The Connected MCU Lab

Dr. Alexander G. Dean
Dept. of Electrical and Computer Engineering
North Carolina State University

Imagination Community

These materials produced in association with Imagination.
Join our University community for more resources.
community.imgtec.com/university

MICROCHIP

Imagination

DIGILENT®

NC STATE UNIVERSITY

Presentation Overview

- Introduction
- Course Context and Mechanics
- Course Content
- Access and Timeline

Introduction

The Connected MCU Lab Course Goals

Introduce design of embedded systems and IoT devices

- Why is this course special?
  - Why embedded systems?
    - Critical industry need for embedded systems developers
  - Why IoT?
    - An IoT device leverages the Internet for remote data logging, monitoring and control, offloaded processing, sophisticated user interfaces, faster development, code reuse, etc.
  - Why so early in the curriculum?
    - Grab students' attention early with compelling, hands-on work, providing momentum to get through rough spots before the fun begins
    - Provide fuel and tools for their imagination (*I know what I could do with this…*)
- How to make it work?
  - Balance breadth and depth: provide **just enough depth** to understand fundamental concepts (and get students excited!)
    - Computer Architecture
    - Embedded Systems
    - Operating Systems
    - Compilers
    - Networking
    - Algorithms and Data Structures
  - Provide context and motivation for in-depth courses later in curriculum

Sponsors

Imagination

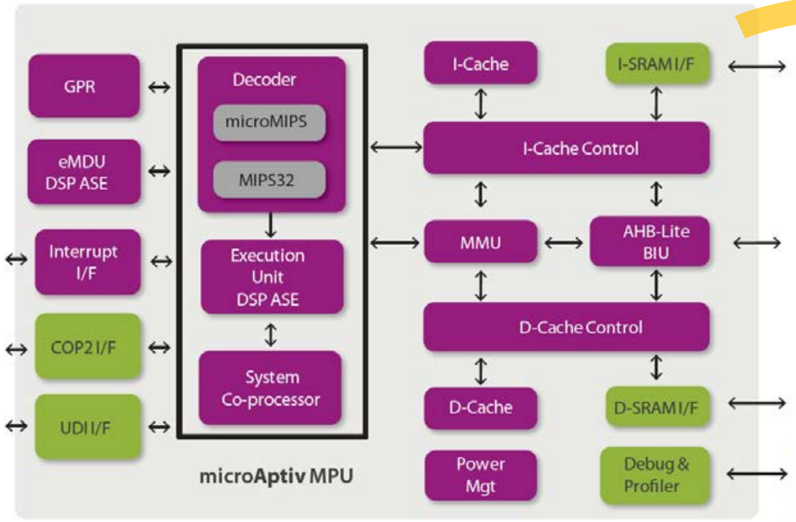
MIPS32 microAptiv processor core

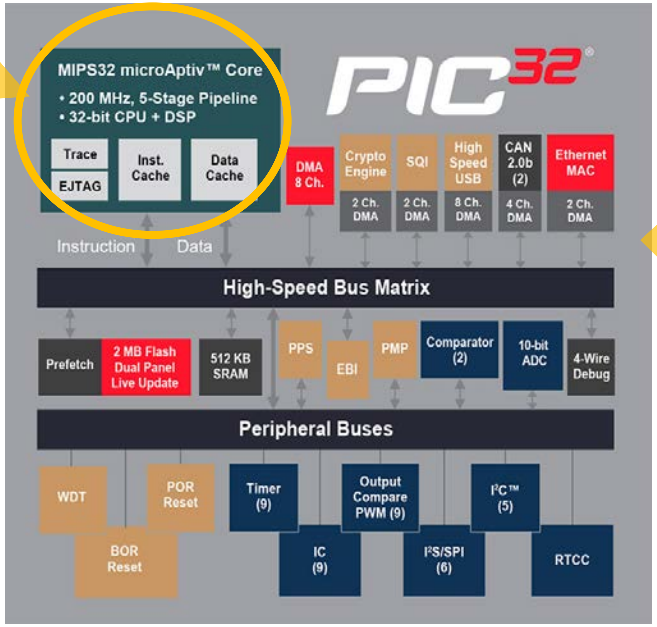
MICROCHIP

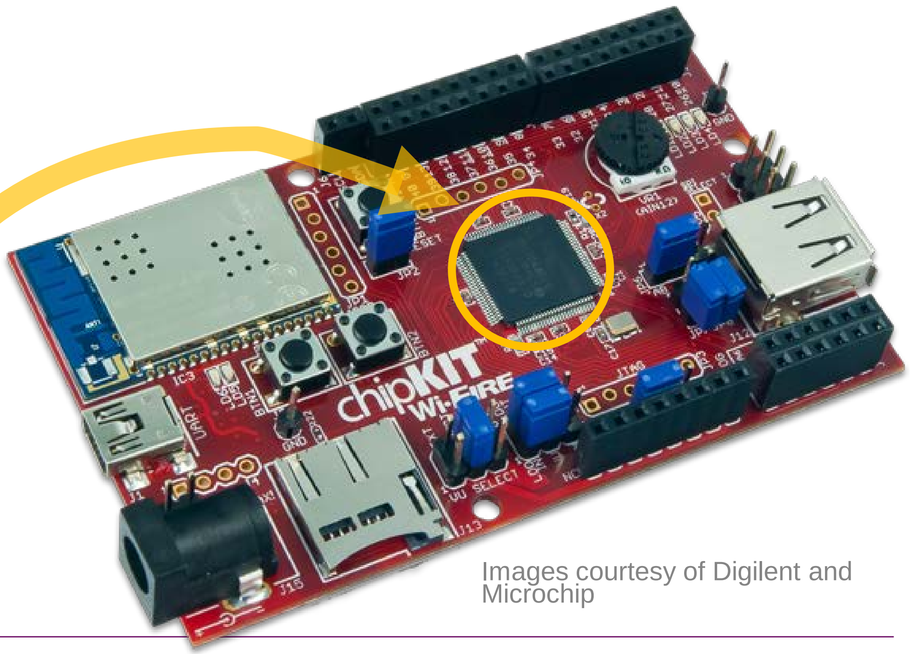
PIC32MZ microcontroller

DIGILENT

chipKIT Wi-FIRE development board







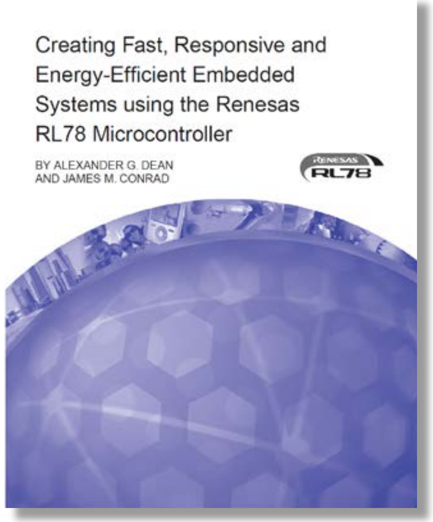
Images courtesy of Digilent and Microchip

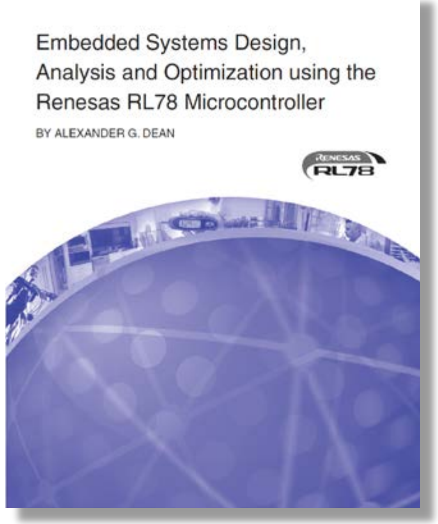
My Background

- Schooling
  - BSEE UW-Madison (1991), MS & PhD ECE Carnegie Mellon (1994, 2000)
- Work
  - Electronics technician in college (3 y)
  - United Technologies Research Center (2 y)
  - At NCSU since 2000
- Consulting
  - Over 80 embedded software design reviews and training workshops for industrial control applications since 2001

My Background: Teaching

Courses Developed and Taught





- Introduction to Embedded Systems (Junior/Senior) ➔ Used in two industry university programs
  - Programming an MCU in C
  - CPU architecture and interrupts
  - Using peripherals
  - Embedded system design practices
- Embedded System Design, Analysis and Optimization (Senior/Grad) ➔ Used in two industry university programs
  - Designing multithreaded systems with a real-time kernel
  - Analysis and optimization for…
    - **Responsiveness** (schedulers, real-time systems)
    - **Throughput** (profiling, code tuning and optimization)
    - **Power** or **Energy** (modeling, analysis, software & hardware aspects, sleep modes)
    - **Memory Size** (analysis, optimizations)
  - Embedded system S/W engineering
- Advanced Embedded Systems (Grad)
  - Linux on a Beaglebone Black
  - Linux and embedded systems (device interfacing, thread scheduling, etc.)
  - Code throughput optimization using SIMD instructions
  - Integrating a web server with an embedded system
  - High-level synthesis to FPGAs

My Background: Research

- Areas
  - Embedded and real-time systems
  - Compilers and object code analysis
  - Computer architecture and memory systems
  - Kernels and schedulers
  - Switch-mode power conversion
- Major Project Examples
  - Software Thread Integration: compiler creates implicitly multithreaded functions to provide **real-time concurrency** or better **instruction-level parallelism**
  - Static timing and stack depth analysis for AVR and ARM object code
  - Using scratchpad memory to reduce or eliminate **cache-based timing variability** multithreaded real-time systems
  - Replacing hardware switch-mode power converter controllers with software on an application MCU using real-time system design methods to **reduce costs and size**
  - Internet-connected coastal environmental monitoring system (water depth, temperature, clarity, salinity, oyster feeding activity)
  - Ultrasonic communication system for down-well applications

Course Context and Mechanics

Teaching Approach

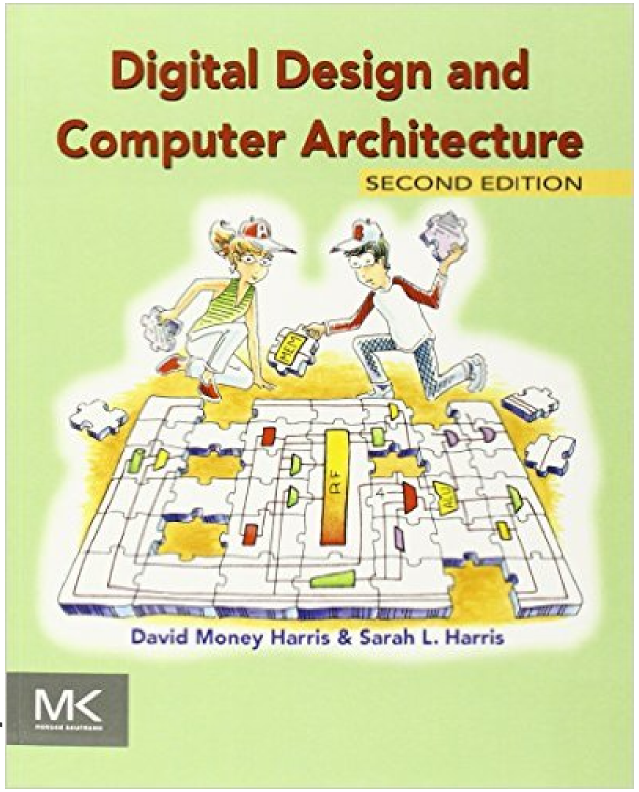
Hands-on with real hardware and software

- Materials are sequenced to give a hands-on experience from the start
  - First lab follows first lecture, features tweaking code to flash an LED
- Modules draw from different traditional ECE and CS subject areas…
  - When appropriate, and with appropriate depth and breadth.
    - Not this: “The CPU executes machine code to implement an instruction set architecture. Let’s examine all the possible ISA types, dimensions and features: RISC, CISC, Load/Store, Memory/Memory, Memory/Register, VLIW, SIMD, MIMD…”
- Instead, what’s most important for an embedded systems designer to know?
  - For example, about compilers:
    - Backus-Naur Form grammars? No
    - Parsers? No
    - Code generation? Yes
    - Optimizations? Yes
    - Type promotion? Yes

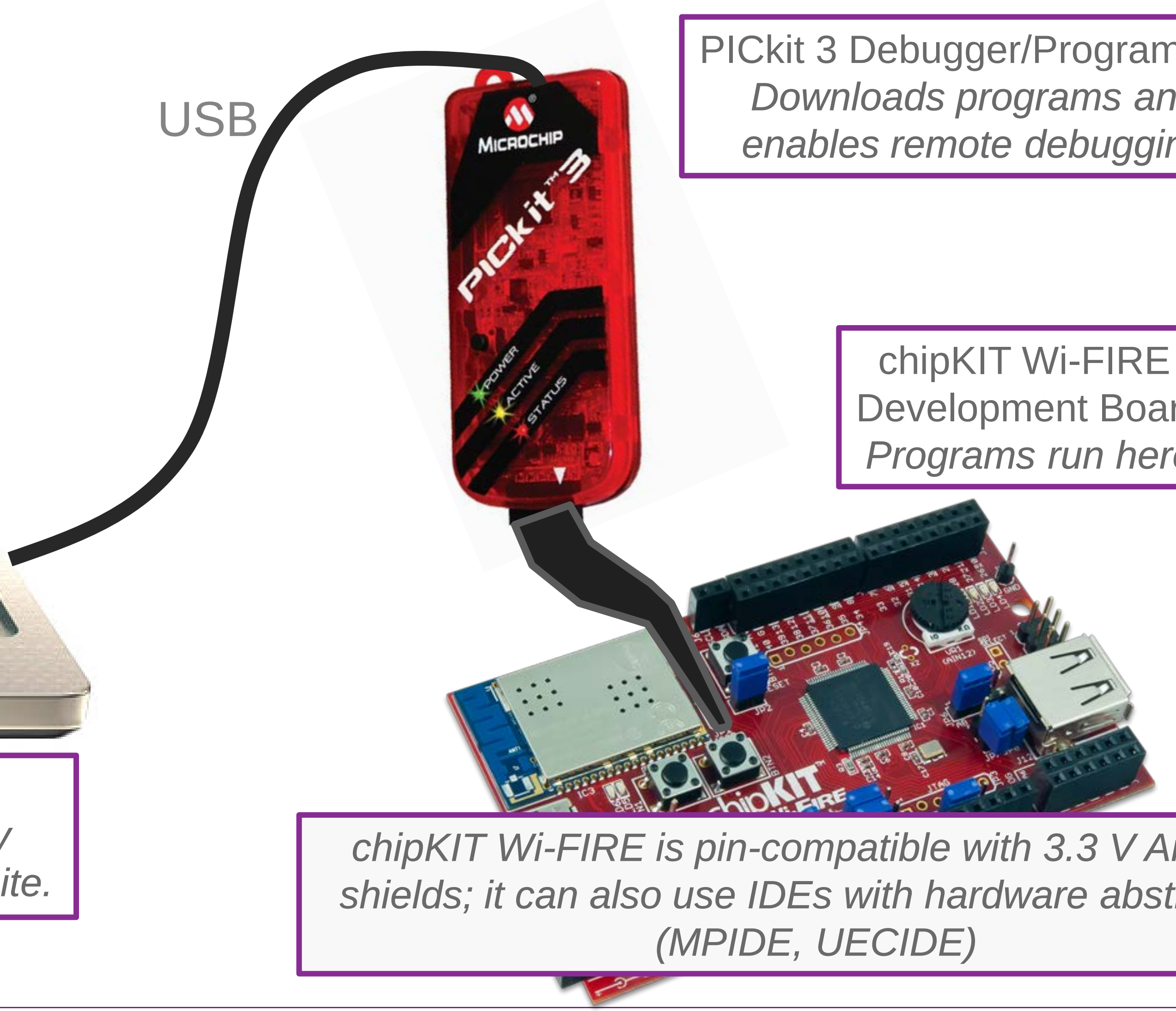
Where Does This Fit in the Curriculum?

- EE/ECE undergrad curricula typically have two early mandatory lab courses
  - Digital Design
  - Embedded Systems or Microcontrollers
- First or second-year course, one semester (16 weeks)
  - Prerequisite: introductory C programming class

- Leverages **Digital Design and Computer Architecture**, Harris & Harris for depth on:
  - C programming
  - MIPS architecture and assembly language
  - Interrupts
  - Embedded I/O
  - Fixed and floating-point number systems
  - Caches
  - Virtual memory



Development Environment Overview



PICkit 3 Debugger/Programmer
Downloads programs and enables remote debugging

chipKIT Wi-FIRE Development Board
Programs run here

MPLAB X IDE on Development PC
Create and debug programs in C, C++, assembly language here with Microchip's professional tool suite.

chipKIT Wi-FIRE is pin-compatible with 3.3 V Arduino shields; it can also use IDEs with hardware abstraction (MPIDE, UECIDE)

Target Platform

Imagination

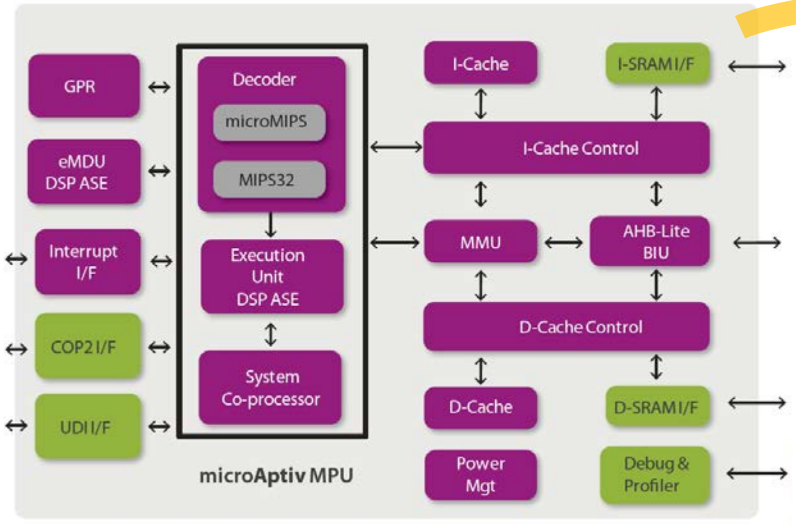
200 MHz MIPS32 microAptiv processor core

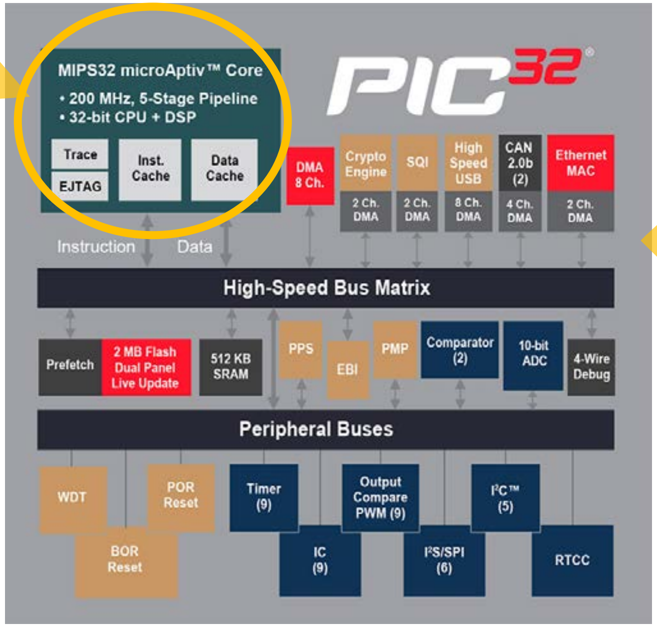
Microchip

PIC32MZ microcontroller with 512 KB SRAM, 2 MB Flash ROM

Digilent

chipKIT Wi-FIRE dev. board with Microchip WiFi module, switches, USB, pot, 4 LEDs, microSD card, expansion headers





Images courtesy of Digilent and Microchip

Expansion Option: chipKIT Basic I/O Shield

- OLED graphical display (128x32 pixels)
- Analog potentiometer (knob)
- 4 slide switches
- 4 pushbutton switches
- 8 LEDs
- Temperature sensor (I^2C)
- 32 KB non-volatile memory (EEPROM)
- 4 open drain FET drivers

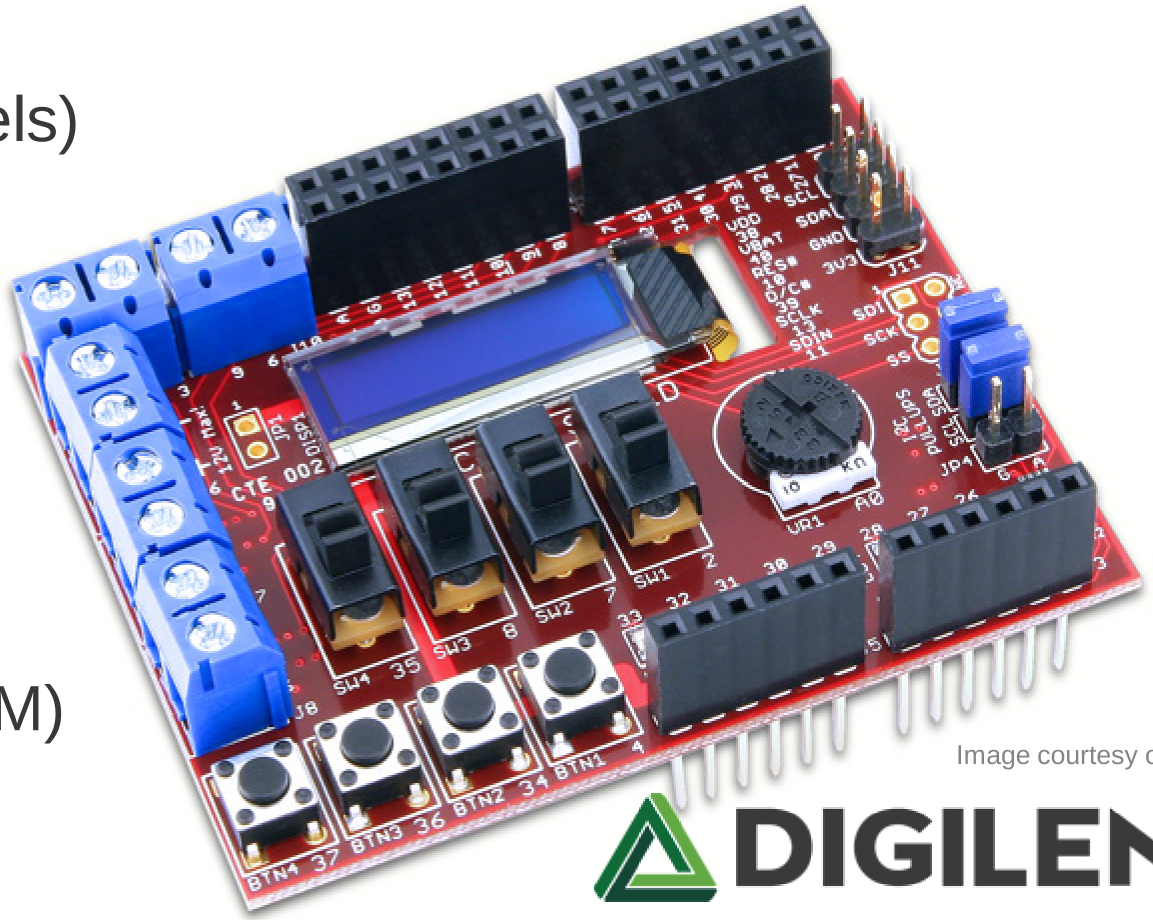


Image courtesy of Digilent

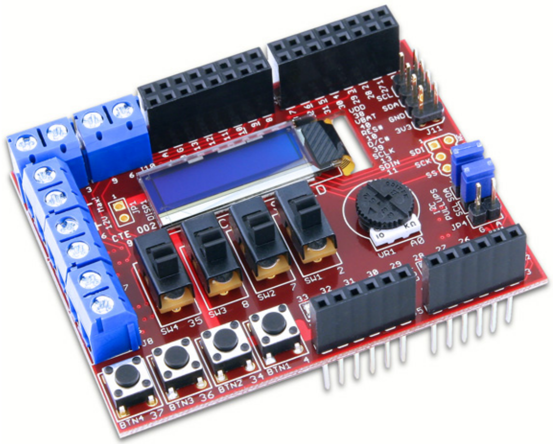
DIGILENT®

www.digilentinc.com

Typical Prices for Materials

PICkit 3
Debugger/Programmer
$47.95 (USD)

chipKIT Wi-FIRE Board
$79 (USD)



chipKIT Basic I/O
Shield
$37.99 (USD)

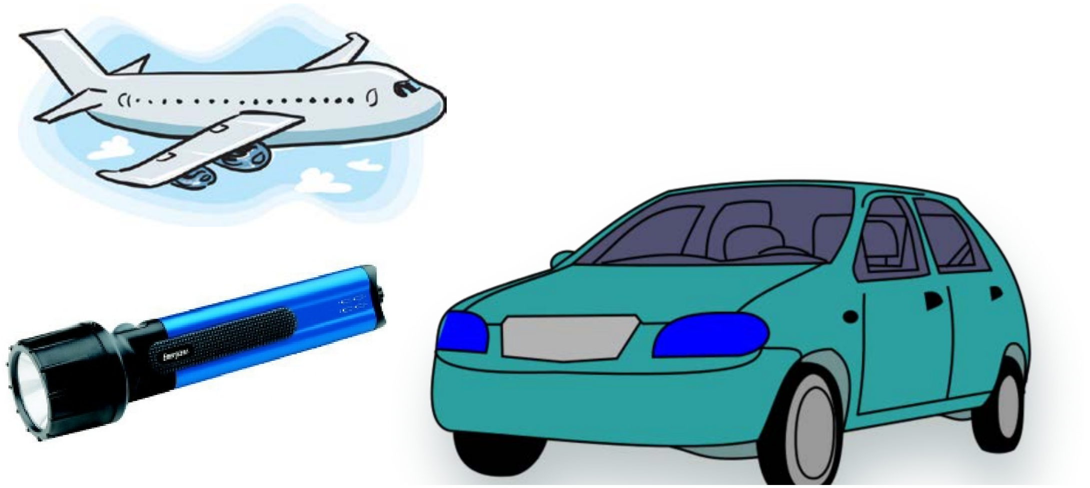
Educators and students receive 25% off when they purchase through www.microchipDIRECT.com

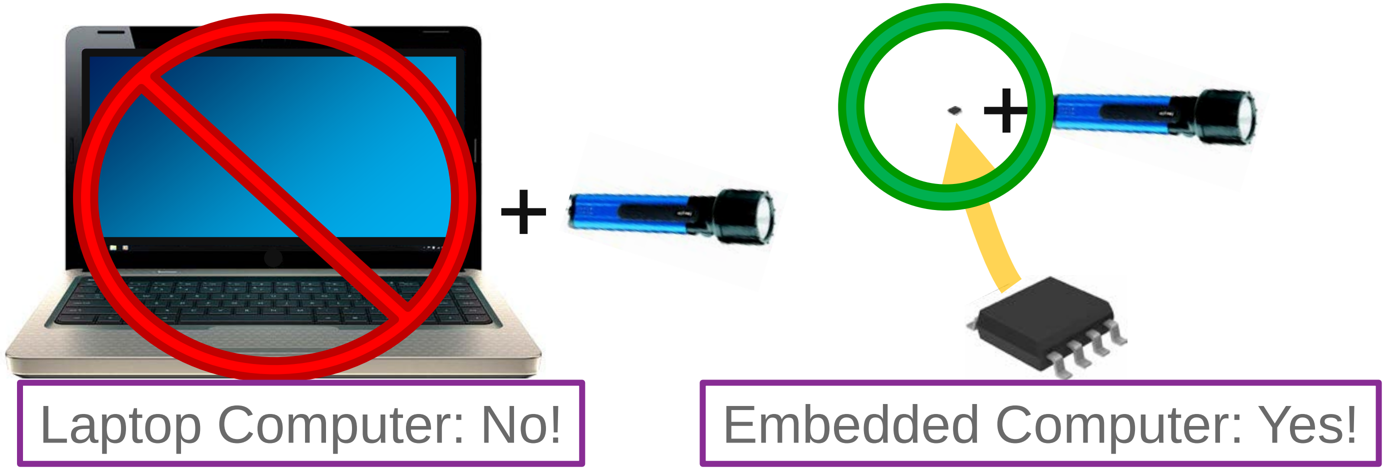
Course Contents

Embedded (Computer) Systems

What, why and how?

- What is an embedded computer system?
- Why embed a computer?
  - Performance, efficiency, features, component costs, development costs, flexibility, longevity
- How to embed a computer?
  - Microcontroller vs. microcomputer
  - Peripherals
  - Multithreaded software and illusion of concurrency
- Differences from general-purpose computers

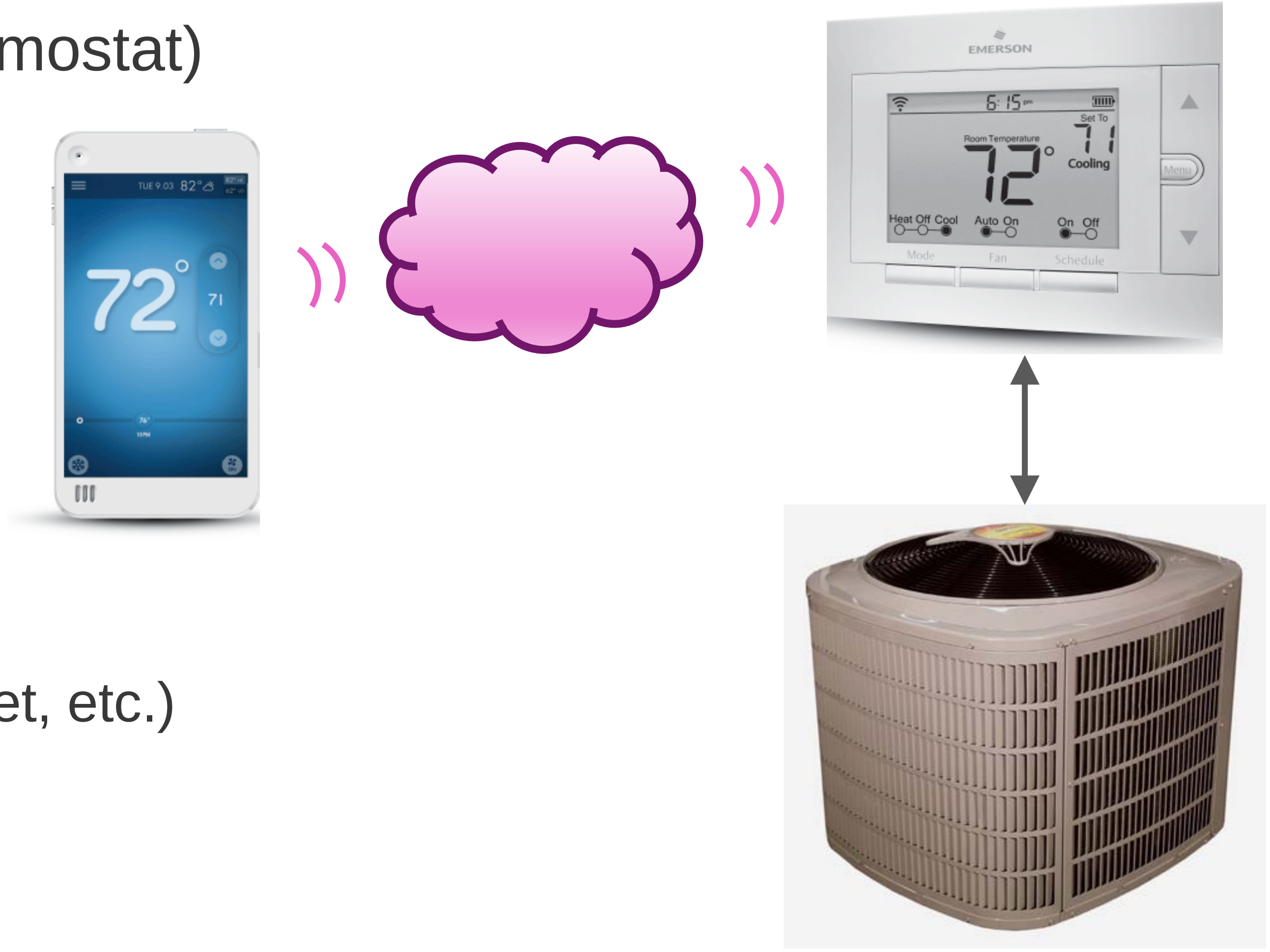




Internet of Things

Network connection provides benefits

- Application-specific benefits (e.g. thermostat)
  - User can ease temperature setpoints if stuck at office working late
  - Power company can ease temperature setpoints if too much power draw on grid
  - System can notify service company if equipment is failing
- General benefits
  - Richer user interfaces (smartphone, tablet, etc.)
  - Sophisticated features, data analysis by offloading processing to cloud
  - Remote data archiving
  - Easier code maintenance and updates



Microcontroller Fundamentals

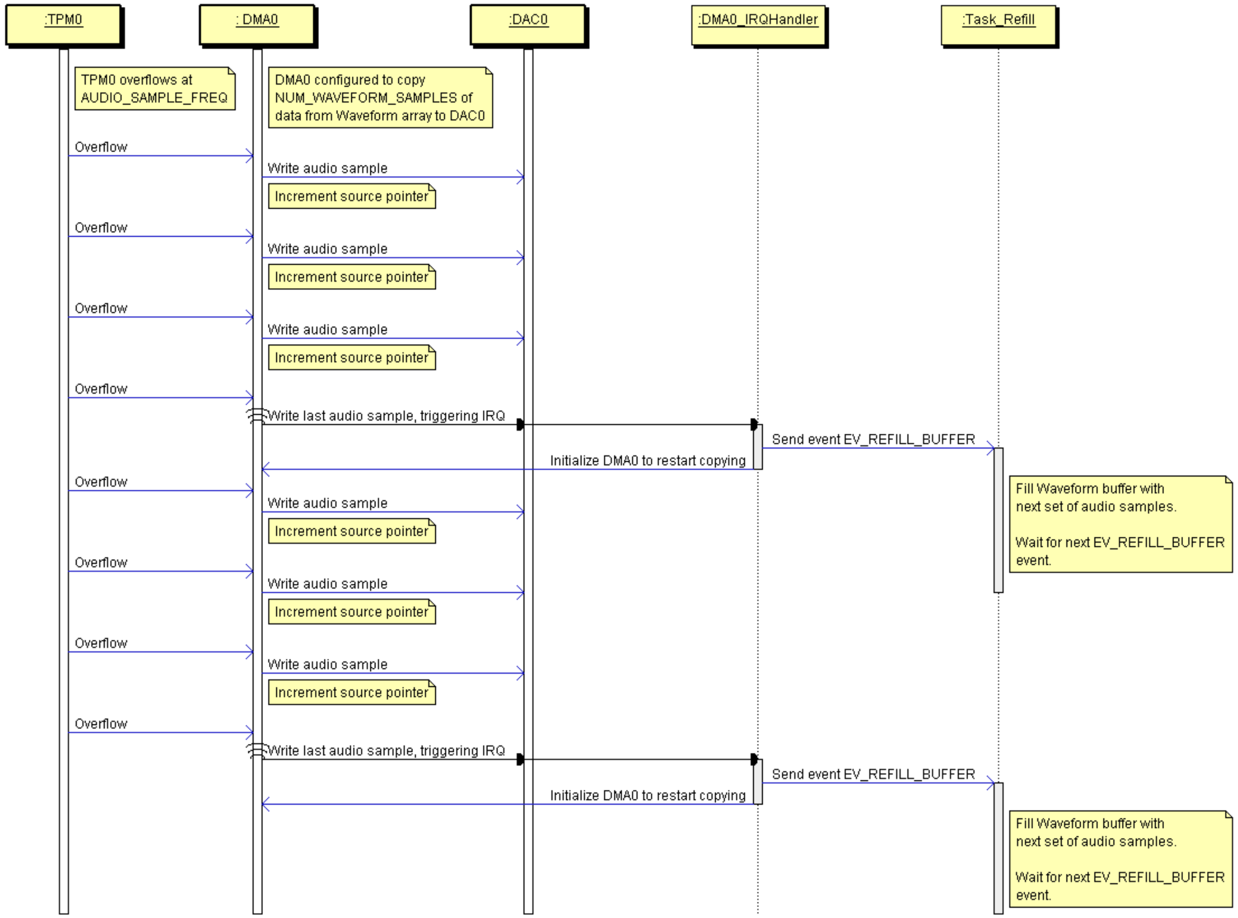
Microcontroller Programming and Interfacing

- MCU Concepts
  - CPU, memory, peripherals
  - Concurrent execution
- Hands-on experience with most important peripherals, survey of others
  - GPIO
  - Analog-to-Digital and Digital-to-Analog converters
  - Timers: interval timers, counters, pulse-width modulation, input capture
  - Communications: Async. serial, SPI, I^{2}C, WiFi (Microchip MRF24G), USB, Ethernet
  - Robustness: Watchdog timer, low-voltage detector
  - Other: clock generators, DMA controllers, accelerators
  - Interfacing with Arduino shields
- Software development
  - Source code vs. machine code
  - Software toolchain and the build process
  - Debugging tools and process

Creating Multithreaded Systems

Operating Systems

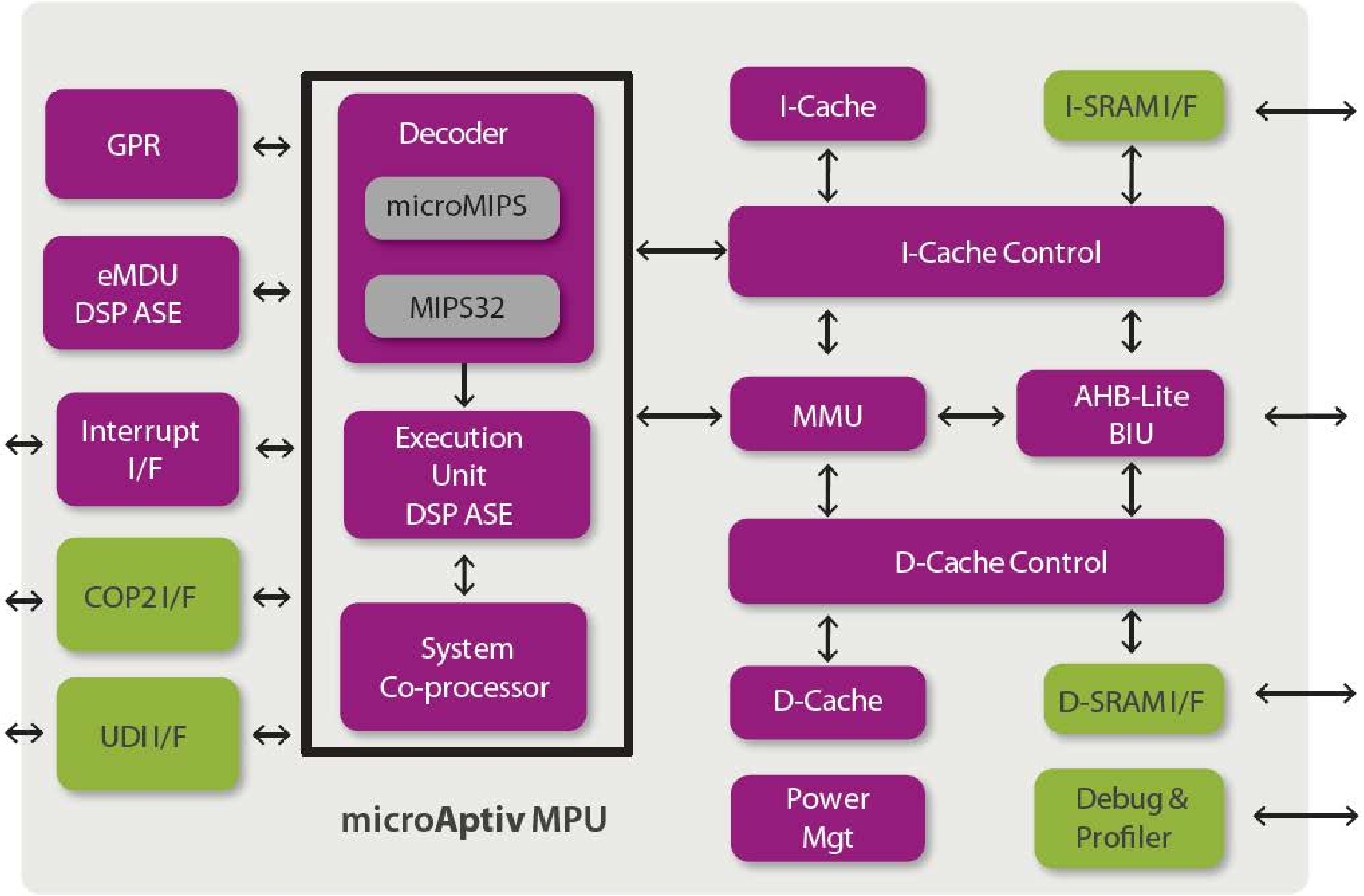
- Concurrency
  - Hardware and software
  - Sequence diagrams
- Supporting concurrency
  - Task prioritization and preemption concepts
  - Interrupts
  - Preemptive kernel - FreeRTOS
- Creating multitasking systems
  - Synchronization and communication
  - Basic real-time system scheduling concepts (schedulability, worst-case response time)



Looking Under the CPU’s Hood

Computer Architecture, Operating Systems

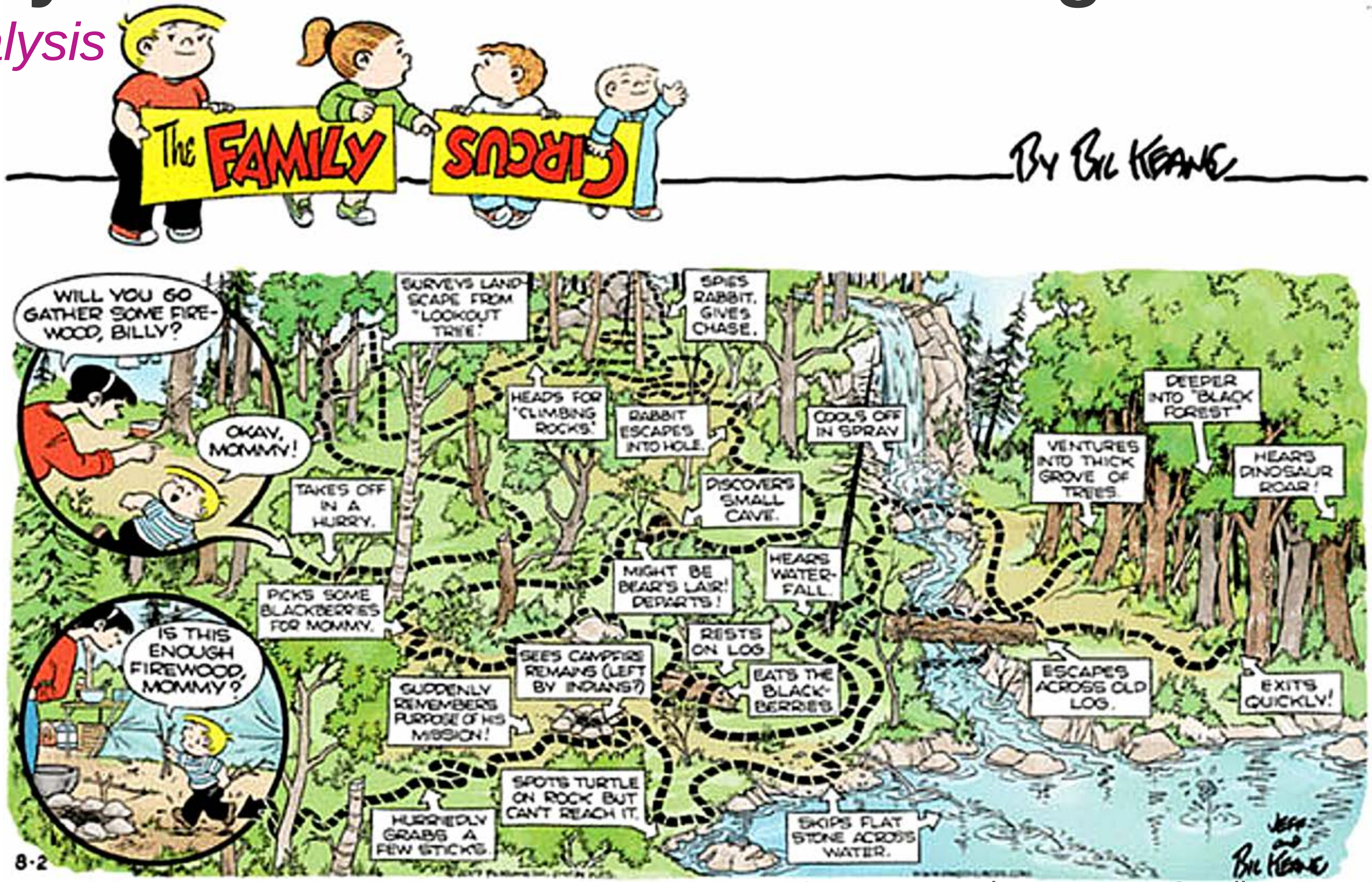
- PIC32MZ CPU features
  - MIPS32 ISA
  - DSP ASE
  - Interrupts and exceptions
- Microarchitecture concepts
  - Instruction pipelining
  - Branch prediction
  - Out-of-order execution
- Memory systems
  - RAM, ROM, caches
  - Memory management (memory protection and virtual memory)



Performance Analysis: What's the CPU Doing?

Compilers, Performance Analysis

- What's the compiler telling the CPU to do?
  - Start-up code
  - Functions, activation records, call stacks
  - Variables and data structures
  - Data processing
  - Control flow: if/else, loops, …
  - Calling functions, returning
  - Libraries
- Profiling
  - Determine which code dominates execution time
  - Methods: Periodic PC sampling vs. instrumentation

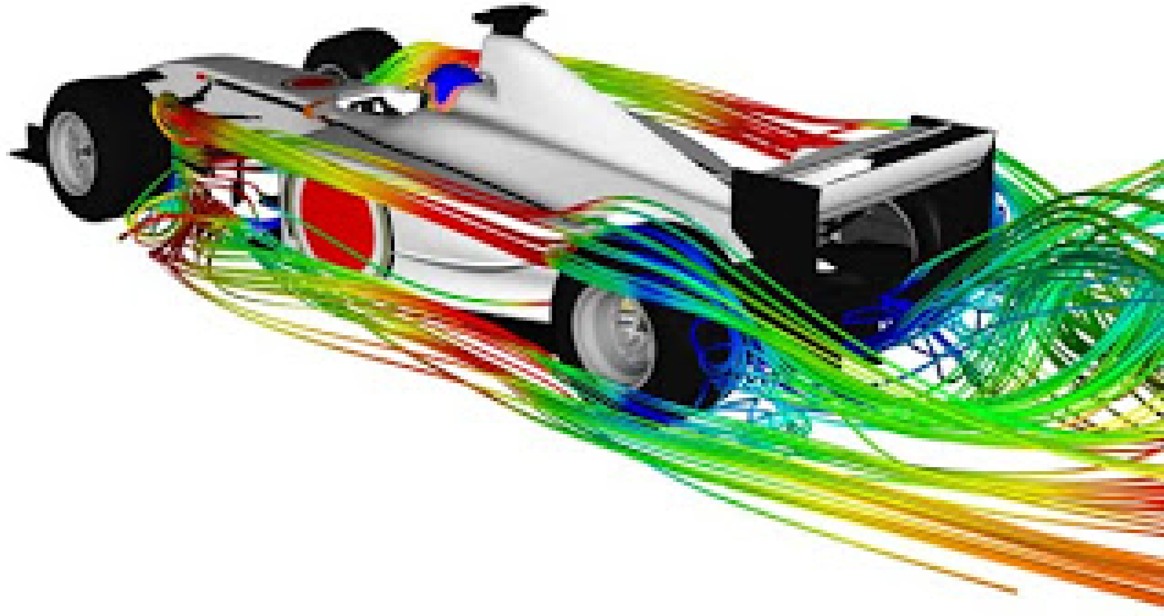


J & B. Keane, King Features Syndicate

Performance Optimization: Make it Faster!

Compilers, Data Structures, Algorithms

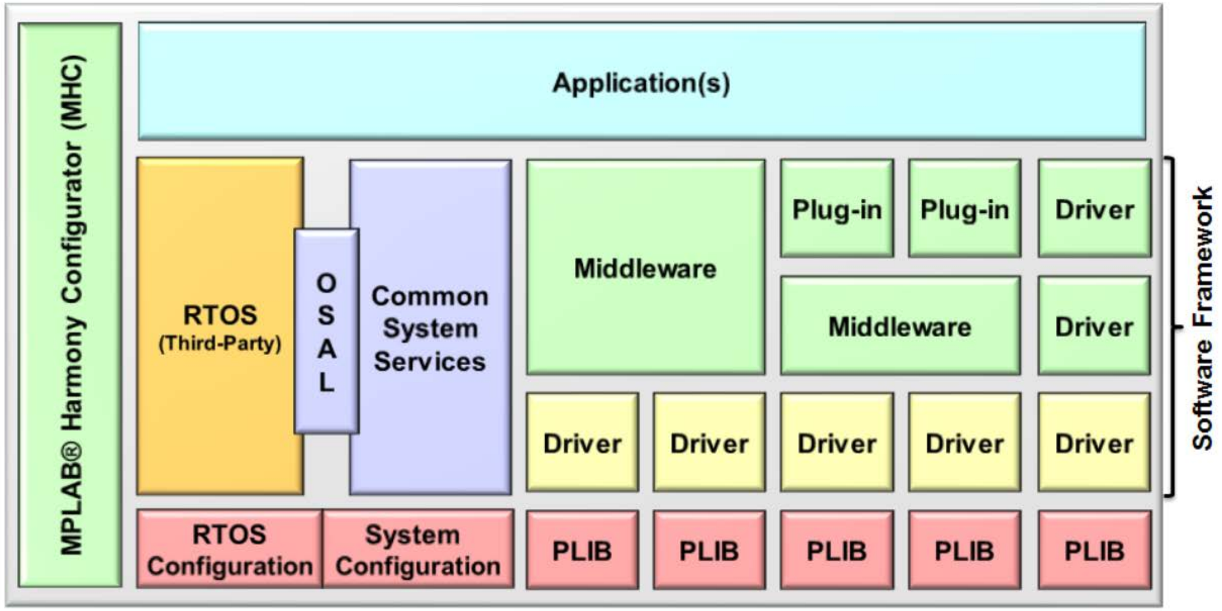
- Examining object code
  - Did the compiler do a good enough job?
  - Can we help the compiler and toolchain more?
- Modifying source code
  - Fit compiler and tool-chain better
  - Delete or delay work when possible
  - Use ISA-specific instructions and mechanisms
- Modifying algorithms
  - Complexity theory
  - Better data organization (sorted data, data structures)



Creating Complex Systems

Using the Microchip MPLAB Harmony Integrated Software Framework

- Structure
  - Configurator for MCU and peripherals
  - Device drivers and peripheral abstraction
  - Middleware
  - System services to manage shared resources
- Benefits
  - Written in C, portable across Microchip MCUs
  - Saves development time & effort
  - Sample applications
  - Scheduling approach
    - Built on state machines, so works with RTOS or superloop
    - Non-blocking (asynchronous) module execution with notifications when complete

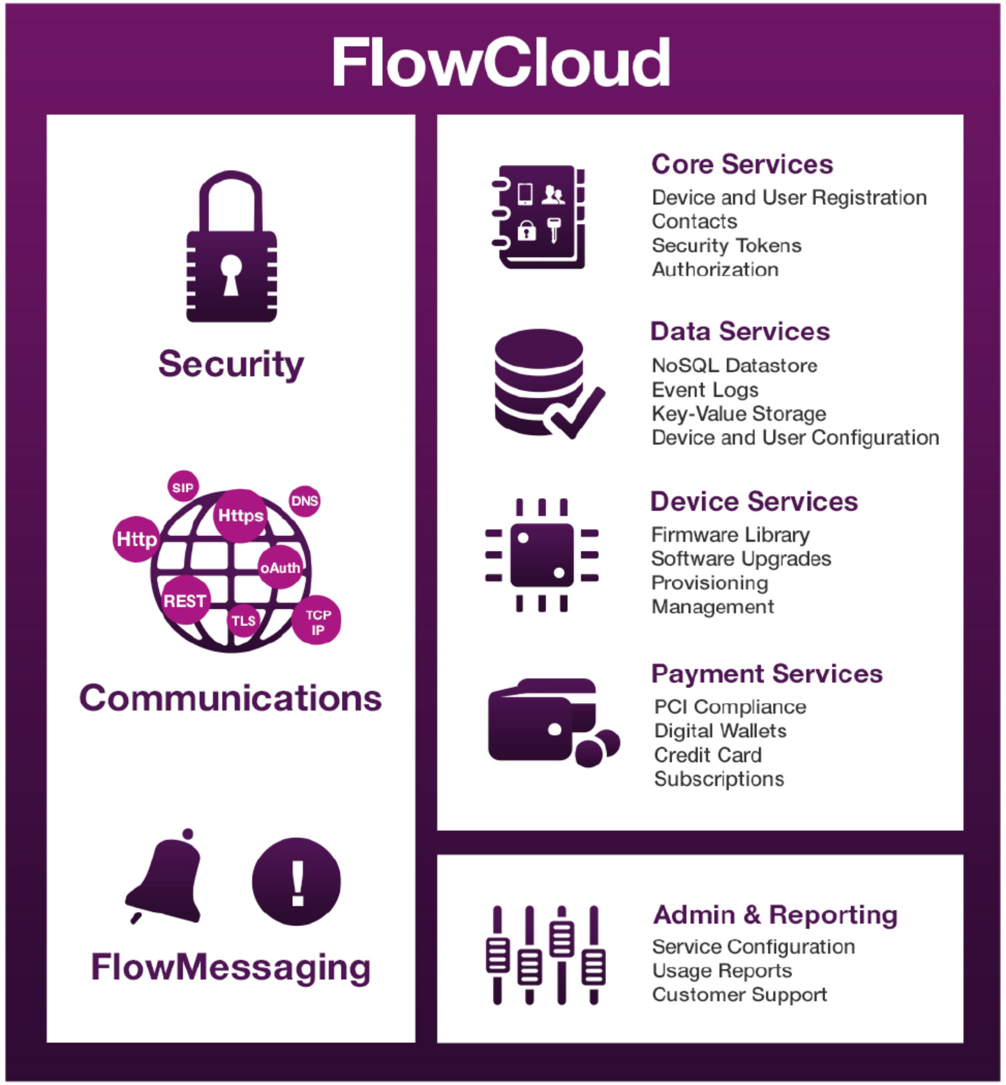


- Highlights
  - Middleware: Bluetooth, audio decoders, graphics, fixed-point math, TCP/IP, USB, crypto, bootloader, TLS, …
  - Device Drivers: Ethernet MAC & PHY, LCD controllers, I2C, I2S, audio codecs, CAN, ADC, camera, SD Card, touch controller, WiFi …

Creating an IoT Device

Uses Imagination Technologies' FlowCloud

- Leverages Wi-FIRE's WiFi module (Microchip MRF24G), Microchip Harmony
- Introduction to IoT Concepts
- FlowCloud Services
  - Core
  - Data
  - Device
  - Payment
- Application design
- Project: Turning a weather station into an IoT device



Access and Timeline

Access to Teaching Materials

1. Join IUP: Visit www.imgtec.com/university
   - Click Join IUP
   - Includes e-mail verification stage
2. Visit Imagination Community http://community.imgtec.com/university/resources
   - Log in
   - Request download
   - Accept license (unrestricted for academic use), provide details of intended use
3. Receive “Request Approved” e-mail with download confirmation

- Potential early adopters are encouraged to contact IUP or me for beta-test opportunity
- Contact IUP via forum: http://community.imgtec.com/forums/cat/university/

Timeline

- Material development until September 2015
- Beta-test in October. Materials will be available through website using password-protected invitation
- Material launch online in December
- Press releases and promotions Q1 2016
- Workshops March and April 2016
- Video tutorials online Q2 2016
- Hardware donations available for select Universities March – July 2016 using request form

Thank you for your attention.

Are there any questions?

agdean@ncsu.edu

http://community.imgtec.com/forums/cat/university/