

XP for Embedded Software

Keith Braithwaite: WDS Global

xp@keithbraithwaite.demon.co.uk

Nancy Van Schooenderwoert: XP Embedded

nancyv@xp-embedded.com

Introduction

Audience: Embedded Developers who want to know how XP can help / Developers who may be moving to embedded work

- We'll contrast embedded development with regular" providing brief overview for those without embedded experience
- We'll introduce extreme practices specifically for embedded software development

Schedule

- From Desktop to Embedded
- The Robustness Requirement
- TDD for Embedded Software
- Q&A

Schedule

- From Desktop to Embedded
- The Robustness Requirement
- TDD for Embedded Software
- Q&A

From the Desktop to the Embedded Device

Areas of Embedded Software growth (annual)

* Point-of-sale Equipment 26%

* Wireless LAN hardware 25%

* Warehouse Management Systems 19%

Source: Venture Development Corporation

From the Desktop to the Embedded Device

- What's missing?
- What's new?
- What's bridged by the toolchain?
- What's not, even though it looks like it?

What's missing?

- Local storage – no disk!
- File system
- Display – may have only a status LED
- Operating System – many embedded apps are “bare metal”

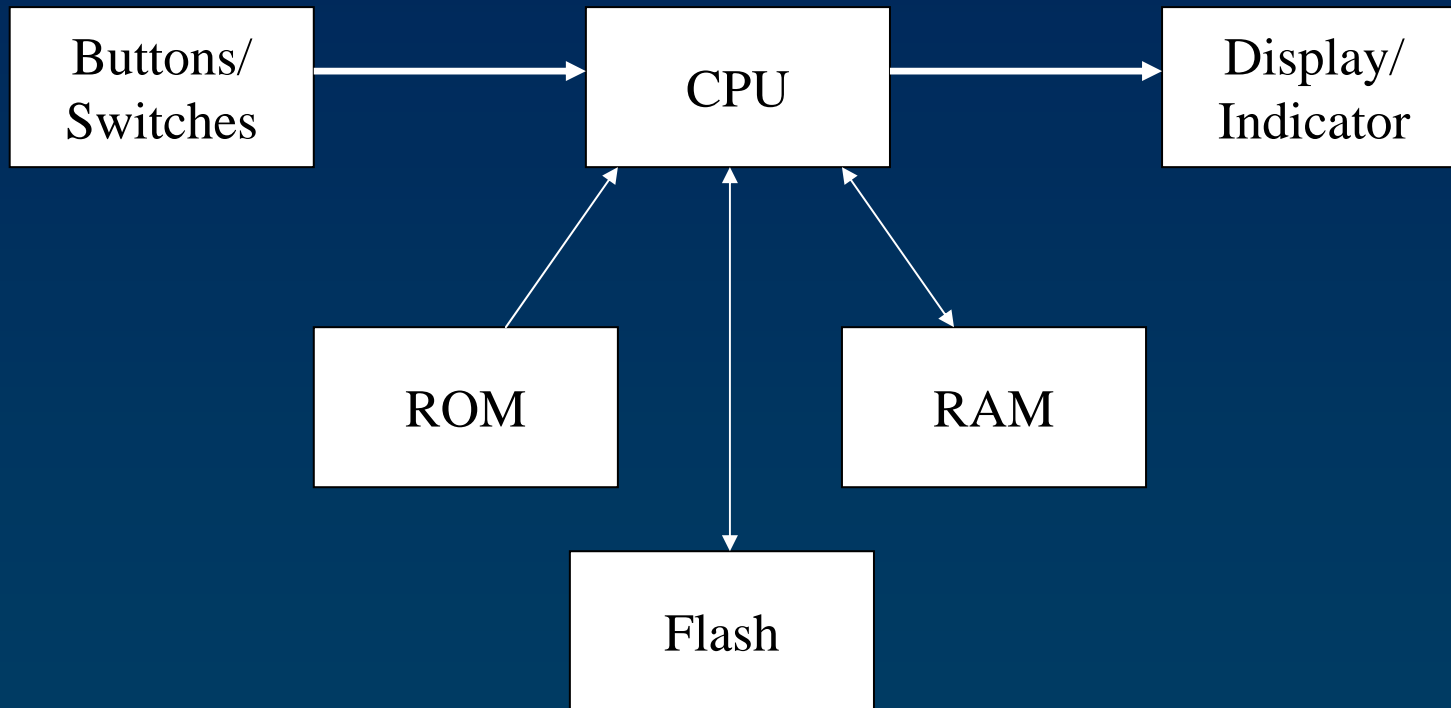
What's new?

- Boot sequence is part of user experience for your app
- App life longer than toolchain life
- Embedded devices networked together -> emergent new uses for them, therefore...
- App design should not assume user behavior

What's new? (Cont'd)

- Interrupts, NMI, enable registers
- “Floating” inputs when unused pins left unwired

Typical Embedded System



Timeline – Power-up to Shutdown



What's bridged by the toolchain?

- Cross-compilation to target CPU
- Endian-ness?
- Boot code built into debugger
- Initialization of board devices
- Monitoring of execution status
- Role of Flash, ROM, RAM masked

What's not, even though it
looks like it?

Schedule

- From Desktop to Embedded
- The Robustness Requirement
- TDD for Embedded Software
- Q&A

The Robustness Requirement

- What is it?
- How do we do it?

What is it?

- resource hygiene (e.g. Semaphores, mem leak detection?)
- fail safe
- fail operational
- automatic restart...
- Leave a trail – trouble log
- Leave a signal – status indicators

How do we do it?

- RTOS techniques, e.g.
Semaphores, tasking design
- tuned language implementations
- patterns
- Tools...
- Co-operative design

Schedule

- From Desktop to Embedded
- The Robustness Requirement
- TDD for Embedded Software
- Q&A

TDD for Embedded Software

- What is it?
- How do we do it?
- Why we do it even more than usual?

What is it?

- TDD keeps you on track
- Makes it hard to code unnecessary extras
- Fairly complete test coverage – but you must modify the tests as the code changes
- TDD produces good raw material to help QA and Testers

How do we do it?

- Only write code after you've written a test that is not working – helps you to know when you're done
- Dual-Target your code for desktop & target processor
- Make each Package (“Domain”) able to run solo on both platforms
- Trouble log always 'on' in memory

Why do it More Than Usual?

- Best way to cut risk
- Surface issues early
- TDD proves unambiguously what was covered by testing - “audit trail”
- Keep the code malleable; provide good foundation for next mods

Schedule

- From Desktop to Embedded
- The Robustness Requirement
- TDD for Embedded Software
- Q&A

Q&A