A Zephyr User Story

Franco Saworski
ProGlove

- First bare metal
  - Hardware components dictated by product
  - Bare metal prototypes became product
ProGlove

- First bare metal
  - Hardware components dictated by product
  - Bare metal prototypes became product
- Then port to Zephyr
  - Externalize build system
  - Abstract architecture
  - Modular board support
ProGlove

- First bare metal
  - Hardware components dictated by product
  - Bare metal prototypes became product
- Then port to Zephyr
  - Externalize build system
  - Abstract architecture
  - Modular board support
- From prototype to CE in twelve, later nine months
blik

- Zephyr on the sensor unit from day one
- From prototype to CE in six months
Lessons Learnt

- Four years shipping industry grade products with small teams in startups
- First job bare metal
  - Did not scale
  - Reinventing the wheel
Lessons Learnt

- Four years shipping industry grade products with small teams in startups
- First job bare metal
  - Did not scale
  - Reinventing the wheel
- Then port to Zephyr
  - Ported too many legacy dependencies
  - Did not upstream enough custom code
Lessons Learnt

• Four years shipping industry grade products with small teams in startups
• First job bare metal
  – Did not scale
  – Reinventing the wheel
• Then port to Zephyr
  – Ported too many legacy dependencies
  – Did not upstream enough custom code
Lessons Learnt

- Four years shipping industry grade products with small teams in startups
- First job bare metal
  - Did not scale
  - Reinventing the wheel
- Then port to Zephyr
  - Ported too many legacy dependencies
  - Did not upstream enough custom code
- Second job native Zephyr
  - Still not upstreaming enough
  - Using off the shelf components where possible
  - Sticking them together with Zephyr
Lessons Learnt

- Four years shipping industry grade products with small teams in startups
- First job bare metal
  - Did not scale
  - Reinventing the wheel
- Then port to Zephyr
  - Ported too many legacy dependencies
  - Did not upstream enough custom code
- Second job native Zephyr
  - Still not upstreaming enough
  - Using off the shelf components where possible
  - Sticking them together with Zephyr
Requirements

Product

- Fast prototypes
- Fast iterations
- Fast to market
Requirements

Product

- Fast prototypes
- Fast iterations
- Fast to market
- Low resources
- Growing feature backlog
Requirements

Product
• Fast prototypes
• Fast iterations
• Fast to market
• Low resources
• Growing feature backlog

Development
• Solid build system
• Solid architecture
• Solid interfaces
• Solid ecosystem
Requirements

Product
- Fast prototypes
- Fast iterations
- Fast to market
- Low resources
- Growing feature backlog

Development
- Solid build system
- Solid architecture
- Solid interfaces
- Solid ecosystem
- Community
- Standards
The RTOS choice

- Hardware Support
- Toolchain
- Testing
- APIs / Interfaces
- Drivers
- Application libraries
- License
The RTOS choice

- Hardware Support
- Toolchain
- Testing
- APIs / Interfaces
- Drivers
- Application libraries
- License

- Zephyr
- mbedOS
- Contiki
- FreeRTOS
- (RiOT)
Looking back

• Zephyr meets all requirements
Looking back

- Zephyr meets all requirements
- It helped me ship products fast
Looking back

- Zephyr meets all requirements
- It helped me ship products fast
- I was able to maintain products with small teams
Looking back

- Zephyr meets all requirements
- It helped me ship products fast
- I was able to maintain products with small teams
- Its active community encourages participation
Looking back

- Zephyr meets all requirements
- It helped me ship products fast
- I was able to maintain products with small teams
- Its active community encourages participation
Thank you!
Questions?

Join the monthly Munich Embedded Meetup: https://groups.io/g/embeddedmeetup

franco.saworski@blik.io

All pictures attribution free from https://pixabay.com unless indicated otherwise.