Introduction to Embedded System Design

Programming the MSP430

Dhananjay V. Gadre
Associate Professor
ECE Division
Netaji Subhas University of Technology, New Delhi

Badri Subudhi
Assistant Professor
Electrical Engineering Department
Indian Institute of Technology, Jammu

Software Development Tools for the MSP430

- Texas Instruments Code Composer Studio IDE
- Energia IDE
- Third Party Development Environments (IAR)
- GCC Open Source Packages

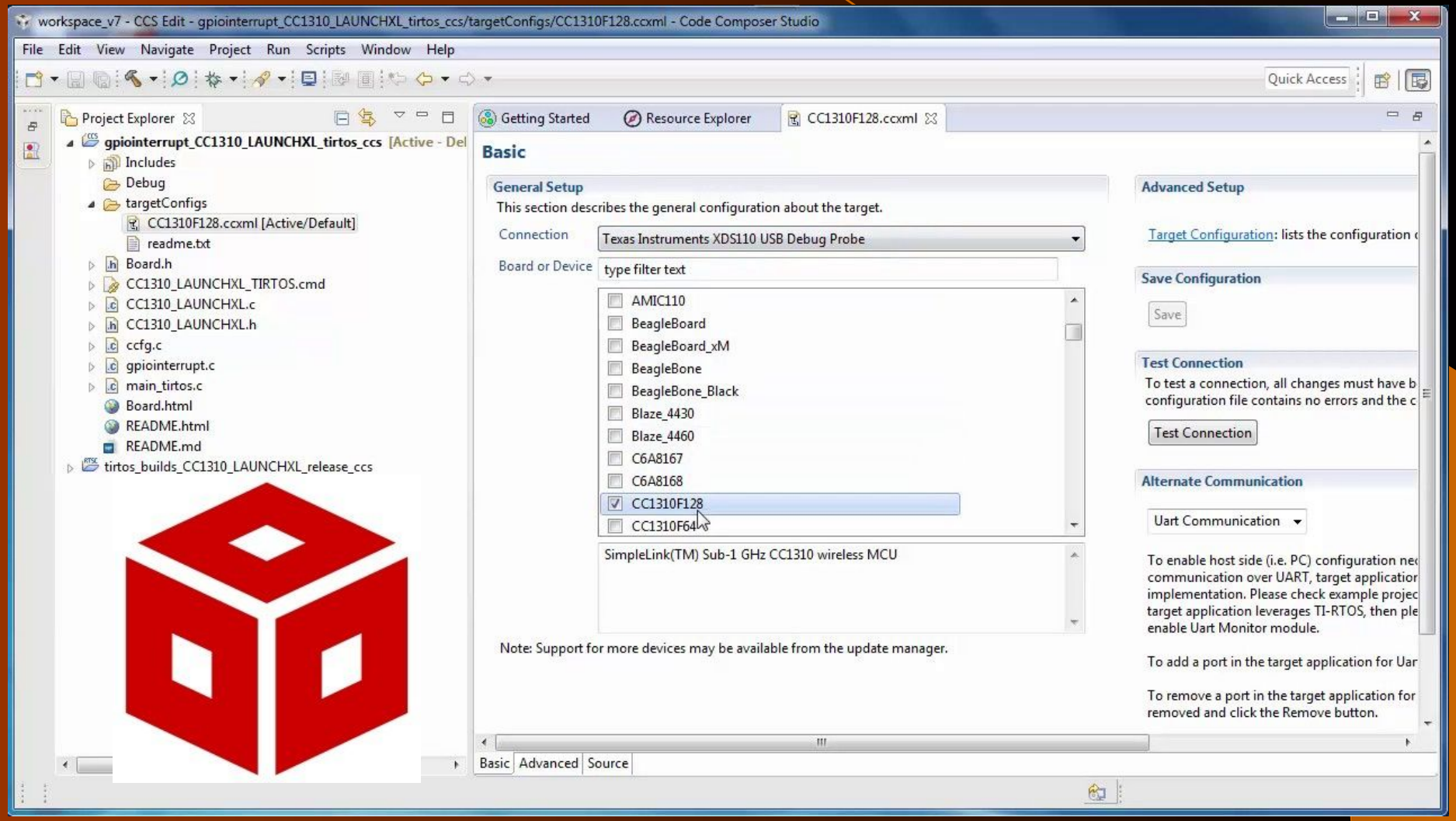
Code Composer Studio IDE

- Code Composer Studio (CCS) is the integrated development environment for TI's DSPs, microcontrollers and application processors. Code Composer Studio includes tools used to develop and debug embedded applications.
- It includes:-
    1. Compilers
    2. Source code editor
    3. Project build environment
    4. Debugger
    5. Simulators
- The intuitive IDE provides a single user interface taking you through each step of the application development flow.

Code Composer Studio IDE

- Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compelling feature-rich development environment for embedded developers.
- Supported on Mac OS, Windows, and Linux.
- E2E support forum from TI for resolving issues.

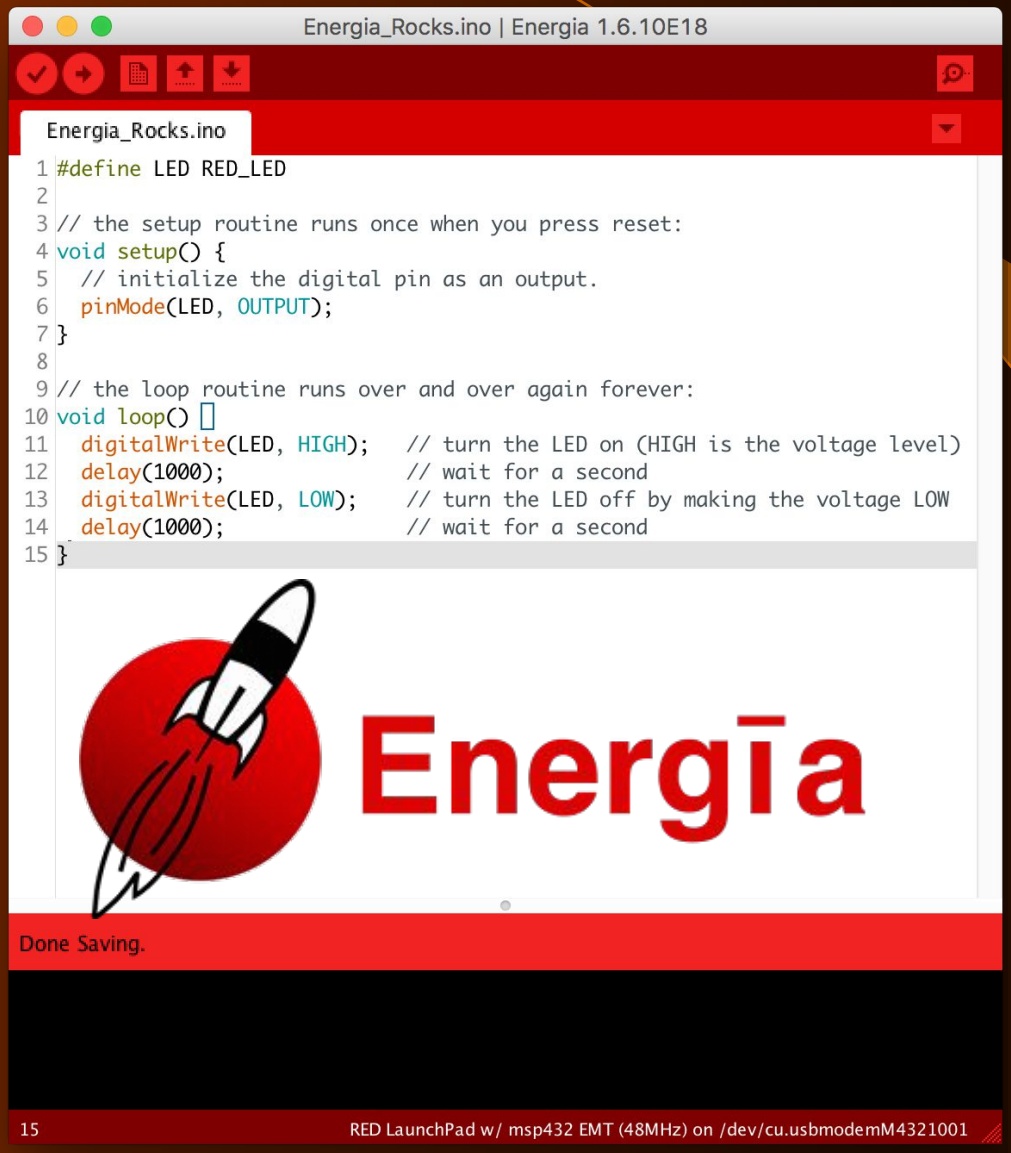
Code Composer Studio IDE



Energia IDE

- An open source & community-driven IDE and software framework. Based on the Wiring framework, Energia provides an intuitive coding environment as well as a robust framework of easy-to-use functional APIs & libraries for programming a microcontroller.
- Supported on Mac OS, Windows, and Linux.

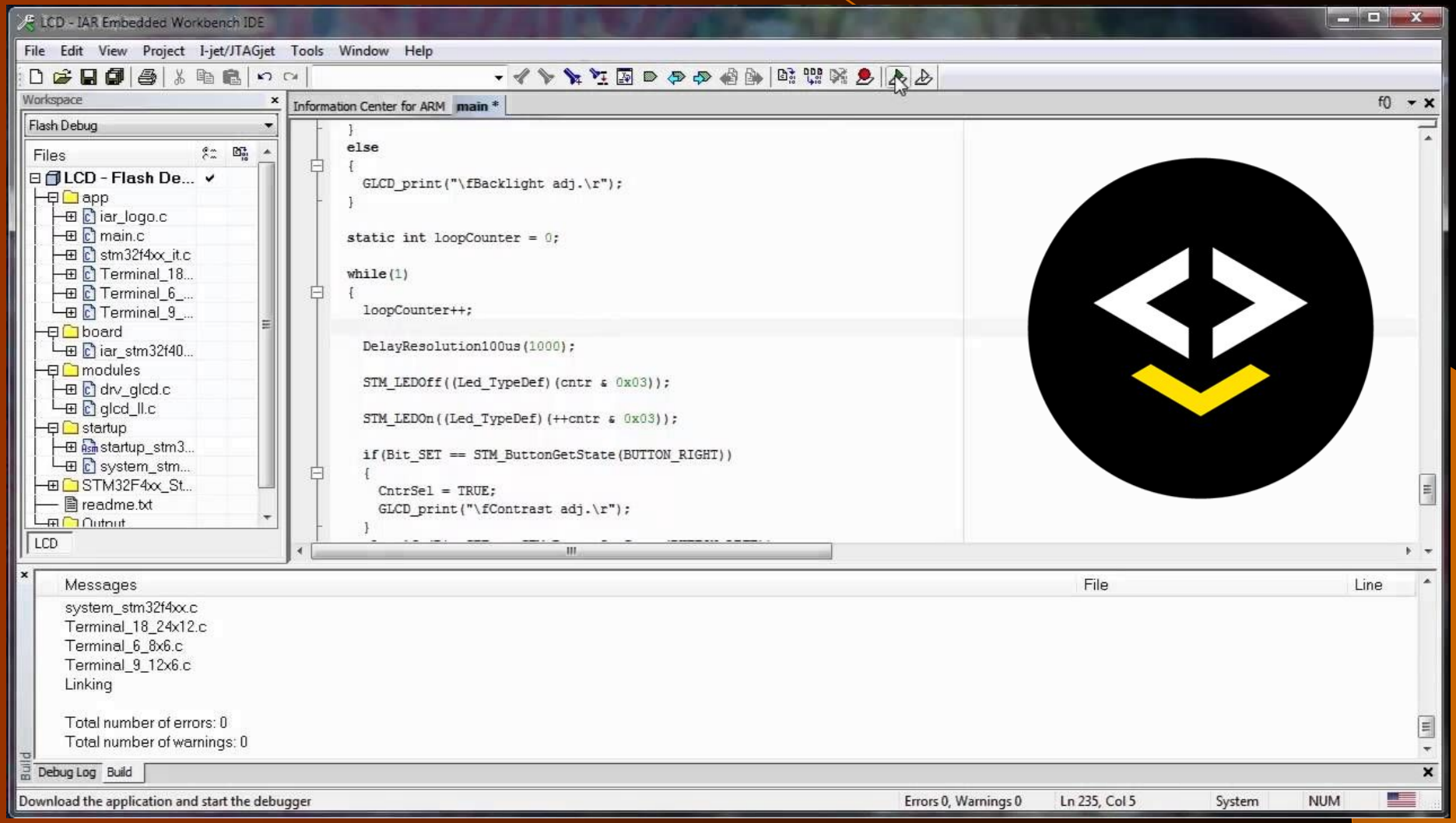
Energia IDE


Energia_Rocks.ino | Energia 1.6.10E18
Energia_Rocks.ino
#define LED RED_LED
// the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
pinMode(LED, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
Energīa
Done Saving.
15
RED LaunchPad w/ msp432 EMT (48MHz) on /dev/cu.usbmodemM4321001

IAR Embedded Workbench

- A complete debugger and C/C++ compiler toolchain for building and debugging embedded applications based on MSP430 MCUs. The debugger is fully integrated for source with support for complex code and data breakpoints.

IAR Embedded Workbench



GCC Open Source Packages

- Free, complete debugger and open source C/C++ compiler toolchains for building and debugging embedded applications using MSP430 MCUs without code size limitations. The compilers can be used standalone from the command-line or used within the Code Composer Studio IDE.

Program Download Mechanisms on the MSP430

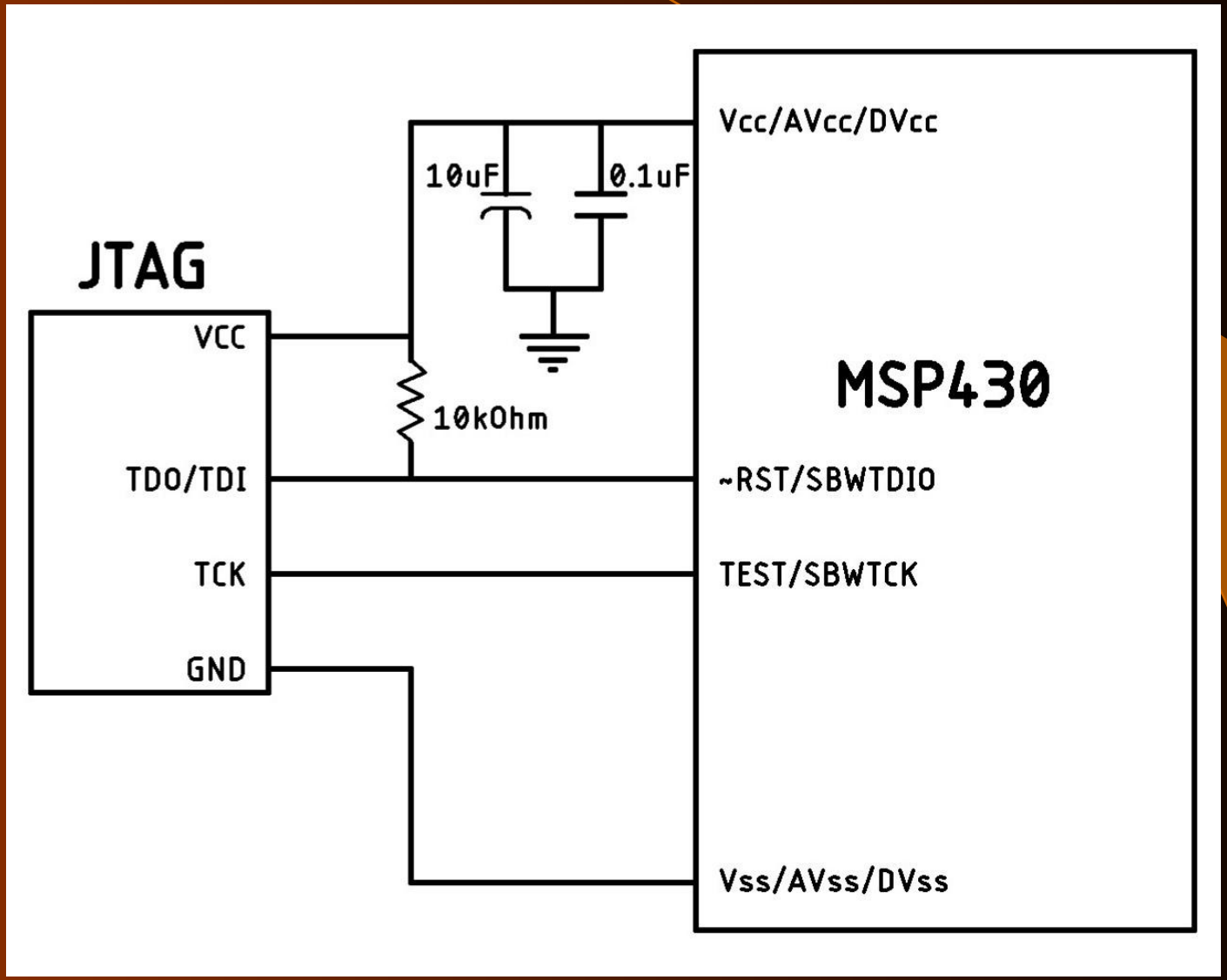
- Spy-Bi-Wire (SBW) using TEST/RESET
- JTAG Debugger
- UART/USB Bootloaders

SPY-BI-WIRE

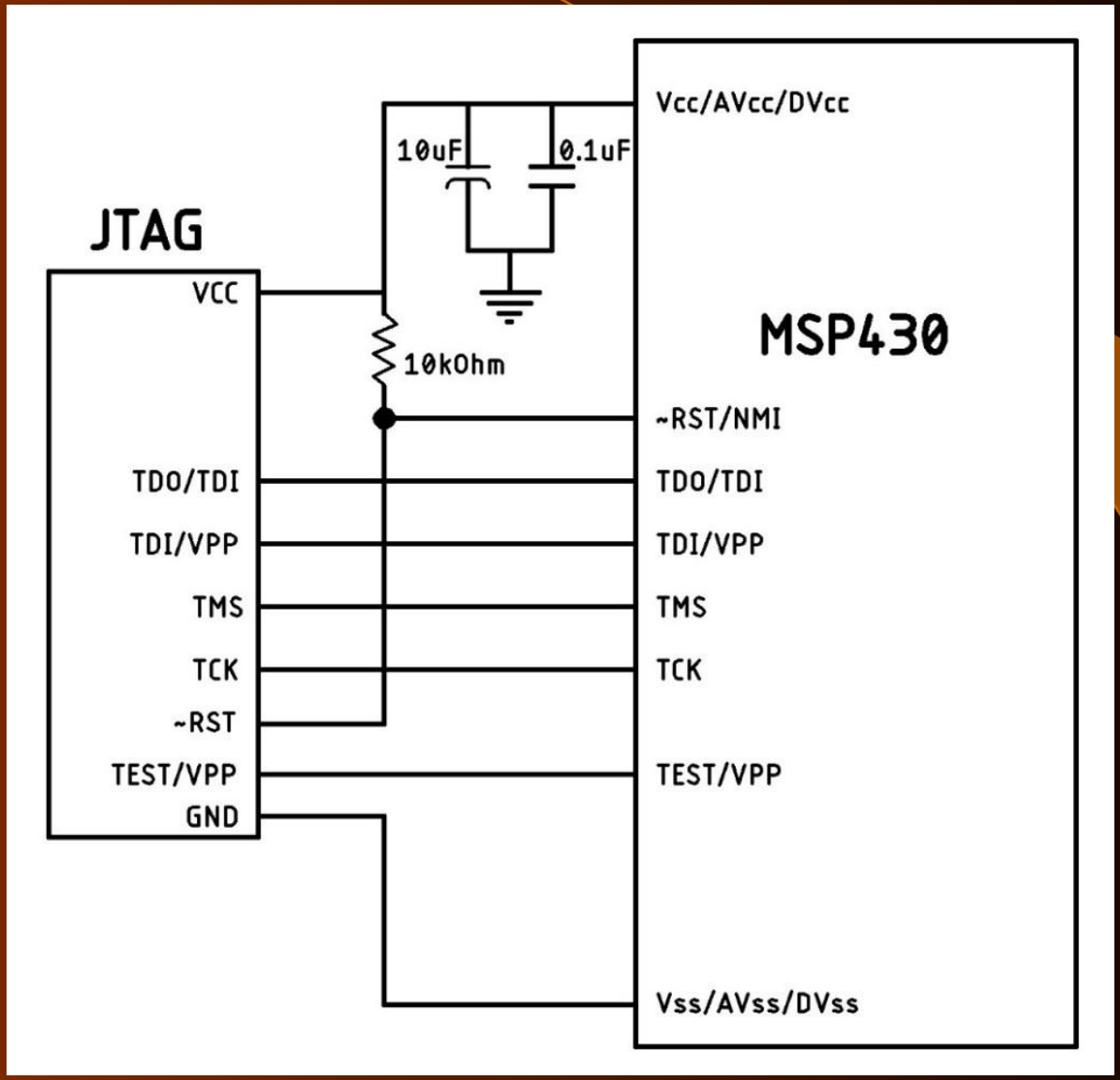
- Also known as 2 wire JTAG

PROS	CONS
• Only 2 wires used (TEST & RESET) • No overlapping with GPIO	• Slower than 4 wire JTAG

SPY-BI-WIRE


JTAG
VCC
TDO/TDI
TCK
GND
10uF
0.1uF
10kOhm
Vcc/AVcc/DVcc
MSP430
~RST/SBWTDIO
TEST/SBWTCK
Vss/AVss/DVss

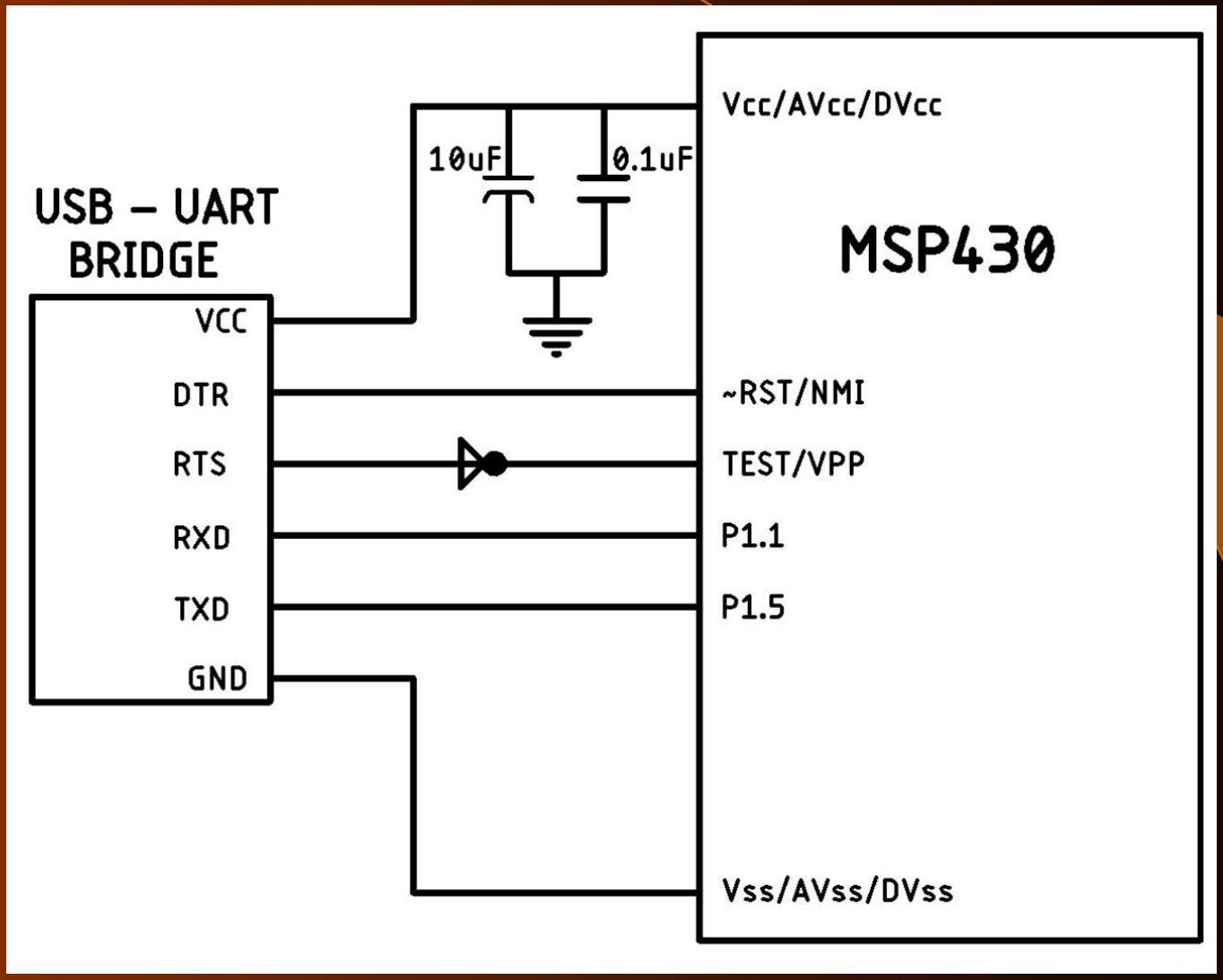
JTAG Debugger



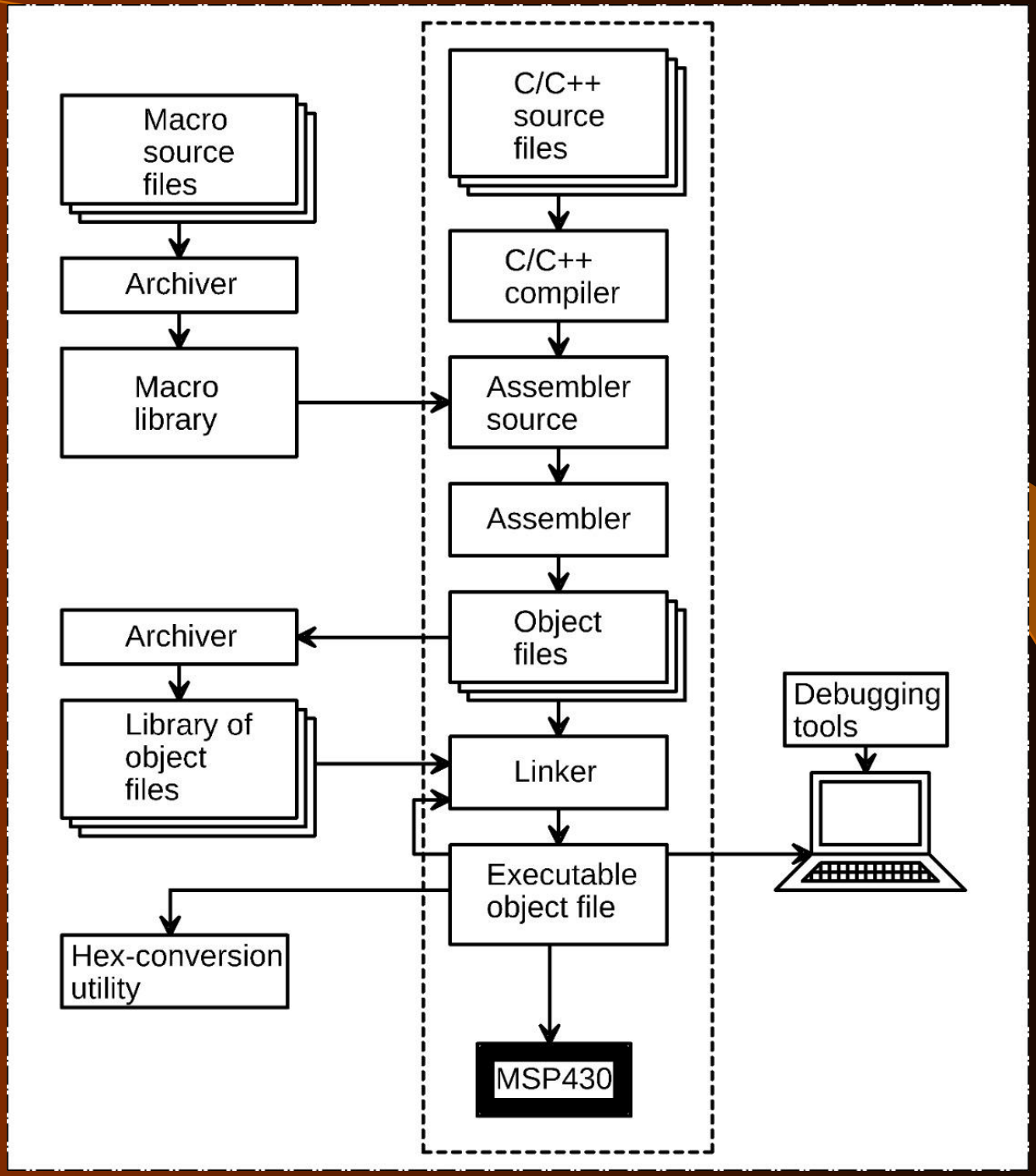
BSL

- The MSP430™ bootloader (BSL) (formerly known as the bootstrap loader)
- Allows users to communicate with embedded memory in the MSP430 microcontroller (MCU) during the prototyping phase, final production, and in service
- Both the programmable memory (flash memory) and the data memory (RAM) can be modified as required.

USB/UART Bootloaders



PROGRAM FLOW


Macro source files
Archiver
Macro library
C/C++ source files
C/C++ compiler
Assembler source
Assembler
Object files
Archiver
Library of object files
Linker
Debugging tools
Executable object file
Hex-conversion utility
MSP430

Writing Programs for the MSP430

- Assembly Language Programming
- Register-level Access using TI/GCC Compilers
- Using APIs and Third Party Libraries

Usage in this Course

- Code Composer Studio as Software Development Tool
- UART Bootloader as program download mechanism
- Register level and third party API/Libraries programming technique.

Why do we prefer high level language for programming microcontrollers?

Thank you!