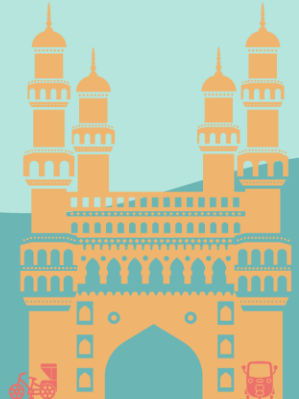
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
INDIA

GITAM
DEEMED TO BE UNIVERSITY

RUST Meets Embedded Linux
- Build Memory Safe and Reliable Kernel Modules & Applications

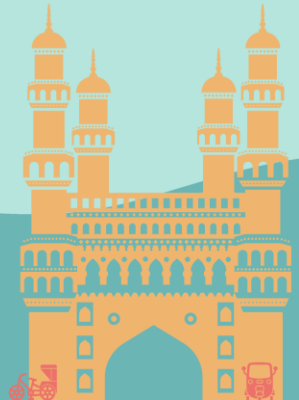
Rajesh Sola,
Centre for Academic Innovation and Advancement (CAIA),
GITAM (Deemed to be) University

5 August,
Hyderabad, India
#OSSummit



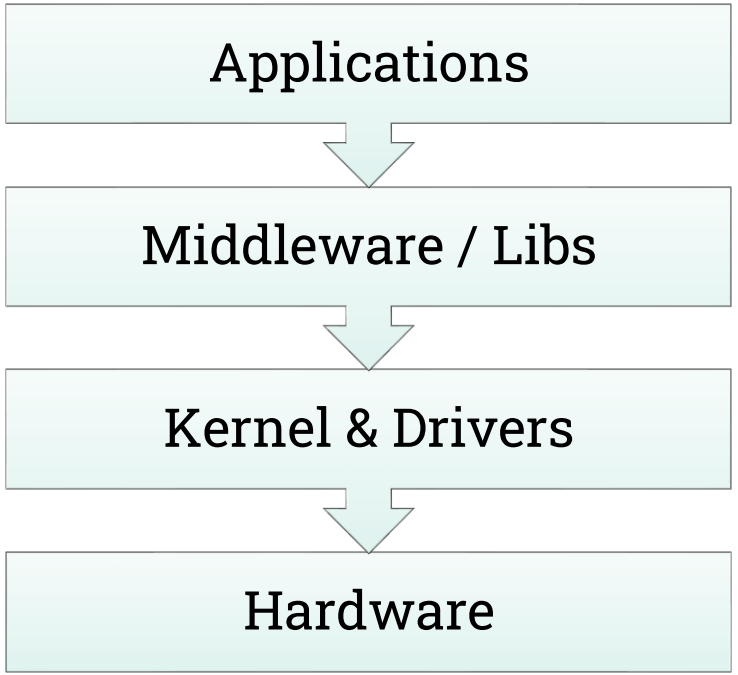
Outline | RUST Support in Linux

- RUST in Kernel space – Modules & Drivers
- RUST in User space – Applications & Libraries
- Cross compiling the code
- Yocto Recipes for RUST components



Embedded Linux - Overview

- Bootloader
- Kernel – Modules, Drivers ★
- Libraries ★
- Applications ★



Need for RUST & Key Features

No undefined behavior	No Standard Library
Memory Safety without Garbage	Unsafe Blocks
Low level access (Native, ELF)	Traits and Generics for Abstraction
Abstractions	Pattern Matching & Enums for State Machines
Toolchain Support	Concurrency Primitives
Unified Language	Error Handling
Safe Concurrency	FFI (Foreign Function Interface)
Rich library – std library support	Inline Assembly Support
Interoperability with C	Zero-cost Abstractions

Note:- Safety doesn't mean safety-critical

GITAM
DEEMED TO BE UNIVERSITY

RUST in Kernel space

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
INDIA

RUST in Kernel Space

- Build kernel with RUST Support
- Important modules
- Writing simple Module
- Misc Char Device
- RUST Bindings
- Documentation

- RUST support in upstream kernel
  `kernel.org, github.com/torvalds/linux`
- RUST support in custom kernel maintained by Rust for Linux project
  `https://github.com/Rust-for-Linux/linux`
- Refer any LXR like
  https://elixir.bootlin.com/
- Format your code, if any in-tree module is added , e.g. in samples/rust
  `make LLVM=1 rustfmt`

Please refer https://github.com/rajeshsola/rust-linux-examples for code snippets

Important Modules (under Kernel Trait)

- kernel::module
- kernel::chrdev
- kernel::file_operations
- kernel::fs
- kernel::thread
- kernel::sync
- kernel::task
- kernel::timer
- Kernel::alloc
- kernel::print
- kernel::c_str

- RUST support in Linux Kernel : Current Status & Gap Analysis
- Refer sub folder rust/kernel in KSRC for accurate list of supported bindings
- Build docs for custom kernel source
  `make LLVM=1 rustdoc`
- RUST documentation of mainline, Rust for Linux available at

  https://rust.docs.kernel.org/kernel/
  https://rust-for-linux.github.io/docs/kernel/

GITAM
DEEMED TO BE UNIVERSITY

RUST in Userspace

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
INDIA

RUST in User Space

- Simple Programs
- Static Library
- Dynamic Library
- Interoperability with C, FFI
- Cross Compilation

Cross Compilation

- Installing additional targets
- Building code for desired target

- ❑ Name Mangling Issues
- ❑ FFI : Foreign Function Interface
- Calling C functions from RUST
- Calling RUST functions from C
- Callback handling

Embedded HAL

Various crates available with HAL implementation available

- Architecture specific, e.g. Cortex-M
- Board specific, e.g. Rpi
- Peripheral specific, e.g. FTDI

> *https://github.com/rust-embedded/*

> *https://github.com/rust-embedded/awesome-embedded-rust*

- linux-embedded-hal
- embedded-io
- embedde-can
- embedded_hal_async

System Programming

GITAM
DEEMED TO BE UNIVERSITY

Multithreading	• std::{thread, future}, async fn, async_std::task
Concurrency	• std::sync:: {Arc, Mutex, Condvar, Barrier}
IPC	• std::sync::{mpsc, mpmc}
Networking,	• std::net ::{TcpListener, TcpStream, UdpSocket}

Debugging

- By logs/print messages, log levels
- Debug trait , #[derive(Debug)]
- rust-analyzer
- rust-gdb, rust-lldb

GITAM
DEEMED TO BE UNIVERSITY

Yocto Recipes

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
INDIA

Yocto Recipes

For Kernel Modules

- module class (inherit module)

For Application & Libraries

- cargo class (inherit cargo)

meta-rust layer

Generate recipe from cargo (bitbake cargo)

THANK YOU

Queries?