Embedded Systems

AA 2020-2021

POLITECNICO DI MILANO





How to program a microcontrollers

- Role of the IDE, V 2.0

William FORNACIARI

Politecnico di Milano – Dipartimento di Elettronica e Informazione

Via Ponzio 34/5 – 20133 Milano, Italy

william.fornaciari@polimi.it

Microcontrollers

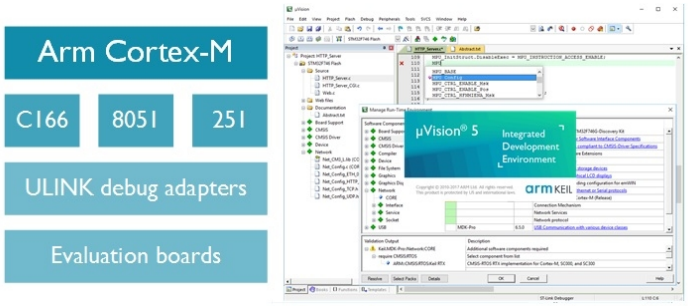
- Conceived to host the final application, not tailored to support the development of the code
- Limited computing capability/resources, poor/missing GUI
- Not X86 architecture and assembly (ARM, PIC, TI, …)

PC-based environments

- Advanced and user friendly GUI
- Powerful cores, wide storage, networking
- Development of code typically targets X86 architecture

Need of an IDE (Integrated development environment)

- Editing, versioning
- Cross compilation of the code
- Debugging and Profiling
- Link to the evaluation boards



Example of IDE - ISS

Few samples from the Embedded Systems arena

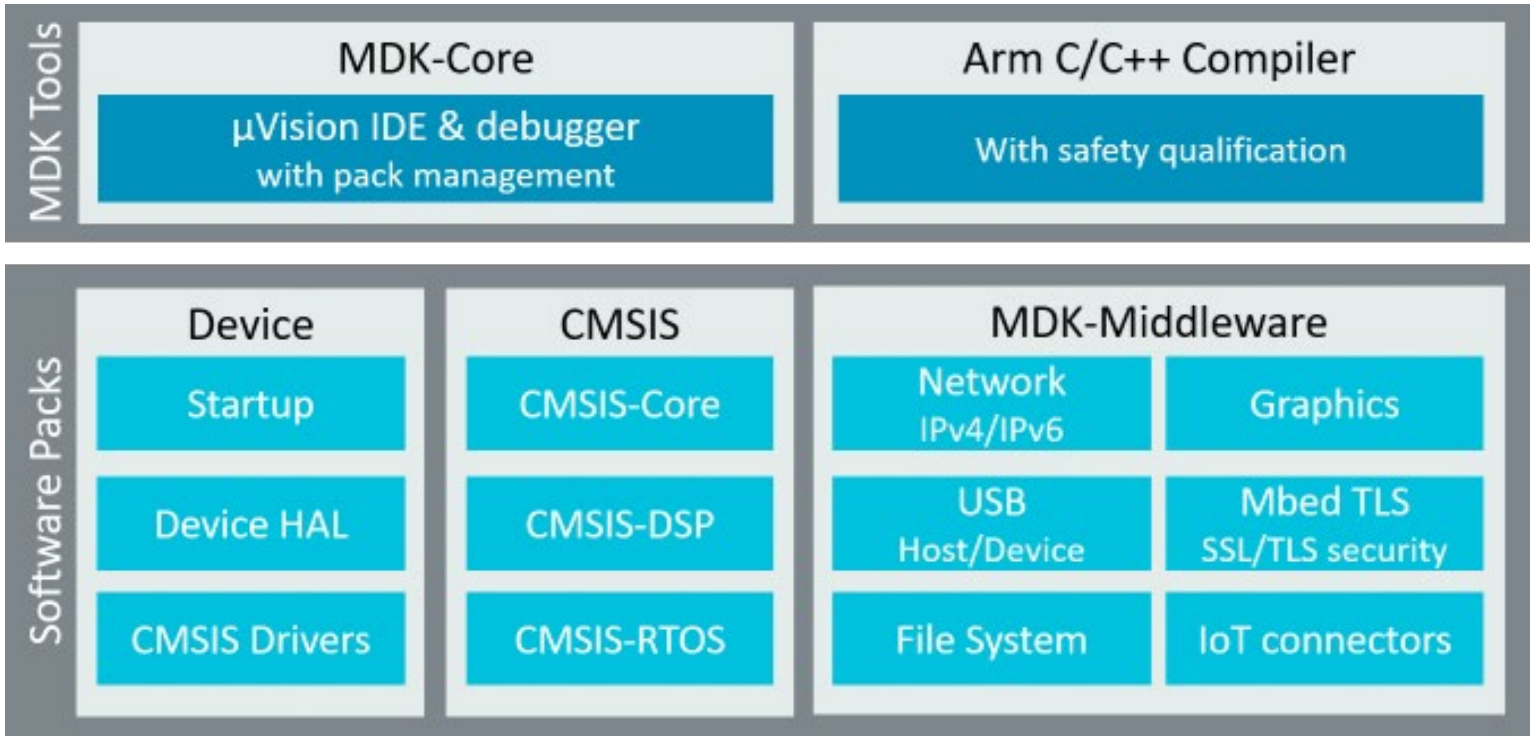
KEIL - Tigthly linked to ARM: https://www.keil.com/

Gem5 - Modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture. gem5 is a community led project with an open governance model: https://www.gem5.org/

- **Code Composer Studio -** Focus on TI products**:** https://www.ti.com/tool/CCSTUDIO
- **MPLAB - Microchip microcontrollers:** https://www.microchip.com/mplab/mplab-x-ide
- **CodeWarrior -** NXP, Freescale (ex **Motorola):** https://www.nxp.com/design/software/development-software/codewarrior-development-tools:CW_HOME
- **IAR Embedded Workbench -** https://www.iar.com/
- **Frequently linked to evaluation/development boards, some features depends on the family of MCUs**

Arm Keil MDK

For ARM microcontrollers, complete suite with many components



Not only a code editor

- IDE+Debugger, libraries, device support, templates, RTOS features

MDK-Professional

supports software development for

- Devices based on Arm Cortex®-M cores
- Secure and non-secure applications based on Arm Cortex-M23/M33 processors
- Legacy Arm7™ and Arm9™ devices
- Arm SecurCore®-based and Arm Cortex-R4-based devices
- Safety critical applications and projects that require long-term compiler support
- Next generation Cortex-M microcontrollers based on Armv8-M architecture
- Early prototyping with Fixed Virtual Platform simulation models
- Applications that require proven middleware components for Cortex-M devices
- IoT applications that require secure network connectivity to the cloud

MDK

Some of the main features

- **MDK** includes Arm C/C++ Compiler
  - with assembler, linker, and highly optimized run-time libraries that are tailored for optimum codesize and performance
- **Software Packs**
  - May be added any time to MDK-Core making new device support and middleware updates independent from the toolchain
  - They contain device support, CMSIS libraries, middleware, board support, code templates, and example projects
- **Cortex Microcontroller Software Interface Standard (CMSIS)**
  - Is a vendor-independent hardware abstraction layer for microcontrollers that are based on Arm® Cortex® processors
  - It defines generic tool interfaces and enables consistent device support. Its software interfaces simplify software re-use, reduce the learning curve for microcontroller developers, and improve time to market for new devices
  - CMSIS provides interfaces to processor and peripherals, real-time operating systems, and middleware components. It includes a delivery mechanism for devices, boards, and software and enables the combination of software components from multiple vendors

µVision IDE

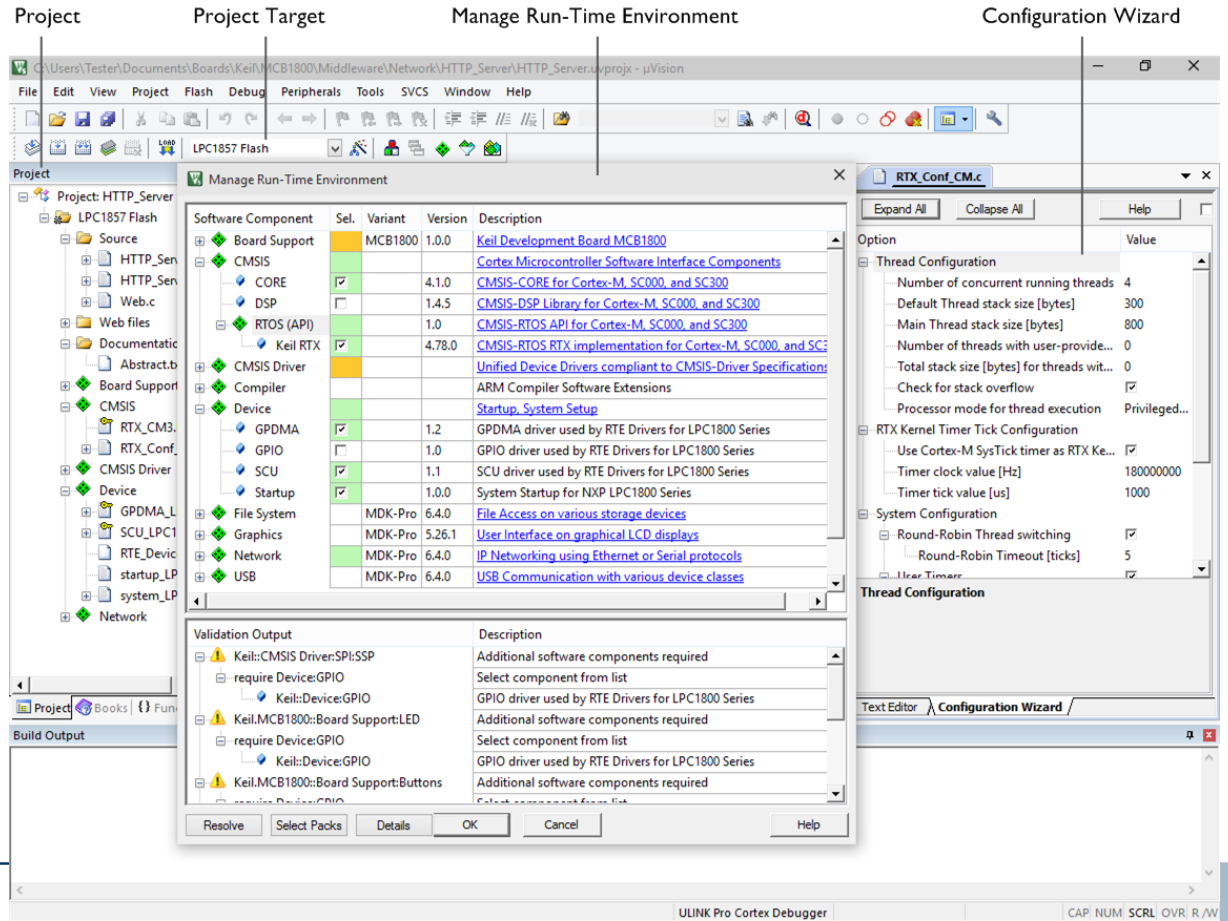
- Combines project management, run-time environment, build facilities, source code editing, and program debugging in a single environment. µVision accelerates embedded software development
- Supports multiple screens and allows to create individual window layouts anywhere on the visual surface

µVision Debugger provides a single environment

- where to test, verify, and optimize application code
- The debugger includes traditional features like simple and complex breakpoints, watch windows, and execution control and provides full visibility to device peripherals

µVision Project Manager and Run-Time Environment

- Create software application using pre-build software components and device support from Software Packs
- The software components contain libraries, source modules, configuration files, source code templates, and documentation



- Software components can be generic to support a wide range of devices and applications

Verification tools

Parasoft C/C++test

- Parasoft C/C++test provides a complete quality testing solution which improves software development team productivity and software quality for C and C++ applications.

Features

- Static code analysis and coding policy enforcement
- Automated code review with a graphical interface and progress tracking
- Automated unit and regression testing
- Host and target test execution
- Code coverage analysis with code highlighting
- C/C++test uses the high-speed streaming trace capabilities of the ULINKpro adapter to capture performance and code coverage information which can then be analyzed using the MDK-ARM development kit

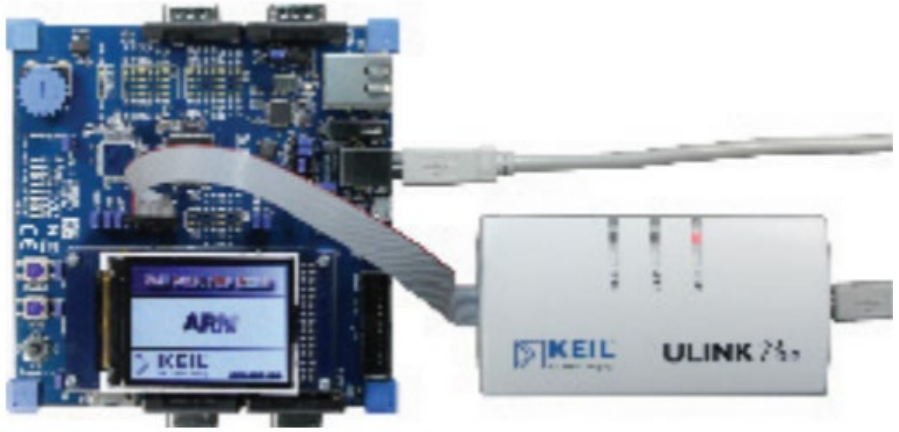
Verification tools

Parasoft …more than just a compiler

- Supports K&R C, ANSI C, and ANSI/ISO C++ standards
- Explicit support for MISRA C and MISRA C++ (Motor Industry Software Reliability Association) standards
- Value tracking of auto and static variables detects subtle initialization and value misuse problems
- Inter-function value tracking - powerful inter-statement value tracking crosses function boundaries
- Optional strong type checking (typedef-based) with a rich option set to detect nominal type differences
- User-defined semantic checking for function arguments and return values

ULINK Debug Adapters

- The ULINK™ family of USB Debug Adapters connect a PC's USB port to a target system (via JTAG or SWD)
- Allows developers to debug and analyze embedded programs executed on target hardware
- The ULINKpro provides unique streaming trace directly to a PC, enabling advanced analysis of your applications such as Execution Profiling and Code Coverage.



Features	ULINK*pro*	ULINK2
Run Control debug (ARM Cortex®-M series)	Yes	Yes
Memory + Breakpoint (while running)	Yes	Yes
Data Trace (Cortex-M3 and Cortex-M4)	Yes	Yes
Instruction Trace (Cortex-M3 and Cortex-M4)	Yes	-
Performance		
JTAG Clock speed	50MHz	10MHz
Memory read/write	1MByte/s	25KByte/s
Data Trace streaming (UART mode)	-	1Mbit/s
Data Trace streaming (Manchester mode)	100Mbit/s	-
ETM Trace streaming	800Mbit/s	
Analysis Tools		
Logic Analyzer	Yes	Yes
Performance Analyzer	Yes	-
Execution Profiler	Yes	-
Code Coverage	Yes	-

Fixed Virtual Platforms

Software Development Without a Hardware Target

- Running at speeds comparable to the real hardware
- FVP are complete simulations of an Arm system, including processor, memory and peripherals
  - More than just an Instruction Set Simulator (ISS)
- Provides a comprehensive model on which to build and test sw
  - start bare metal coding and Linux application development for Arm without the need for a physical target

Code Composer Studio (CCS)

Integrated Development Environment (IDE) – Texas Instruments

- Integrated development environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio
  - Code Composer Studio comprises a suite of tools used to develop and debug embedded applications
  - It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features
  - IDE provides a single user interface taking the programmer through each step of the application development flow
  - 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

Compiler for multi-core

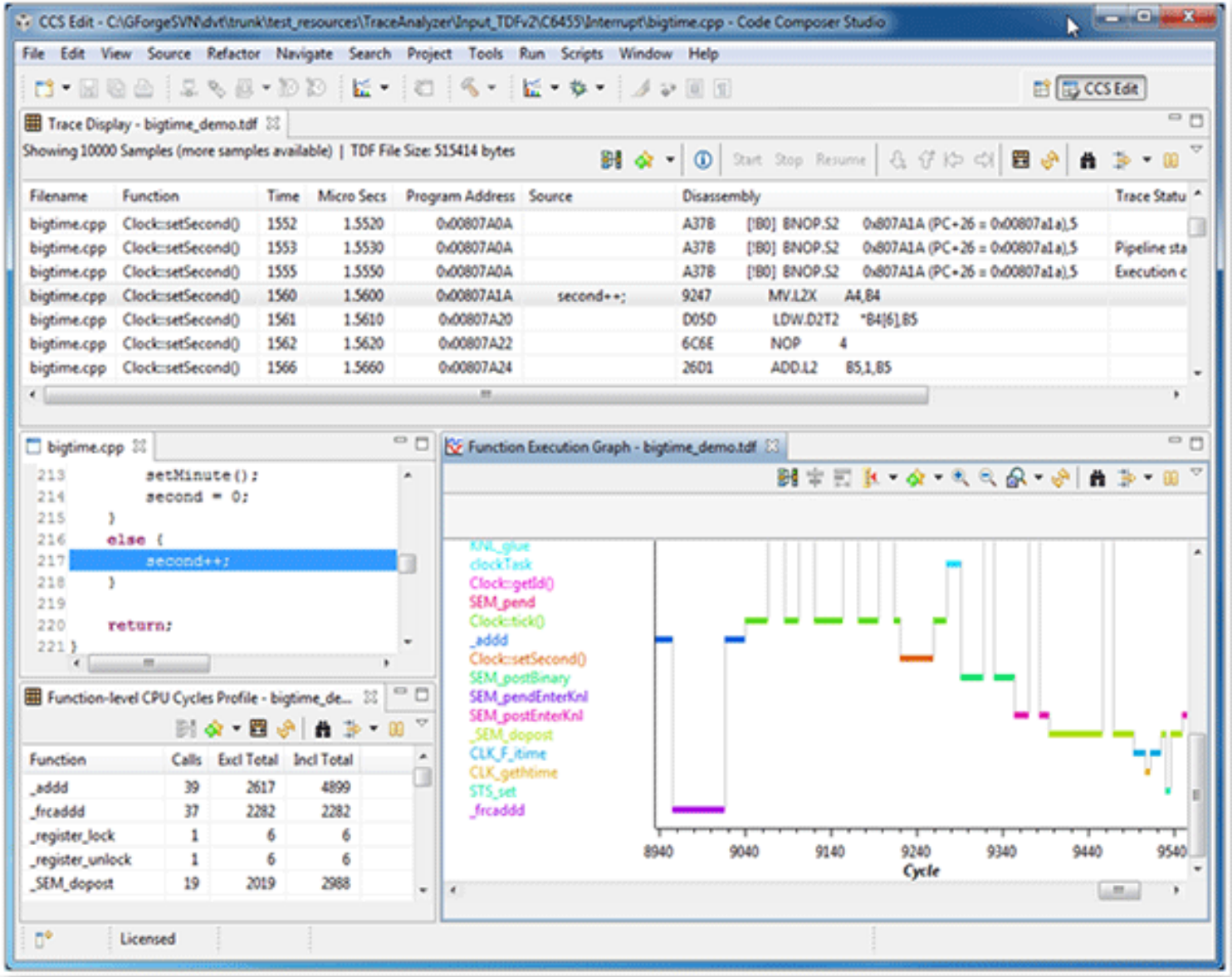
C6000 highly-optimizing C/C++ VLIW compiler

- Can take full advantage of the high performance processors like multicore DSPs
- Performs a wide variety of optimization techniques including automatically software pipelines inner loops, an extensive set of SIMD operations can speedup algorithms by up to 16x a large range of performance vs. code size options
- Quality of the code produced
  - The compiler is verified against industry-standard benchmarks (Plum Hall, Perennial, ..) and several validation suites
  - Continual evaluation of the compilers performance on thousands of benchmarks, ensures performance stability of compiled code across releases

Processor Trace

- Many high performance TI processors include the ability to perform processor trace
- Trace provides detailed, historical account of code execution, timing and data accesses
- This advanced capability is extremely useful in detecting complex, intermittent bugs as wells as profiling to help fine tune code performance
- Trace data can be captured to dedicated on-chip memory (ETB) or exported over pins to be captured by a trace receiver

Processor Trace



System Analyzer

System Analyzer provides visibility into the application, OS and hardware across the system at any given time by correlating software and hardware instrumentation from multicores on the same timeline

System Analyzer is made up of two core components

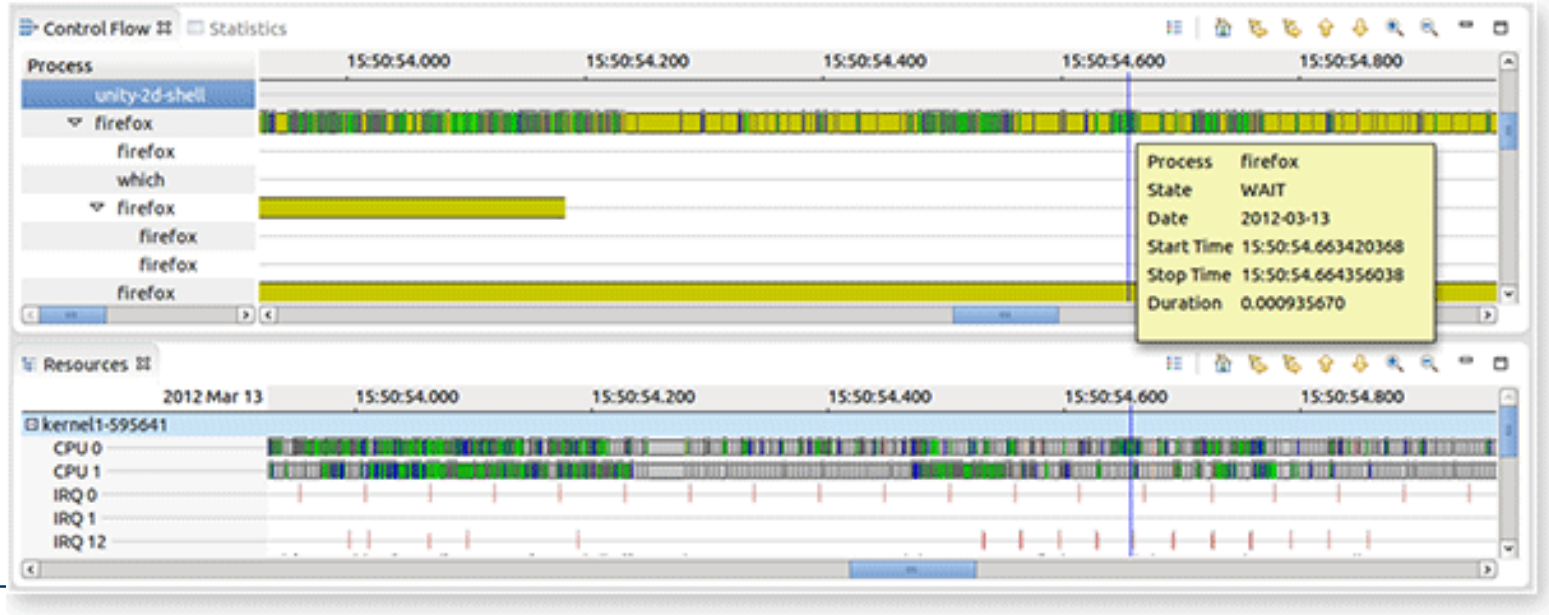
- UIA (Unified Instrumentation Architecture): a software package for logging, runtime control and data movement
- Analysis Displays: Tooling for runtime control, data collection, data decoding, data analysis and data visualization

System Analyzer can obtain data in a number of ways

- It can be captured to the on chip embedded trace buffer (ETB), it can be streamed off the device using System Trace via an XDS560v2 System Trace Receiver or you can have the application continually drain the ETB and push the data out over Ethernet

Linux Development

- Code Composer Studio supports both Linux kernel and application level development
- The kernel can be debugged via JTAG or use GDB for application development. By installing the Linux Development Tools via the App Center additional functionality such as the Linux Trace Tools (LTTng) can be exploited



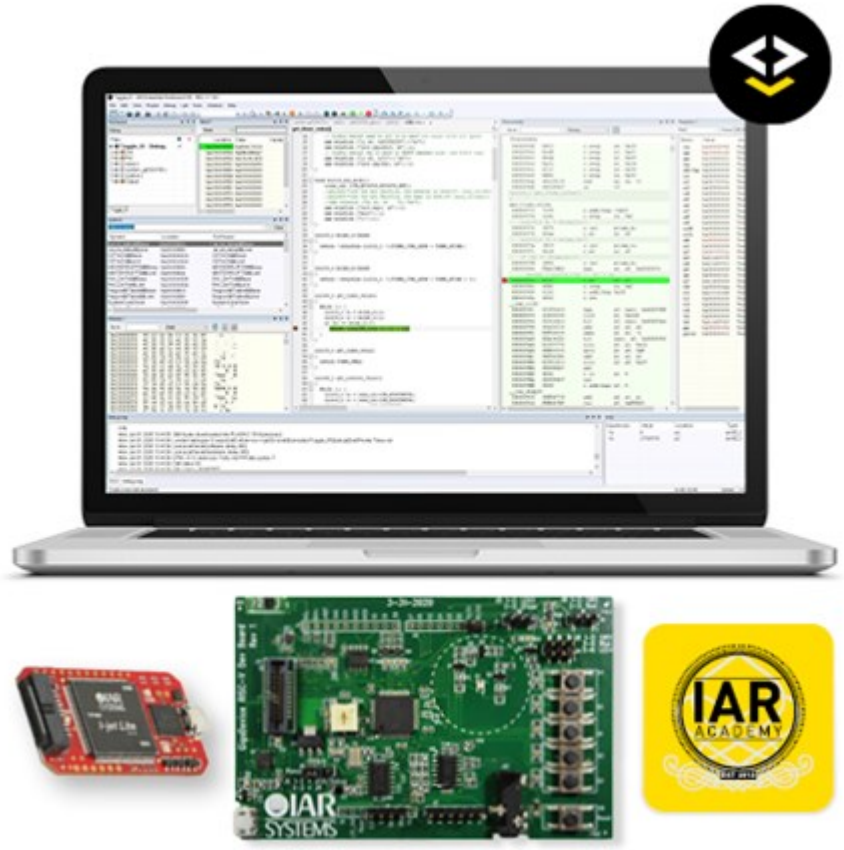
IAR Systems

Encompassing multiple platforms – different approach

- IAR Embedded Workbench development toolchain
- Flexible and complete platform for all aspects of embedded software development with powerful functionality. It supports the entire development process
- Different business model w.r.t. to the previous ones
  - Targeting multiple architectures
  - Dedicated team of experts also available
- More info https://www.iar.com/

Evaluation Kit

- IAR Embedded Workbench fo RISC-V, 30-day evaluation license, including the static analysis tool C-STAT
- RISC-V GigaDevice Evaluatio Board
- I-jet Lite debug probe
- IAR Academy on-demand course Introduction to RISC-V Evaluation Kit, available from the self-learning online portal IAR Academy On Demand



Technical details of the board in the kit

Just an example of a full SDK

- GD32VF103RBT6 RISC-V GigaDevice
- User LEDs
- User SWs
- 3 Axis Accelerometer
- Potentiometer
- Temperature and Humidity Sensors
- Light Sensor
- Onboard Microphone
- iPhone compatible Earbud Jack
- SPI Flash Memory
- Reset button
- JTAG connector 20 pin 0.05”
- USB micro B connector for USB-Serial converter
- Power up through USB connector or through I-jet Lite (pin 11/13 JTAG connector)

Do not forget that…

Linux is still and (forever) existing !

- Wide a wide range of portable tools and analysis environment
  - See lessons at the beginning of the course
- Update to support MCUs and pheriferals, drivers, etc for embedded is slower than X86
  - It is a «community» and apparently «no-profit» activity