Demo: Rust with Yocto Project®

Paul Barker, Konsulko Group

Yocto Project Virtual Summit Europe, October 29-30, 2020
About Me

- Involved in Yocto Project since 2013
- Work across the whole embedded stack
- Principal Engineer @ Konsulko Group
- https://www.konsulko.com/
Contact Details

- Email: pbarker@konsulko.com
- Web: https://pbarker.dev/
- Twitter: pbarker_dev
About This Talk

- Brief Introduction
  - Mostly covered by Randy already
- License Compliance with Rust & Yocto Project
- Installing Rust
- Demo
License compliance with Rust & Yocto Project (1)

- Like many newer languages Rust has its own package manager / build tool called Cargo
- These tools often present issues for Embedded development and license compliance
  - These just don’t seem to be first class concerns
License compliance with Rust & Yocto Project (2)

- Features we need from language package managers:
  - Offline build support
  - Download source archive
  - Including license text & other collateral
  - HTTP/HTTPS proxy support
  - Source mirror support
License compliance with Rust & Yocto Project (3)

- Cargo actually integrates quite well
- All features in the previous slide are supported
- However, licenses & license text are not collected for dependency crates
Installing Rust (1)

- Installing Rust natively is not required to build with Yocto Project
- However, it is required to run cargo-bitbake
- Being able to build natively can help with debugging
Installing Rust (2)

- See [https://rustup.rs/](https://rustup.rs/)

- Run the following:
  - curl https://sh.rustup.rs -sSf | sh
  - source $HOME/.cargo/env

- For cargo-bitbake:
  - cargo install cargo-bitbake
Demo Tricks

- Pre-populated sstate cache
- Local mirrors of poky, meta-openembedded & meta-rust
- Using podman to run builds on a stable distro
  - Insert Arch Linux evangelism here
Thanks for your time