

Úvod

IoT zariadenia sa pripájajú na internet, čím sú viac vystavené bezpečnostným hrozbám ako kedykoľvek predtým. Preto embedded zariadenia ďalšej generácie majú potenciál implementovať memory safe jazyk Rust.

Spôsobý komunikácie systému

- MQTTv5 a TLS: Zabezpečená komunikácia s brokerom.
- CAN bus: Komunikácia v priemyselných aplikáciách pre zariadenia bez konektivity na internet.
- ESP-NOW: Komunikácia medzi ESP zariadeniami.
- OTA(Over-the-Air): vzdialená aktualizáciu firmvéru.

Hardvér

- Dual-core procesor s frekvenciou 240 MHz.
- CAN a integrované WiFi a Bluetooth
- Lokálny display
- Flexibilné napájanie USB-C a low-power módy

Vlastný PCB dizajn

- Kompatibilita s legacy systémom.
- Optimalizácia pre vlastné riešenie a bezpečnosť.
- Nákladovo efektívne a ľahko škálovateľné riešenie.
- Možnosť integrácie špecifických funkcií a externých periférií.

Záver

Prečo Rust?

- Pamäťová bezpečnosť bez garbage collector
- Predvídateľný výkon v reálnom čase
- Async/await neblokujúce programovanie
- Comptime checking zabraňuje runtime errorom
- Vytvorenie funkčného prototypu master control jednotky pre kritickú IoT aplikáciu.
- Overenie konceptu integrácie bezpečného softvéru v Ruste s vlastným hardvérovým dizajnom.
- Praktická demonštrácia riadenia vysoko-napäťových systémov.
- Vzor pre vývoj spoľahlivých a cenovo efektívnych embedded riešení.