EE2905 Intro

Last updated 9/6/22

EE2905 Intro

- This class is an introduction to Embedded Systems Programming
  - This course introduces students to embedded systems, embedded programming concepts, and basic electronics interfacing. Course topics include microcontroller architecture, subsystems, and embedded systems terminology. The course includes limited coverage of electrical interfacing of embedded systems to external digital and analog electronics. An integrated high-level programming environment is used. Students complete a course project that emphasizes the interaction between the hardware and software components of a practical embedded system. (prereq: none)

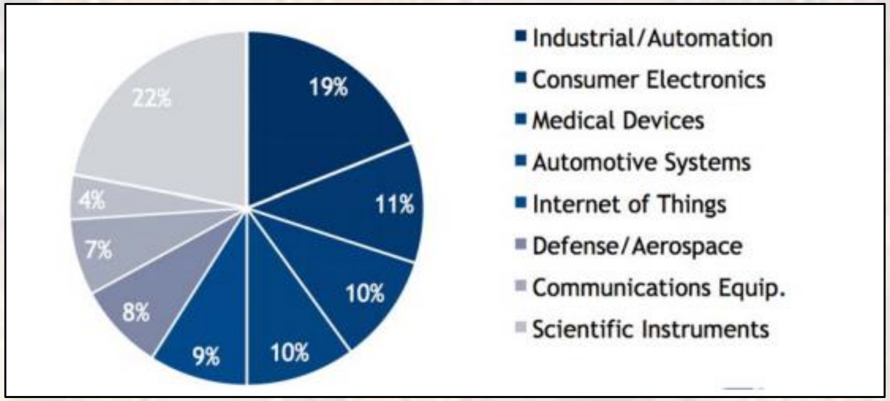
EE2905 Intro

- **Course Learning Outcomes**
  Upon successful completion of this course, the student will be able to:
  - Describe the general sub-systems and operation of embedded controllers
  - Describe biomedical applications of embedded systems
  - Describe the purpose of integrated development environments
  - Describe and effectively use data types in a modern, high-level computing environment
  - Describe and effectively use control constructions in a modern, high-level computing environment
  - Describe and effectively use digital inputs and outputs, PWM outputs, and analog inputs and outputs in a high-level computing environment running on modern embedded system hardware
  - Describe and effectively use user defined functions or blocks in a modern, high-level computer programming environment
  - Describe and effectively use provided classes and libraries in a high-level computer programming environment to program on a modern embedded processor
  - Design, create, and document relatively simple embedded program

EE2905 Intro

- What is an Embedded System
  - Computer (processor, core) based electronic system
  - Runs manufacturer code
  - Does not support User/3rd Party code
    - Note: In this class **you** are the manufacturer
  - Think: washing machine, defibrillator, Bluetooth headset
  - Not: PC, application portion of cell phone

End Market Applications for Embedded Systems



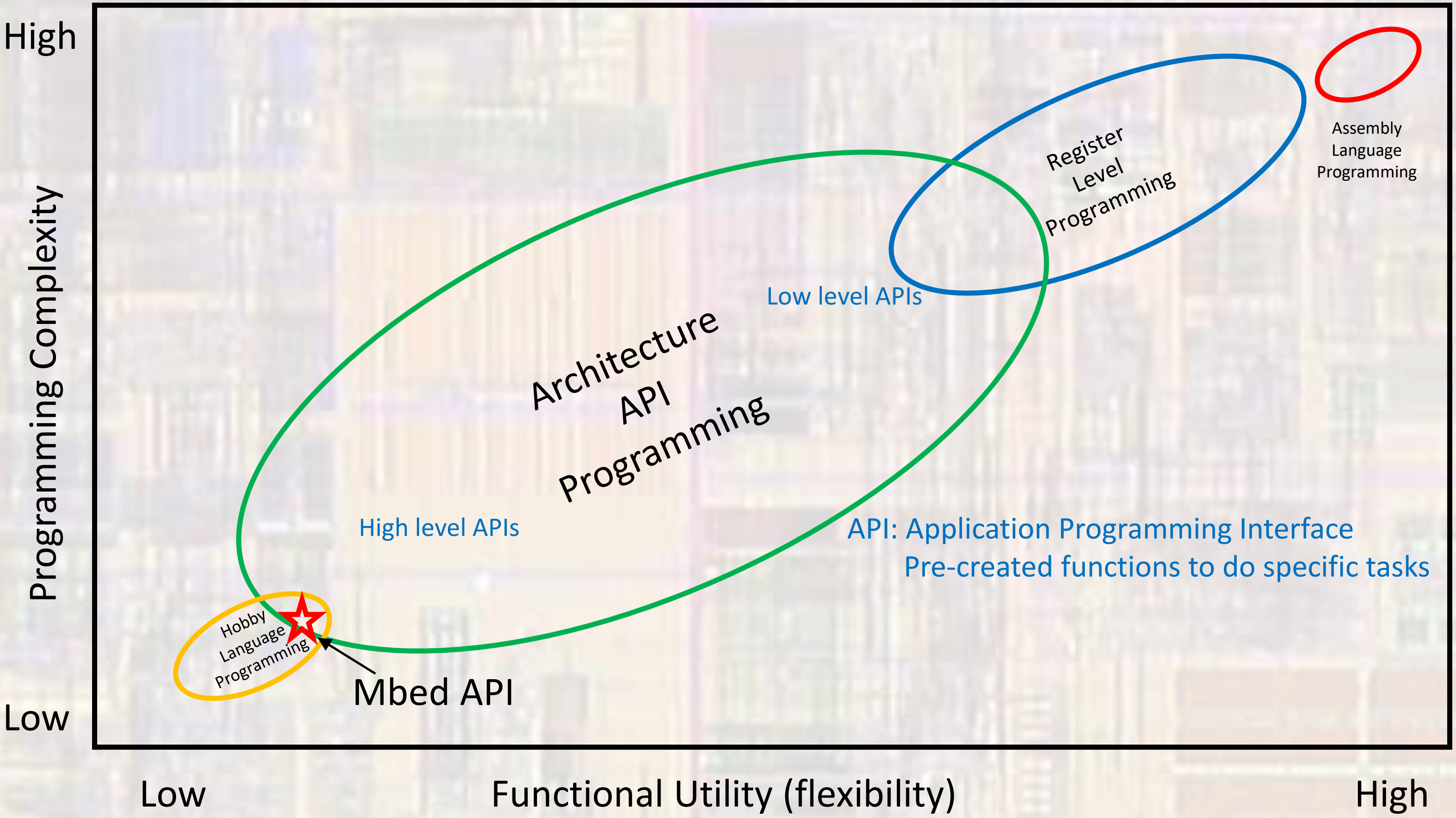
Src: Barr Group

EE2905 Intro

- Issues Driving Embedded Systems Programming
  - Resources
    - Memory
    - CPU power
  - Power Consumption
    - Week long watch operation
    - Day long cell phone operation
  - Operating Systems
    - None
    - Real Time (RTOS)
  - Hardware
    - Cores are dominated by ARM – but not exclusive
    - Peripherals vary significantly

EE2905 Intro

- Embedded Systems Programming Environments

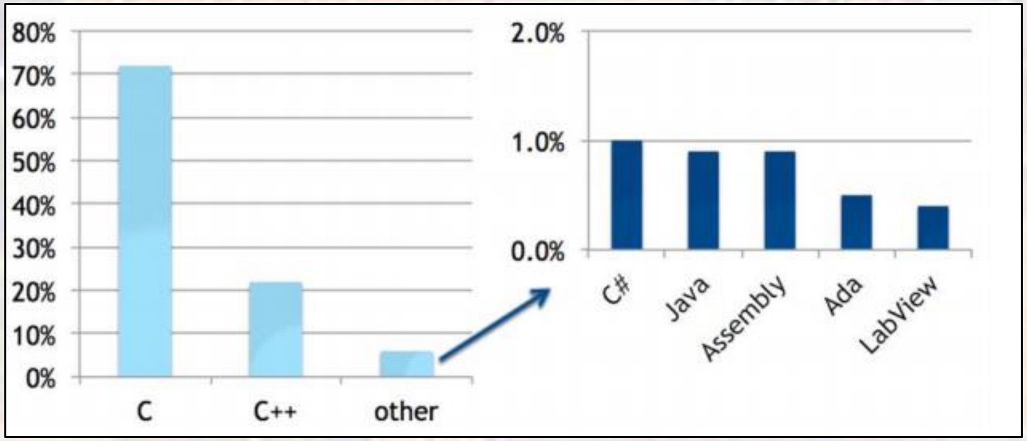

High
Programming Complexity
Low
Assembly Language Programming
Register Level Programming
Low level APIs
Architecture API Programming
High level APIs
Hobby Language Programming
Mbed API
Low
Functional Utility (flexibility)
High

API: Application Programming Interface
Pre-created functions to do specific tasks

EE2905 Intro

- Embedded Systems Programming Languages
  - C
    - Original and traditional choice
    - Allows for low level access to processor capabilities
    - High memory efficiency (program and data)
  - C++
    - Simplifies programming (OOP)
    - Less efficient wrt. memory

Programming Languages for Embedded Systems

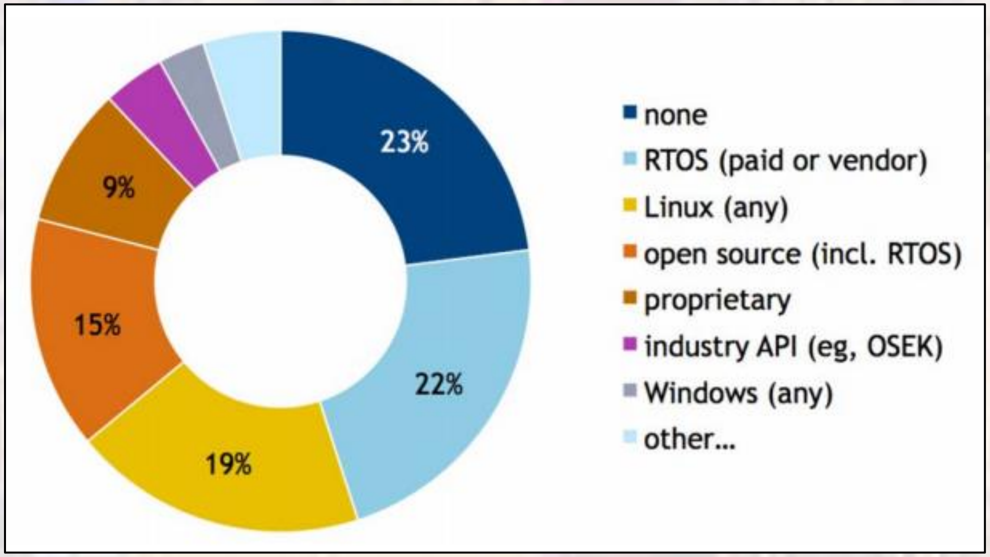


Src: Barr Group

EE2905 Intro

- Embedded Systems Operating Systems

Operating Systems for Embedded Systems



Src: Barr Group

- Mbed in an opensource RTOS
  - We will use very few of it's capabilities

EE2905 Intro

- Mbed Platform
  - Open source embedded system development environment
  - Based on C++
  - Includes an RTOS

  - On the hobbyist end of the spectrum
    - Little to no low level functionality
    - Good high level functionality (heavily constrained)
  - Tools are Bad
  - Constantly changing
  - Bloated

  - We will barely use the RTOS
  - We will use primarily C programming techniques
    - With C++ Objects only where necessary

EE2905 Intro

- My understanding about YOU
  - >= Junior status
  - Programming experience from Matlab
  - Basic electric circuit understanding

- What you should know
  - 3rd graders can be taught to code
  - What makes you valuable is your ability to solve problems

- My primary goal
  - Make you successful – for the long term
    - Understand why you are doing something – not just what to do
    - Teach you how to teach yourself new concepts

EE2905 Intro

- University Policy

  - *For each academic credit hour, a typical undergraduate student is expected to spend two clock hours outside of class preparing for and studying for the class. Outside of class activities are required learning activities completed outside of instructional time, such as homework and reading.*

  - 3 class hrs → 6 reading/study/hw/programming hrs/week
  - 1 lab credit hr → 1 lab completion/documentation hr/wk (2 lab class hrs)

EE2905 Intro

- Plan for Success - Class
  - Night before class
    - Pre-read the class notes – 30min x 3 → 90min/wk
    - List any questions you may have
    - Print them out 4 or 6/page if you want to take notes on them in class
  - In class
    - Pay attention
    - Ask questions – I guarantee you are not the only one
    - Annotate anything that is not clear in the notes
  - HW
    - Should be solvable at least 3 days before due date – START EARLY
    - Targeted at 30min/hw x 2/week → 60min/wk
  - Programming
    - Should be solvable at least 3 days before due date – START EARLY
    - Targeted at 30min/program x 2/week → 60min/wk
  - Total 'outside' class commitment
    - 90min + 60min + 60min = 3.5hr << 6 hour university expectation

EE2905 Intro

- Plan for Success - Lab
  - At least 2 days before lab → 30min/week
    - Pre-read the lab
    - Do any Pre-lab activities
  - In Lab
    - Pay attention – critical information is provided at the beginning of lab
    - Have a plan
      - Flow chart
      - Hardware setup and verification
    - Ask questions
      - Spend a few minutes trying to figure out what's not working
      - Ask for help before you use up too much time
    - Document along the way
      - Save any pictures, scope plots, program outputs as you get them otherwise you may have to redo that part of the lab
    - Make sure you know what you need to do when you leave the lab
  - Post lab → 30min/week
    - Read the lab for required documentation
    - Err on the side of too much information