Rozsdás BluePill intro
avagy mikrokontrolleren is biztonságban

Veres-Szentkirályi András
vsza@vsza.hu

Online Meetup
2020. augusztus 15.

Rust 1/3: intro

- alapokhoz: rhapsodhy előadása a 3. online meetupról
  - https://www.youtube.com/watch?v=iWYuEFD1fms#t=531s
  - https://gist.github.com/rsdy/7d93222bd9bcf851ab9239c38ac6d8ef
  - https://hsbp.org/online-meetup
- amit most kihasználunk:
  - nincs runtime / GC
  - kompakt gépi kódra fordul

Bluepill 1/3: intro

- kb. kis számítógép
- van CPU-ja: 32-bites ARM mikrokontroller
- van memóriája: 20k RAM
- van perzisztens tár: 64k Flash
- vannak porjai: USB, USART, I^2C, SPI, CAN, ...

Rust 2/3: multiarch

- LLVM-nek köszönhetően sok architektúrára fordul
- nem feltétlenül igényel maga alá operációs rendszert
- így közvetlenül OS készítésére is jó: Redox OS
- vagy mikrokontrollerekre!
- `https://rust-embedded.github.io/book`

Bluepill 2/3: ARM

- ARM: Advanced RISC Machine
- maga a cég csak IP-t ad el, gyártók: ST, Apple, ...
- ARM és ARM között van különbség
  - Cortex-M: Bluepill, nincs MMU, nincs/pici OS, *bare metal*
  - Cortex-A: okostelefon, van MMU, konzumer OS, *hosted*

Rust 3/3: OS nélkül

- mit adott nekünk az MMU/OS?
- fájlrendszer, hálózatkezelés, multithreading, multiprocessing
- virtuális memória
  - Windows usereknek: access violation
  - Unix(-szerű) usereknek: segmentation fault (`SIGSEGV`)
- legnagyobb baj, ami történhet: process kilövése
- ha ezek eltűnnek, jól jön egy-két plusz ellenőrzés

Bluepill 3/3: MMU/OS nélkül

- „mint DOS-ban”
- nincs heap (magától)
- van stack, de felülírhat bármit
- statikus méretű pufferek „kézzel”
- C++: lásd Arduino, semmi new

Blink: Hello world hardverre

- „villogjon egy LED ütemesen”
- demonstrálja a boilerplate kódot
- GPIO inicializálás és írás
- időzítés
- `https://github.com/stm32-rs/stm32f1xx-hal/blob/master/examples/blinky.rs`

Blink: gyakorlat

- érdemes README-t végigolvasni:
  `https://github.com/stm32-rs/stm32f1xx-hal`
- környezetenként egyszer
  - `rustup` – ha amúgy nem lenne Rust környezet
  - `rustup target add thumbv7m-none-eabi` – cross compiler
  - OS függőségek: GDB multiarch, OpenOCD
  - `$HOME/.gdbinit` fájlban `auto-load safe-path` beállítása
  - OpenOCD parancssor kitalálása (STlink verziótól függően)
- projektenként egyszer
  - `cargo init` – business as usual
  - függőségek hozzáadása
  - `.cargo/config` hozzáadása: mire, hogyan forduljon
  - `memory.x` hozzáadása: mi hol van a memóriatérképen
  - `.gdbinit` hozzáadása: `cargo run` utáni „mágia”

Hogyan tovább?

- cargo crate-ek
  - kulcsszó: `no_std`
  - legtöbb funkcióra van megoldás, még ha kényelmetlenebb is, mint `std`-vel
  - demo: HMAC + SHA-512 + Base64
- (semi?)online workshop
  - mindenki megkapja ugyanazt a hardvert
  - bütykölés, konzultáció, tapasztalatok megosztása
  - jelentkezz: hack@hsbp.org

Köszönöm a figyelmet!