MDK-ARM

Microcontroller Development Kit
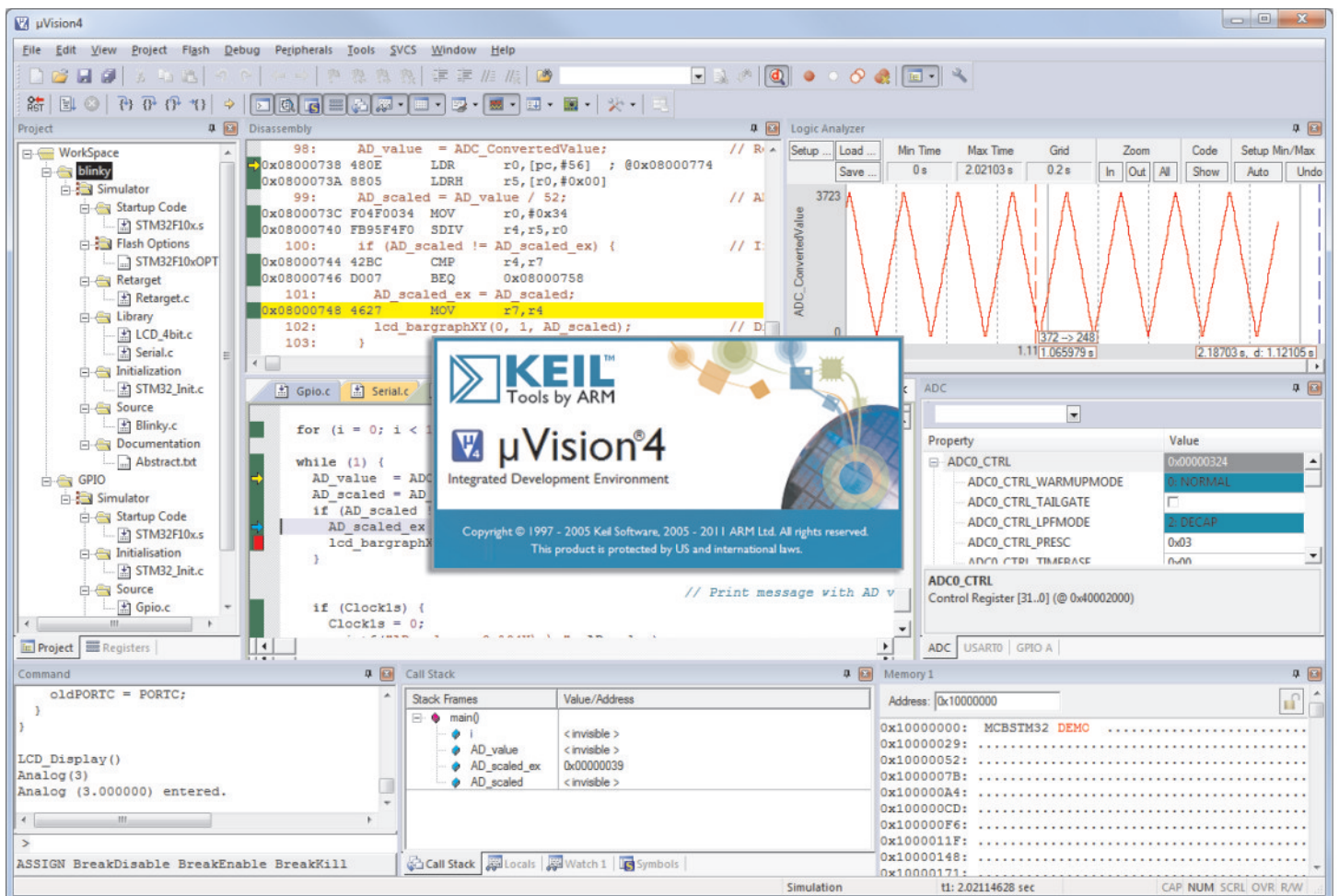
KEIL™
Tools by ARM

The **MDK-ARM** (Microcontroller Development Kit) is the complete software development environment for ARM7™, ARM9™, Cortex™-M, and Cortex-R4 processor-based devices.

MDK is specifically designed for microcontroller applications and combines the ARM C/C++ Compiler with the Keil RTX real-time operating system and middleware libraries. All tools are integrated into µVision which includes project management, editor and debugger in a single easy-to-use environment.

The fully integrated ARM C/C++ Compiler offers significant code-size and performance benefits to the embedded developer, however, MDK can also be used with the GNU GCC Compiler.

The Keil RTX is a deterministic real-time operating system with small memory footprint. MDK-Professional includes a flexible File System and Middleware for TCP, USB, and CAN peripherals. The TCP Networking Suite offers multiple internet protocols and provides various applications such as HTTP, Telnet, and FTP server. USB Device and USB Host stacks are provided with standard driver classes. Numerous application examples help you to get quickly started with application development.

MDK works with several debug adapters. ULINK*pro* enables on-the-fly analysis of running applications and records every executed Cortex-M instruction. It delivers time profiling and complete code coverage for applications optimization and certification.

MDK-Professional integrates all development tools into the µVision IDE/Debugger, and includes a comprehensive middleware collection.

ARM®

µVision IDE

Document outlining gives you a better overview of your source code.

Context menus provide flexible configuration of all aspects of your project.

Each project may contain multiple target configurations.

The **Project Workspace** lists source and header files helping organize your project

Double-click on messages to locate the erroneous line in the source code.

The **Configuration Wizard** simplifies editing setup files.

The µVision IDE incorporates a **Device Database** of supported ARM-Powered microcontrollers. In µVision projects, required options are set automatically when you select the device from the Device Database. µVision displays only those options that are relevant to the selected device.

The **Flexible Window Management System** enables you to drag and drop individual windows anywhere on the visual surface. This interface allows you to make better use of your screen space and to organise multiple windows efficiently.

The **Editor** provides an optimized workflow with intuitive toolbars providing quick access to editor functions. Editor functions are also available in the debug mode allowing easy source code navigation and editing.

The integrated **Source Browser** provides access to all application symbols, together with name, type, and class information. It allows you to instantly navigate to the definition and references of any symbol.

µVision Debugger

The µVision **Debugger** can be configured as a Target Debugger or as a Simulator. It provides a single environment in which you may test, verify, and optimize your application.

The Debugger also simulates many ARM-Powered MCUs including their instruction set and on-chip peripherals.

Debug Windows

The Debugger provides windows and dialogs to help you monitor and control your system. These include:

- **Memory Window** - review and modify memory contents.
- **Watch Window** - view and modify program variables and lists the current function call nesting.
- **Symbol Window** - view debug symbol information of the application program.
- **Disassembly Window** - synchronized with the Source Windows making program debugging easier.
- **Register Window** - view and change register contents.
- **Call Stack Window** - view current call nesting including variable values.
- **Breakpoints Window** - define watchpoints and complex Execution, Access, and Conditional breakpoints.
- **Browse Window** - search for objects in your code.

System Viewer

The **System Viewer** windows display peripheral registers that the processor can read and write to. They display the state, content, and name of peripheral registers. Content values are instantly updated by the target hardware as changes occur. Values can also be changed at run-time by typing a new value from within the System Viewer window.

Analysis Tools

The advanced analysis tools work with the simulator or with target hardware via the ULINK*pro* streaming trace adapter.

The configurable **Logic Analyzer** provides a graphical display of signals and variables. You may click on variable changes to display the instructions that caused that change in the source code editor window.

The Debugger provides **Code Coverage** statistics to verify applications that require certification testing and validation. Color coding highlights the execution status of instructions helping you to refine your testing.

The **Performance Analyzer** displays the execution time recorded for functions in your application. Bar graphs display the time spent in a function, and the number of calls to it.

The **Execution Profiler** records execution statistics for each CPU instruction, including the execution count and execution time for each instruction. These can be reviewed in the editor and disassembler windows.

ULINK Debug Adapters

The ULINK family of USB Debug Adapters connect your PC's USB port to your target system (via JTAG or SWD), allowing you to debug and analyze embedded programs running on target hardware.

The ULINK*pro* provides unique streaming trace directly to your PC, enabling advanced analysis of your applications such as Execution Profiling and Code Coverage.
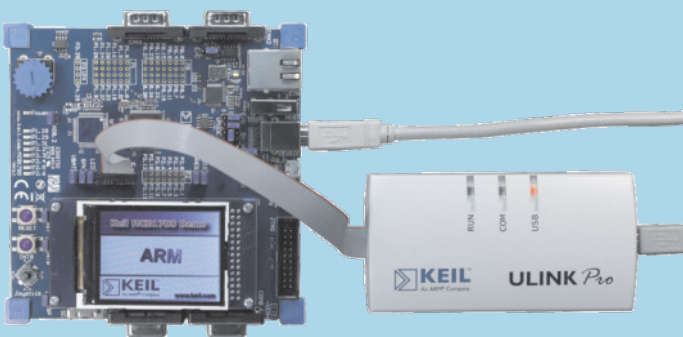
Features	ULINK*pro*	ULINK2
Run control debug (ARM & Cortex-M)	Yes	Yes
Memory + Breakpoint (while running)	Yes	Yes
Data Trace (Cortex-M3/M4)	Yes	Yes
Instruction Trace (Cortex-M3/M4)	Yes	-
Performance		
JTAG Clock speed	50MHz	10MHz
Memory read/write	1MByte/s	25KByte/s
Data Trace streaming (UART mode)	-	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	-

Visit: **www.keil.com/ULINK**

Target Debugging and System Analysis

Cortex-M CoreSight

All Cortex-M based devices feature the ARM CoreSight™ technology with advanced debug and trace capabilities.

MDK-ARM, together with a ULINK adapter, uses these features to enable you to debug your program. You are able to:

- Read/write memory and peripheral registers on-the-fly, while your program is running at full-speed.
- Set up to **8** breakpoints while the processor is running.
- Control the CPU allowing program start/stop.
- Single Step source or assembler lines.

Data and Event Trace

All Cortex-M3 and Cortex-M4 devices provide data and event trace. MDK-ARM provides a number of ways to analyze this information while your system is running:

- **Trace Window** - displays program flow by capturing timestamps, PC samples, and Read/Write accesses.
- **Debug Viewer** - displays the Instrumented Trace (ITM) output in a terminal window.
- **Exceptions window** - displays statistical information about program exceptions and interrupts.
- **Event Counters** - display real-time values of specific event counters providing performance indications.
- **Logic Analyzer** - graphically displays variable changes in captured data trace.
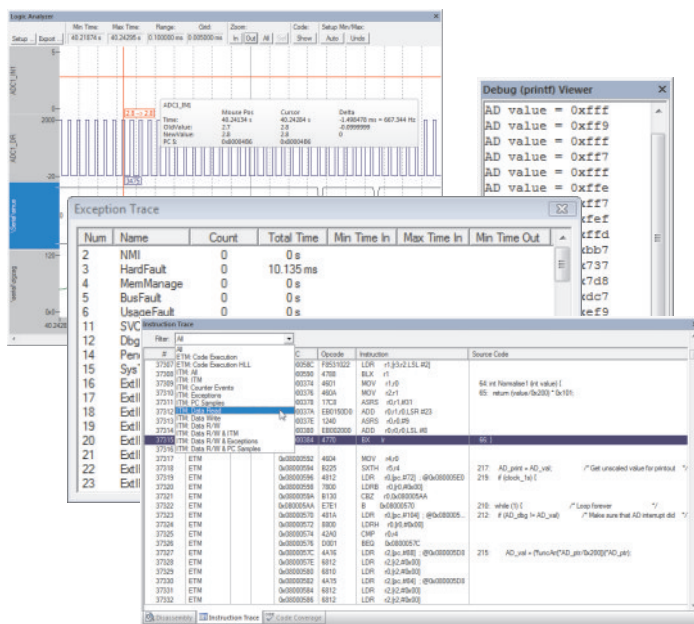
Data Trace Windows provide information from the running target for program data, exceptions, variables, and printf-style outputs

InstructionTrace

Cortex-M devices with ETM provide instruction trace. The Keil ULINK*pro* streams instruction trace directly to your PC. This enables debugging of historical sequences, execution profiling, and code coverage analysis.

The virtually unlimited stream of trace information enables MDK-ARM to provide complete **Code Coverage** of your program. Code coverage identifies every instruction that has been executed, ensuring thorough testing of your application. This is an essential requirement for complete software verification and certification.
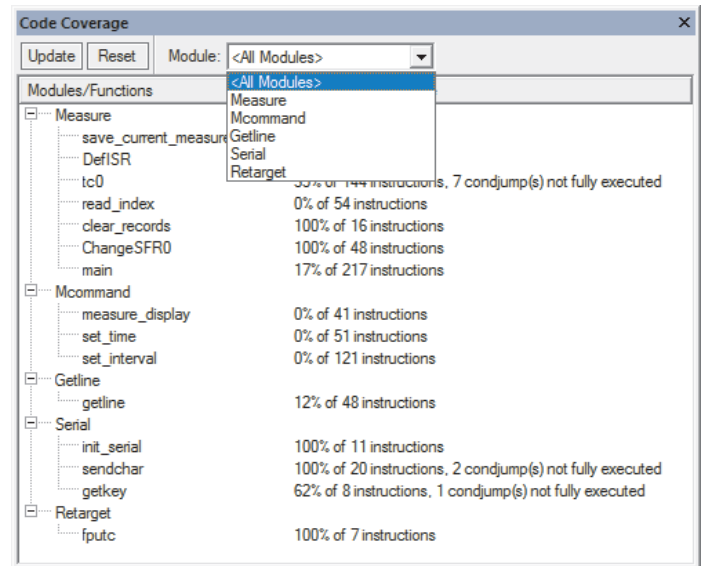
Code Coverage shows the percentage of instructions that have executed.

ULINK*pro* allows applications to be run for long periods of time while collecting trace information. This can be used by the µVision **Execution Profiler** and **Performance Analyzer** to identify program bottlenecks, optimize your application, and to help locate defects.
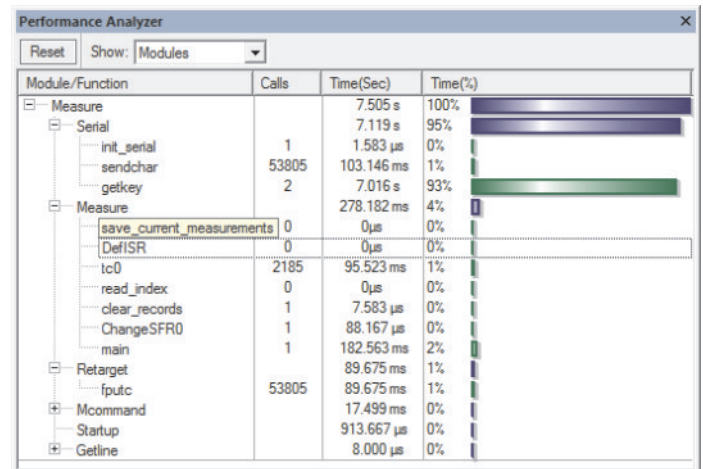
The performance analyzer displays time spent in each part of your program.

RTX Real-Time Operating System

RTX Real-Time Operating System

Today, microcontroller applications often require simultaneous execution of multiple tasks in a real-time environment.

While it is possible to implement an embedded program without using a real-time kernel, the proven Keil RTX allows you to focus on application development, enabling you to save time, and produce a more reliable, expandable system.

RTX is a royalty-free, real-time operating system specifically developed for the ARM and Cortex-M feature-sets. RTX provides features to manage system resources:

- Applications separated into independent tasks.
- Extensive time control (scheduling, time delay/intervals).
- Deterministic execution times and task scheduling.
- Inter-task communication, resource sharing, and memory allocation features with message pools.
- Supports development with error checking, debug and test facilities.

RTX source code is provided in all MDK-ARM Editions.

Visit www.keil.com/rl-arm/kernel.asp for more information.

RTOS Aware Debugging

RTX is fully integrated in the µVision **Debugger** making it easy to monitor task status and kernel activity.

The **RTOS-aware** dialog is available in simulation and also when running on target hardware. It displays information about all aspects of the kernel and the running tasks. This enables you to view statistics about the active tasks, stack loading, and system resource usage.
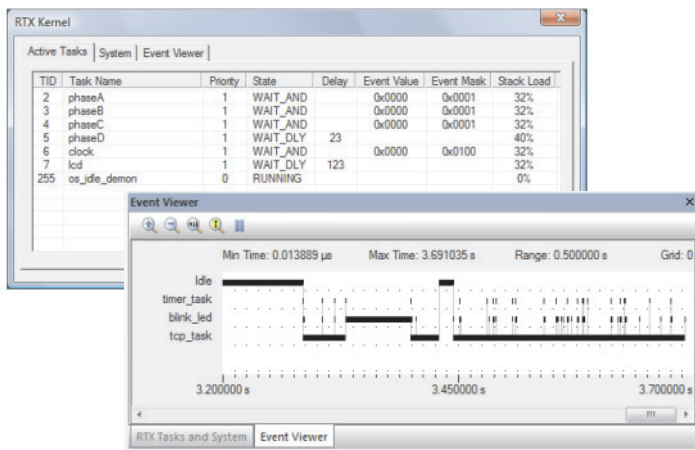
Task and event timing is conveniently displayed in the Event Viewer.
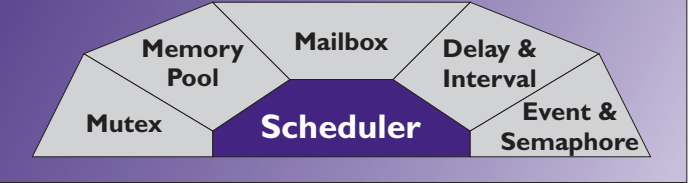
RTX Function Overview

- **Task Management Functions** allows you to create and delete tasks. RTX supports up to 254 active tasks, each with 254 priority levels.
- **Task Stacks** are allocated from a stack memory pool or can be supplied when a task is created.
- Fast **Memory Pool Management** allows you to create an unlimited number of fixed size pools.
- **Event Flag Management** allows synchronization with up to 16 event flags per task.
- **Time Management** and **Timer Callback Functions** provide task time delays/intervals.

RTX Real-Time Operating System Specifications	
General Specifications	
Defined Tasks (max)	Unlimited
Active Tasks (max)	254
Task Priority Levels	1 - 254
Signals / Events	16 per task
User Timers	Unlimited
Semaphores / Mailboxes / Mutexes	Unlimited
Memory Requirements	
CODE Space	<4KB
RAM Space (Kernel)	~500 Bytes
RAM Space (Task)	TaskStackSize + 52 Bytes
Typical Performance (cycles)	
Initialize system, start task	1,147
Create defined task, (no task switch)	403
Create defined task, (with task switch)	461
Delete task	218
Task switch (by os_tsk_pass)	192
Set event (no task switch)	89
Send semaphore (no task switch)	72
Send message (no task switch)	117

CODE and RAM space depend on which RTX functions are used.

Middleware

Today's microcontroller devices offer a wide range of communication interfaces to meet any embedded design requirement. However, implementing these interfaces presents software developers with real challenges. Middleware components are essential for developers to make efficient use of the device capabilities.

MDK-Professional includes a number of royalty-free, tightly coupled middleware libraries which enable developers to more easily implement complex communication interfaces in their applications.

- TCP Networking Suite.
- USB Host and Device interfaces.
- Flash File System.
- CAN drivers.

All middleware libraries have been specifically designed and optimized for ARM-powered MCU devices. The libraries are seamlessly integrated with the µVision environment and offer a modular design with well documented APIs.

MDK Middleware components allow you to develop robust applications using a wide variety of communication protocols.

TCPnet Networking Suite

The TCPnet library is a full networking suite optimized for ARM and Cortex-M processor-based MCUs. It has a small code footprint, and delivers excellent performance.

TCPnet provides comprehensive support for transmission protocols such as TCP/IP and UDP, as well as application level services and clients including HTTP, Telnet, SMTP, SNMP, and FTP. It provides all the features required for modern networking communication in embedded systems.

Visit: www.keil.com/rl-arm/rl-tcpnet.asp
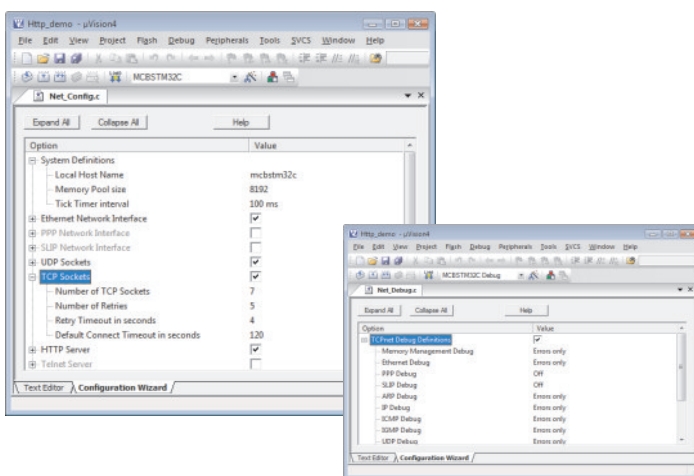
Menu-driven configuration is available for all TCPnet components including the Debug Interface.

TCPnet Networking Suite Features	
Applications	
HTTP Server (web server)	✓
Dynamic HTML and web-based forms	✓
HTTP File Upload via standard web browser	✓
External file download	✓
Web page caching	✓
Remote host filtering	✓
Multipacket POST processing	✓
Silverlight hosting	✓
Java URL checking	✓
Ajax Support (asynchronous and dynamic updates)	✓
SOAP (Simple Object Access Protocol)	✓
Telnet Server (text-based interface)	
Server Unsolicited Message Interface	✓
FTP server (file transfer)	✓
TFTP server (trivial file transfer)	✓
SMTP Client (email)	✓
User authentication	✓
Max MIB objects in SNMP message 20 objects	✓
DNS Client (IP address resolution)	✓
Automatic DNS Server configuration for PPP	✓
SNMP Agent (controls system settings and peripherals)	✓
User authentication	✓
Interfaces	
TCP Socket	
Multiple TCP/IP Connections	✓
TCP/IP Sliding Window Flow Control	✓
UDP Socket	
Multiple UDP Connections	✓
UDP Datagram Multicasting New!	✓
Physical	
Ethernet	✓
PPP (Point to Point Protocol)	✓
SLIP (Serial Line IP)	✓

USB Device and Host

MDK-Professional provides USB Device and USB Host support for embedded systems.

The **USB Device** Interface uses standard device driver classes that are available with all Windows PCs. No Windows host driver development is required. The USB Device interface uses a generic software layer using RTX Kernel features.

USB Device Driver Classes

- **Mass Storage Device** implements a memory device that can be accessed via the file system
- **Human Interface Device** implements a generic device that allows all types of data I/O
- **Audio Device** implements a device allowing streaming of audio as isochronous packets
- **Composite Device** implements support to enable multi-function USB devices

The **USB Host** library is an embedded USB stack supporting USB MSC (Mass Storage Class) and HID (Human Interface Device) classes. It has been designed to be high-performance while using as little memory as possible.
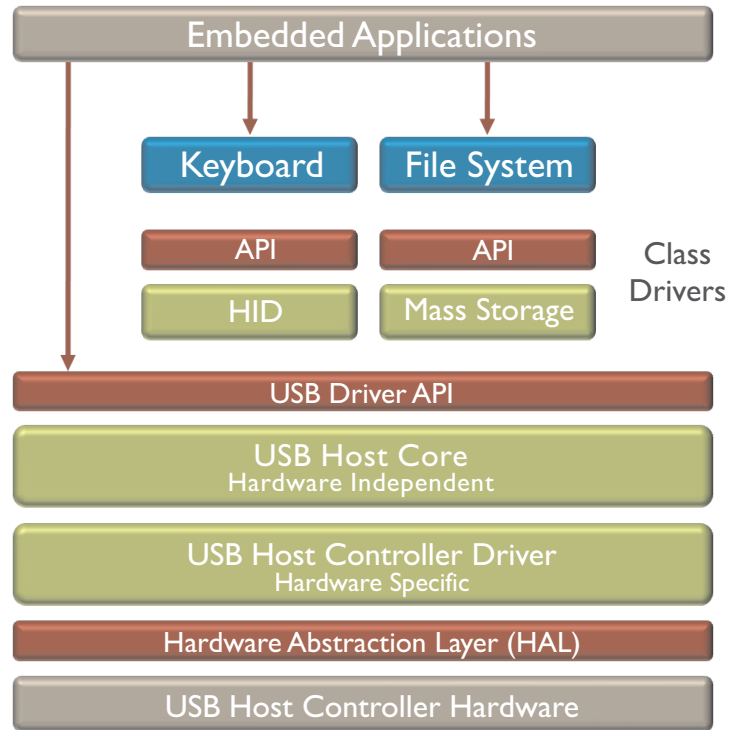
The library uses an abstraction layer allowing a standard API to be used for different USB Host controllers. It includes support for Open Host Controller Interface (OHCI) controllers and a custom host driver for STM32F105/7 devices.

```
┌──────────────────────────────────┐
│      Embedded Applications        │
└──────────────────────────────────┘
   ┌──────────┐    ┌──────────────┐
   │ Keyboard │    │ File System  │      Class
   └──────────┘    └──────────────┘      Drivers
   ┌──────────┐    ┌──────────────┐
   │   API    │    │     API      │
   └──────────┘    └──────────────┘
   ┌──────────┐    ┌──────────────┐
   │   HID    │    │ Mass Storage │
   └──────────┘    └──────────────┘
┌──────────────────────────────────┐
│          USB Driver API           │
└──────────────────────────────────┘
┌──────────────────────────────────┐
│         USB Host Core             │
│       Hardware Independent        │
└──────────────────────────────────┘
┌──────────────────────────────────┐
│     USB Host Controller Driver    │
│          Hardware Specific        │
└──────────────────────────────────┘
┌──────────────────────────────────┐
│  Hardware Abstraction Layer (HAL) │
└──────────────────────────────────┘
┌──────────────────────────────────┐
│   USB Host Controller Hardware    │
└──────────────────────────────────┘
```

The USB Host library is designed to work easily with numerous MCU Host controllers.

Visit: www.keil.com/rl-arm/rl-usb.asp

Flash File System

The Flash File System allows your embedded applications to create, save, read, and modify files in a wide range of standard storage devices.

- Standard ANSI C File I/O application interface
- NOR and NAND Flash support
- RAM, ROM, and SD/MMC/SDHC Memory Cards
- FAT12, FAT16, and FAT32 formats
- SD/MMC card file-caching
- Reentrant and thread-safe operation
- Simultaneous access to multiple storage devices

Visit: www.keil.com/rl-arm/rl-flash.asp

CAN Interface

RL-ARM includes a generic CAN interface layer which provides a standard programming API for all supported MCUs. It provides a quick and easy way to implement a CAN network. RL-CAN is composed of:

- Generic software layer that is identical across all supported microcontrollers
- Hardware-dependent software layer that implements the physical interface to the MCU CAN peripheral

To reduce complexity and ensure high-performance CAN message transmission, RL-CAN uses RTX functions for Mailbox Management and Memory Allocation.

Visit: www.keil.com/rl-arm/rl-can.asp
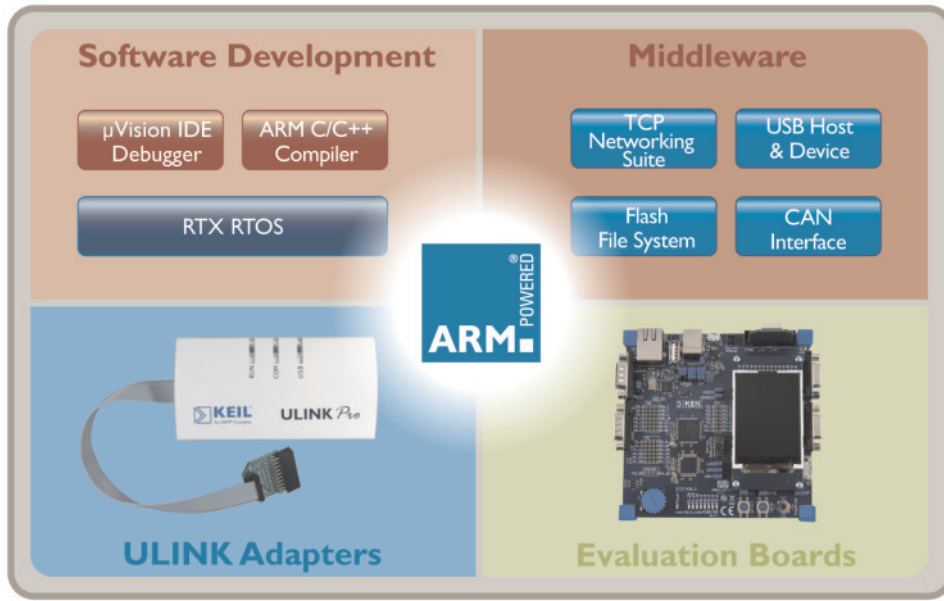
Keil **MDK** (Microcontroller Development Kit) offers a complete development environment for ARM7, ARM9, Cortex-M, and Cortex-R4 processor-based devices.

MDK is available in several product variants that support every level of software developer from the professional applications engineer to the student just learning about embedded software development. MDK is easy to learn and use, yet powerful enough for the most demanding embedded applications.

MDK Product Selector	MDK-Lite	MDK-Basic	MDK-Standard	MDK-Professional
µVision4				
IDE	✓	✓	✓	✓
Debugger	ROM: 32KB max	✓	✓	✓
Simulator	ROM: 32KB max	✓	✓	✓
ARM C/C++ Compiler	ROM: 32KB max	ROM: 256KB max	✓	✓
GNU GCC Support	ROM: 32KB max	✓	✓	✓
ARM Processor Support				
Cortex-M0, M1, M3, M4, & R4	✓	✓	✓	✓
ARM7 and ARM9	✓	✓	✓	✓
RTOS and Middleware Libraries				
RTX RTOS	✓	✓	✓	✓
TCPnet Networking Suite				✓
Flash File System				✓
USB (Host and Device)				✓
CAN Driver				✓

Europe:	+49 89/45 60 40 - 20	sales.intl@keil.com	support.intl@keil.com
United States:	+1 800 348 8051	sales.us@keil.com	support.us@keil.com

KEIL™
Tools by ARM

Program examples and detailed technical information are available from your distributor and our web site (www.keil.com).