# EE712: Embedded Systems Design
# Spring 2022-23

Laxmeesha Somappa
Department of Electrical Engineering
IIT Bombay
laxmeesha@ee.iitb.ac.in

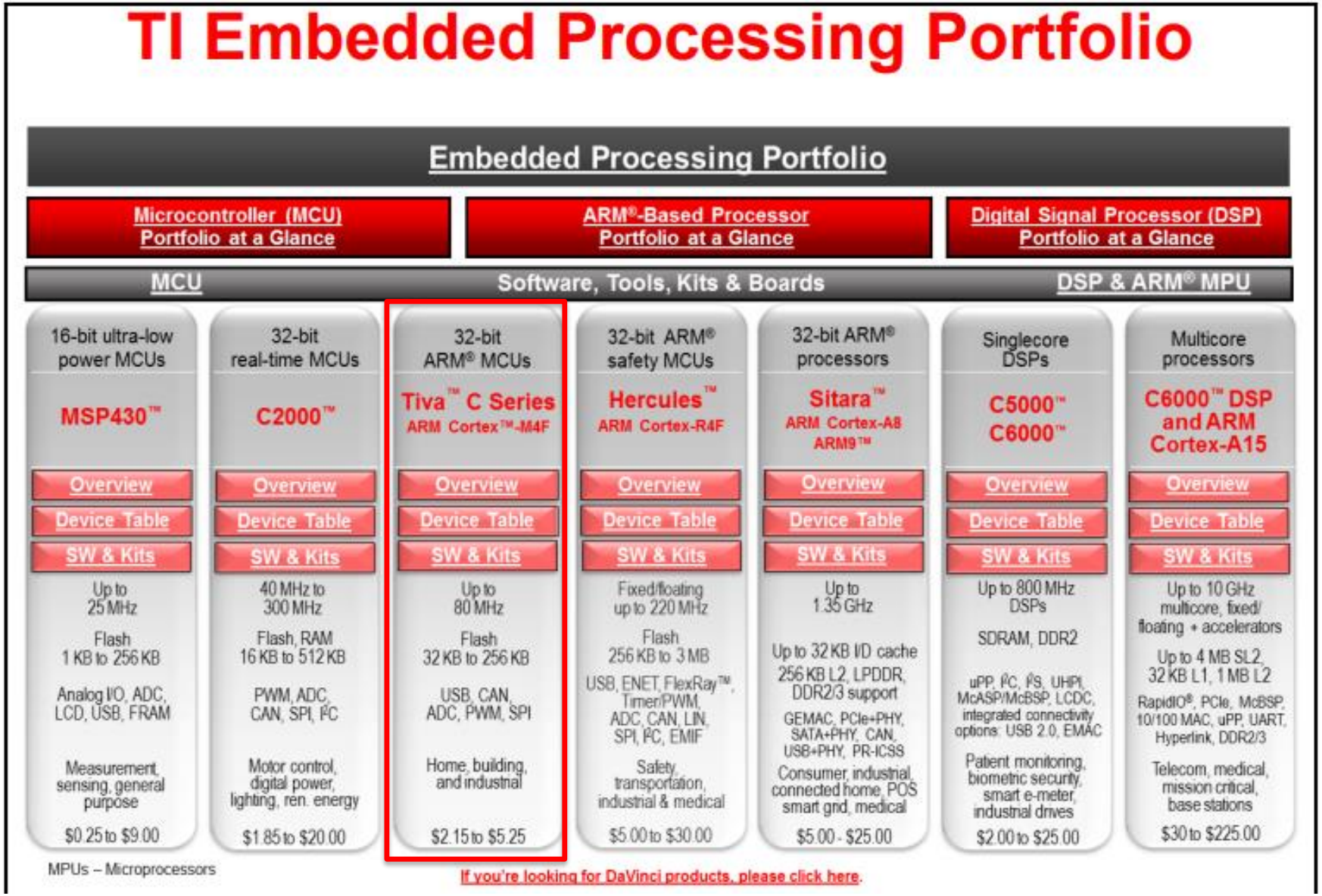
Dinesh Sharma
Department of Electrical Engineering
IIT Bombay
dinesh@ee.iitb.ac.in

# TIVA-C INTRODUCTION

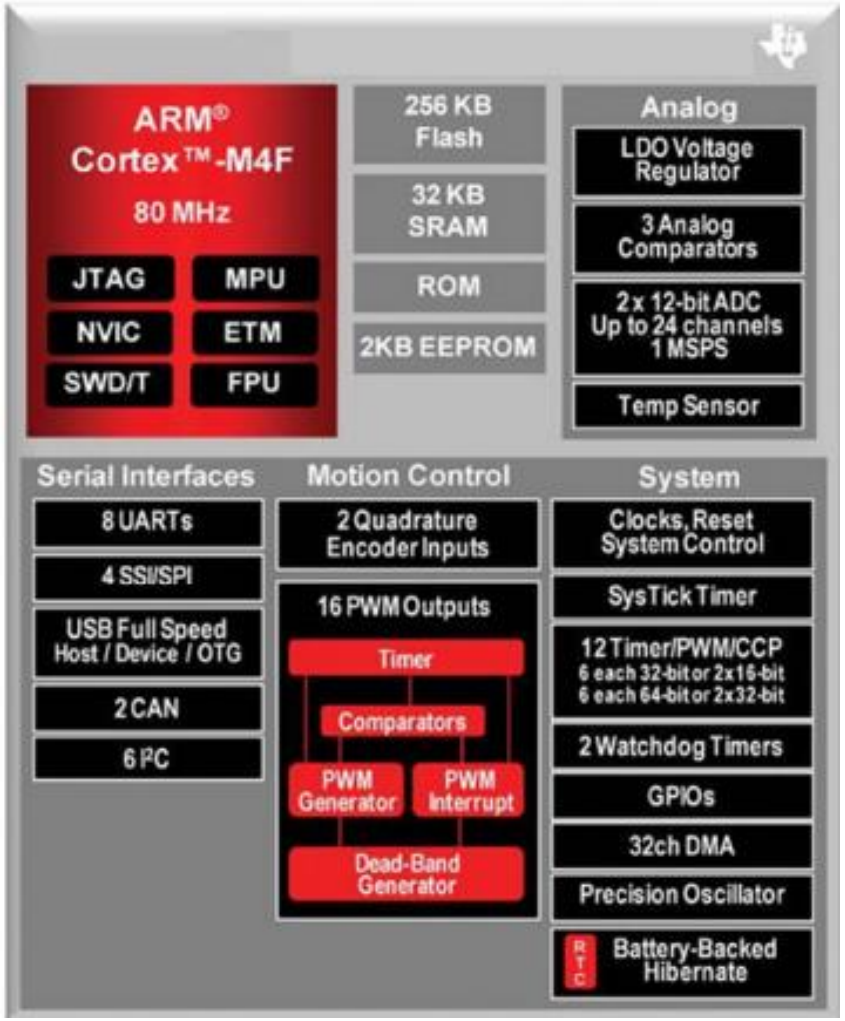
(COURTESY: TEXAS INSTRUMENTS)


laxmeesha@ee.iitb.ac.in, dinesh@ee.iitb.ac.in


# INTRODUCTION



# TIVA TM4C MICROCONTROLLER




- 32-bit ARM Cortex M4 Core
- System clock frequency: 80 MHz
- Unaligned data access
- Atomic bit manipulation
- Single precision floating point unit

- 8 –UART, 6 – I2C, 4 – SPI, 2 – CAN, USB 2.0
- Two SAR ADCs: 12-bit, 1 Msps (12 shared inputs)
- 3 – analog and 16 – digital comparators
- 32-channel uDMA
- 2 watchdog timers (separate clock)
- Nested-vectored interrupt controller


laxmeesha@ee.iitb.ac.in, dinesh@ee.iitb.ac.in

# TIVA TM4C MEMORY

**256KB Flash memory**

- Single-cycle to 40MHz
- Pre-fetch buffer and speculative branch improves performance above 40 MHz

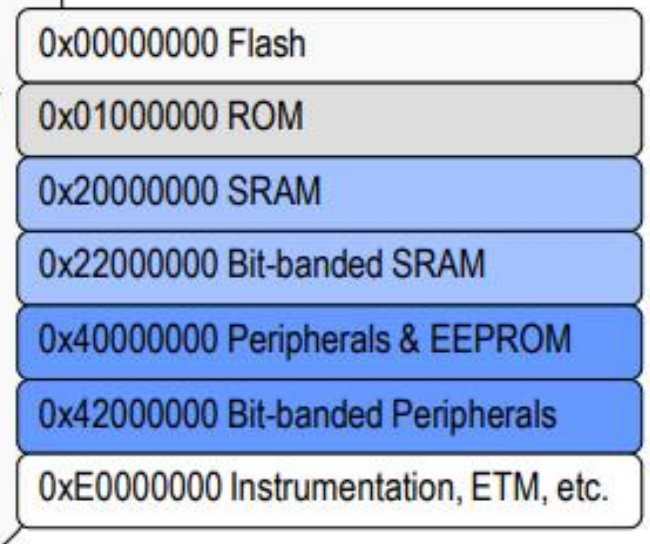
**32KB single-cycle SRAM with bit-banding**

**Internal ROM loaded with TivaWare software**

- Peripheral Driver Library
- Boot Loader
- Advanced Encryption Standard (AES) cryptography tables
- Cyclic Redundancy Check (CRC) error detection functionality

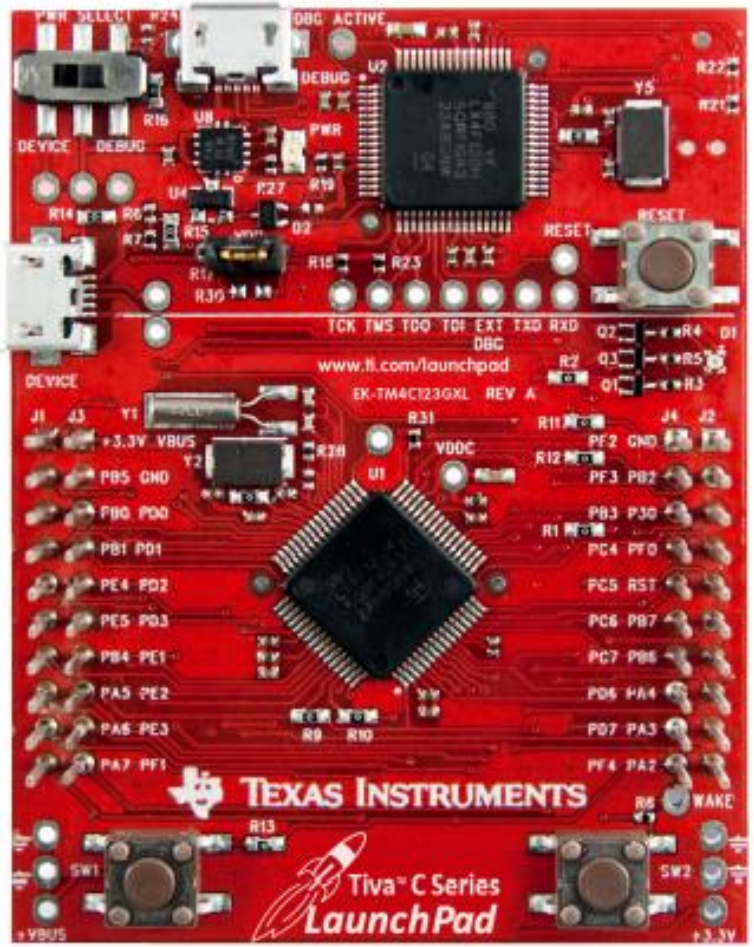
**2KB EEPROM (fast, saves board space)**

- Wear-leveled 500K program/erase cycles
- Thirty-two 16-word blocks
- Can be bulk or block erased
- 10 year data retention
- 4 clock cycle read time

# TIVA-C LAUNCHPAD

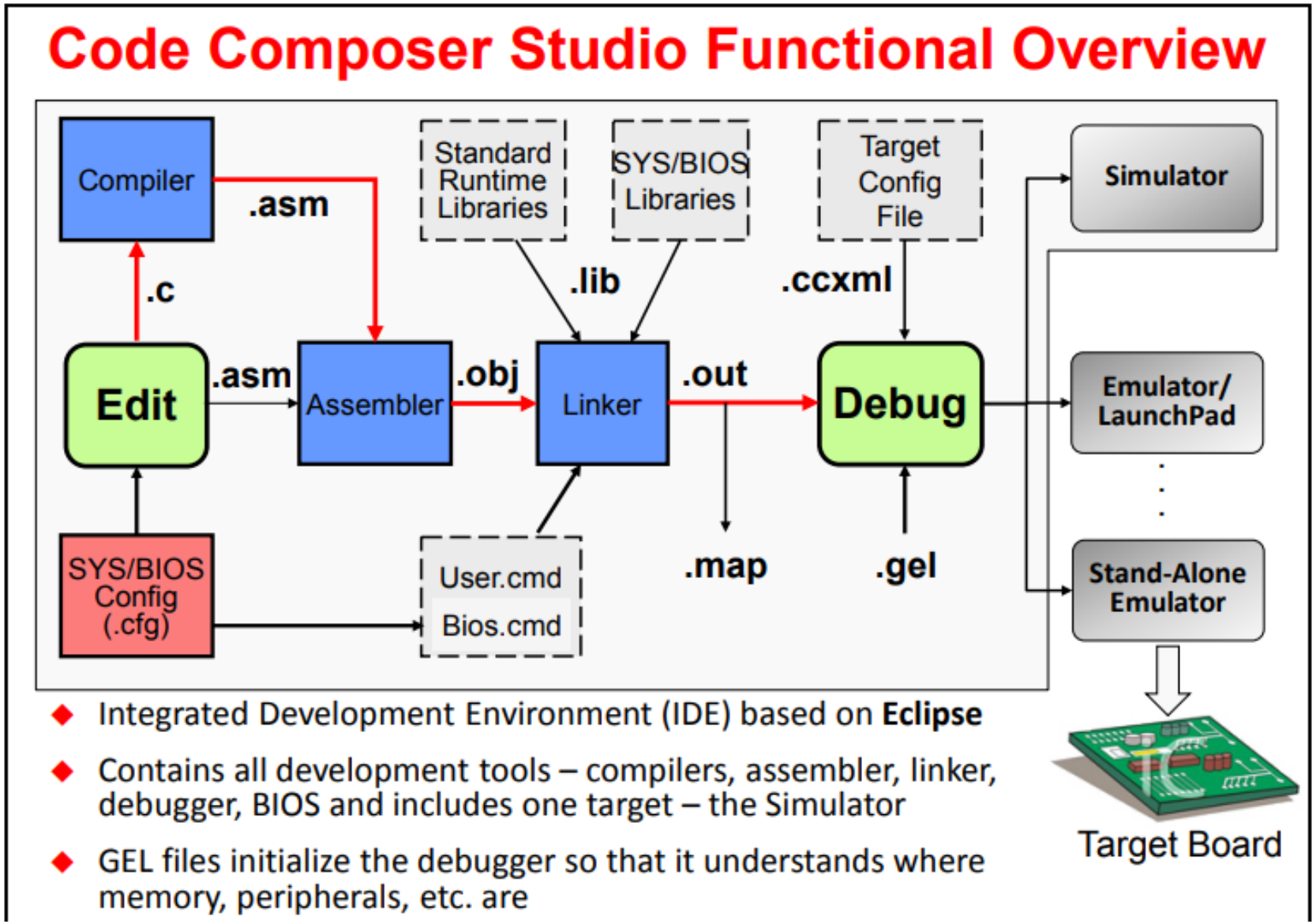
- ARM® Cortex™-M4F
  64-pin 80MHz TM4C123GH6PM
- On-board USB ICDI
  (In-Circuit Debug Interface)
- Micro AB USB port
- Device/ICDI power switch
- BoosterPack XL pinout also supports
  legacy BoosterPack pinout
- 2 user pushbuttons
  (SW2 is connected to the WAKE pin)
- Reset button
- 3 user LEDs (1 tri-color device)
- Current measurement test points
- 16MHz Main Oscillator crystal
- 32kHz Real Time Clock crystal
- 3.3V regulator
- Support for multiple IDEs:





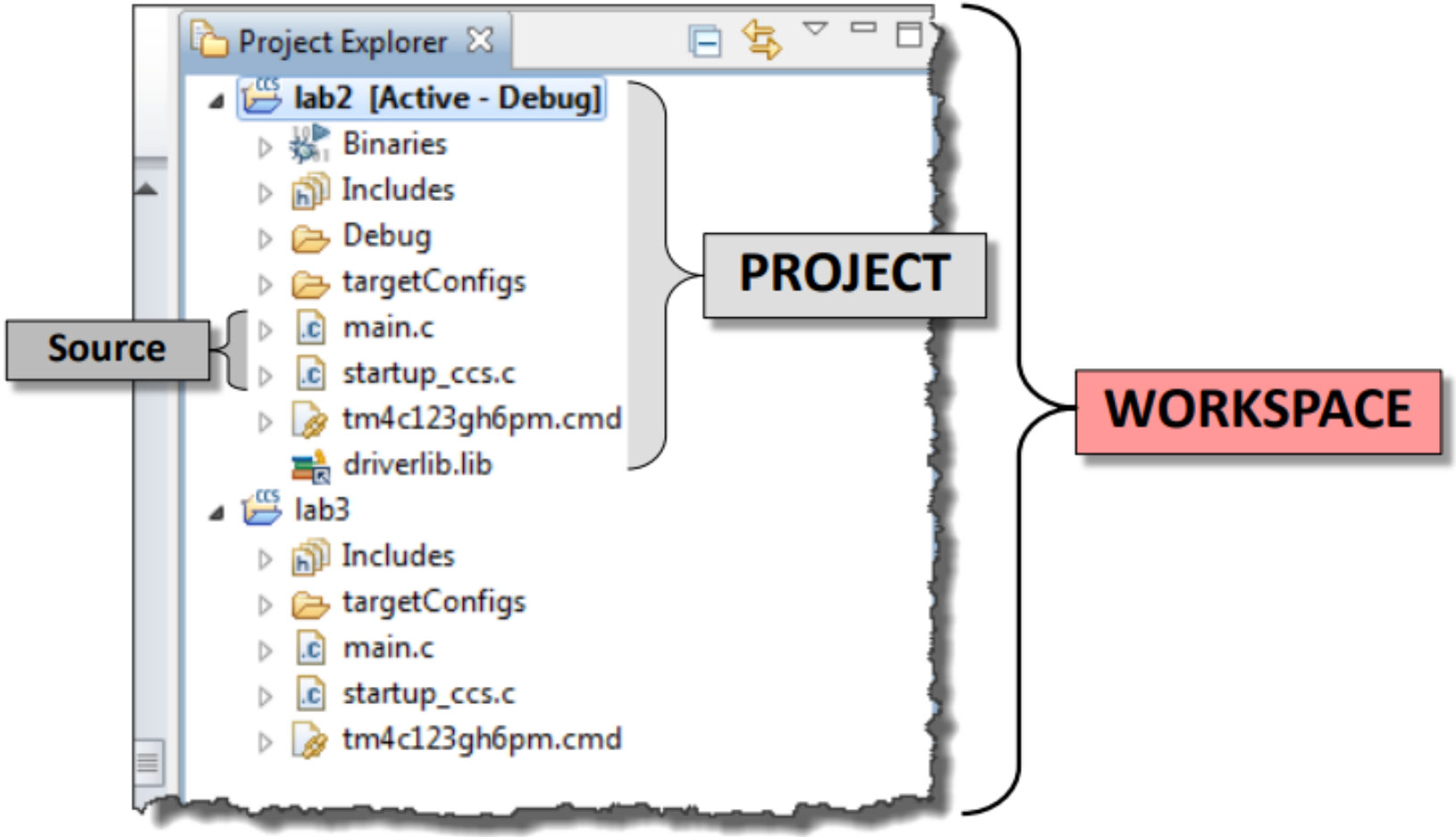
laxmeesha@ee.iitb.ac.in, dinesh@ee.iitb.ac.in

# CCS Overview



# CCS Overview

## Projects and Workspaces (viewed in CCS)

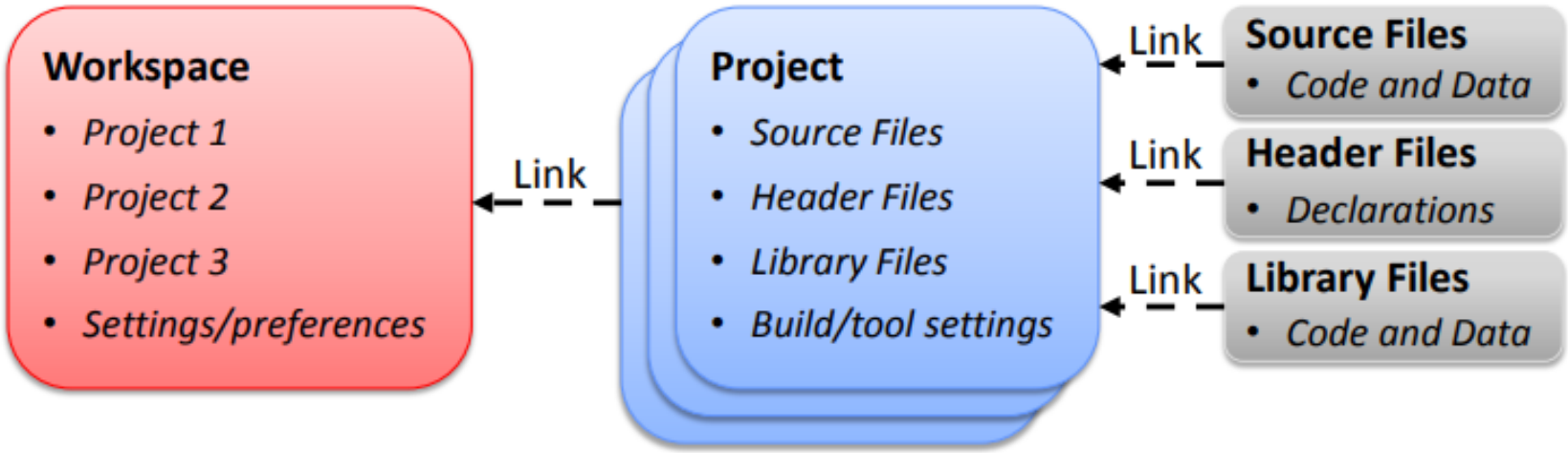




laxmeesha@ee.iitb.ac.in, dinesh@ee.iitb.ac.in

# CCS Overview

## Projects and Workspaces

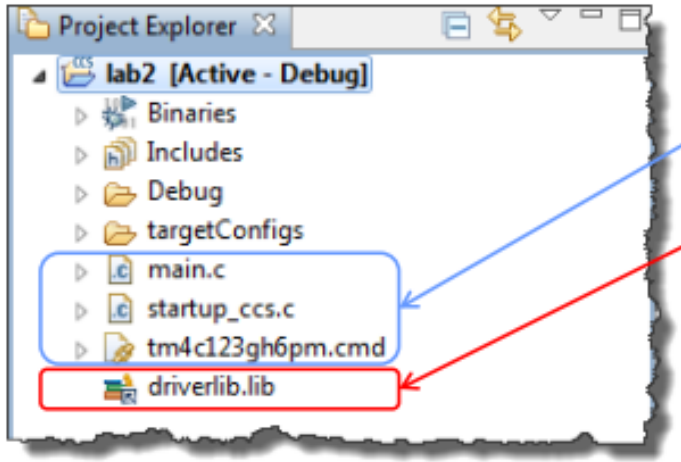



- WORKSPACE folder contains:
  - IDE settings and preferences
  - Projects can ***reside in*** the workspace folder or be ***linked*** from elsewhere
  - When importing projects into the workspace, linking is recommended
  - Deleting a project within the Project Explorer only deletes the link
- PROJECT folder contains:
  - Build and tool settings (for use in managed MAKE projects)
  - Files can be ***linked to*** or ***reside in*** the project folder
  - Deleting a linked file within the Project Explorer only deletes the link

# CCS Overview

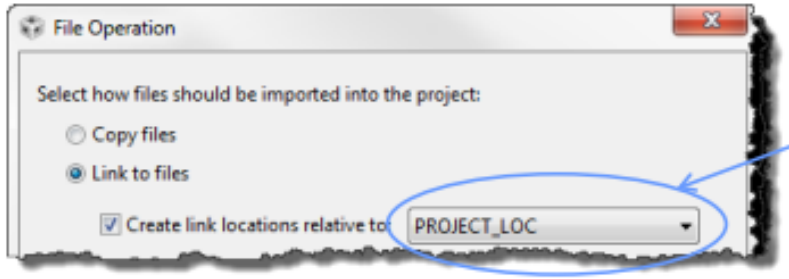
## Portable Projects

- **Why make your projects "portable"?**
  - Simplifies project sharing
  - You can easily re-locate your projects
  - Allow simple changes to link to new releases of software libraries




Copied files are not a problem (they move with the project folder)
Linked files may be an issue. They are located outside the project folder via a:

- absolute path, or
- relative path




This is the Path Variable for a relative path. This can be specified for every linked file.

# CCS Overview

## Path Variables and Build Variables

- **Path Variables**
  - Used by CCS (Eclipse) to store the base path for relative linked files
  - Example: PROJECT_LOC is set to the path of your project, say
    `c:/Tiva_LaunchPad_Workshop/lab2/project`
  - Used as a reference point for relative paths, e.g.
    `${PROJECT_LOC}/../files/main.c`

- **Build Variables**
  - Used by CCS (Eclipse) to store base path for build libraries or files
  - Example: CG_TOOL_ROOT is set to the path for the code generation tools (compiler/linker)
  - Used to find #include .h files, or object libraries, e.g.
    `${CG_TOOL_ROOT}/include` or `${CG_TOOL_ROOT}/lib`

- **How are these variables defined?**
  - The variables in these examples are automatically defined when you create a new project (PROJECT_LOC) and when you install CCS with the build tools (CG_TOOL_ROOT)
  - What about TivaWare or additional software libraries? You can define some new variables yourself

# CCS Overview

## Build Configurations

- Code Composer has two pre-defined BUILD CONFIGURATIONS:
  - *Debug* (symbols, no optimization) – great for LOGICAL debug
  - *Release* (no symbols, optimization) – great for PERFORMANCE
- Users can create their own custom build configurations
  - Right-click on the project and select *Properties*
  - Then click "*Processor Options*" or any other category: