



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要良心教育、做专业教育，更要做受人尊敬的职业教育。

## 《CORTEX-M3+UCOS-II 嵌入式系统开发入门与应用》

作者：华清远见

专业始于专注 卓识源于远见

### 第 3 章 ARM 开发环境 RealView MDK 平台搭建

## 3.1 常用 ARM 开发工具

ARM 应用软件的开​​发工具根据功能的不同，分别有编译软件、汇编软件、链接软件、调试软件、嵌入式实时操作系统、函数库、评估板、JTAG 仿真器以及在线仿真器等。目前世界上有 40 多家公司提供以上不同类别的产品。

用户选用 ARM 处理器开发嵌入式系统时，选择合适的开发工具可以加快开发进度，节省开发成本。因此一套含有编辑软件、编译软件、汇编软件、链接软件、调试软件、工程管理及函数库的集成开发环境（IDE）一般来说是必不可少的；至于嵌入式实时操作系统、评估板等其他开发工具则可以根据应用软件规模和开发计划选用。使用集成开发环境开发基于 ARM 的应用软件，包括编辑、编译、汇编、链接等工作全部在 PC 机上即可完成，调试工作则需要配合其他的模块或产品方可完成。目前常见的调试方法有以下几种。

- 指令集模拟器：部分集成开发环境提供了指令集模拟器，可方便用户在 PC 机上完成一部分简单的调试工作，但是由于指令集模拟器与真实的硬件环境相差很大，因此即使用户使用指令集模拟器调试通过的程序也有可能无法在真实的硬件环境下运行，用户最终必须在硬件平台上完成整个应用的开发。
- 驻留监控软件：驻留监控软件（Resident Monitors）是一段运行在目标板上的程序，集成开发环境中的调试软件通过以太网口、并行端口、串行端口等通信端口与驻留监控软件进行交互，由调试软件发布命令通知驻留监控软件控制程序的执行、读写存储器、读写寄存器、设置断点等。

驻留监控软件是一种比较低廉、有效的调试方式，不需要任何其他硬件调试和仿真设备。ARM 公司的 Angel 就是该类软件，大部分嵌入式实时操作系统也是采用该类软件进行调试，只是在嵌入式实时操作系统中，驻留监控软件是作为操作系统的任务存在的。

驻留监控软件的不便之处在于它对硬件设备的要求比较高，一般在硬件稳定之后才能进行应用软件的开​​发，同时它占用目标板上的一部分资源，而且不能对程序的全速运行进行完全仿真，所以对一些要求严格的情况不是很适合。

- JTAG 仿真器：JTAG 仿真器也称为 JTAG 调试器，是通过 ARM 芯片的 JTAG 边界扫描口进行调试的设备。JTAG 仿真器比较便宜，连接比较方便，通过现有的 JTAG 边界扫描口与 ARM CPU 核通信，属于完全非插入式（即不使用片上资源）调试，它无需目标存储器，不占用目标系统的任何端口，而这些是驻留监控软件所必需的。另外，由于 JTAG 调试的目标程序是在目标板上执行，仿真更接近于目标硬件，因此，许多接口问题，如高频操作限制、AC 和 DC 参数不匹配、电线长度的限制等被最小化了。使用集成开发环境配合 JTAG 仿真器进行开发是目前采用最多的一种调试方式。
- 在线仿真器：在线仿真器使用仿真头完全取代目标板上的 CPU，可以完全仿真 ARM 芯片的行为，提供更加深入的调试功能。但这类仿真器为了能够全速仿真时钟速度高于 100MHz 的处理器，通常必须采用极其复杂的设计和工艺，因而其价格比较昂贵。在线仿真器通常在 ARM 的硬件开发中使用，在软件开发中较少使用。其价格高昂是在线仿真器难以普及的因素。

以下章节重点介绍市面上常见的几种 ARM 开发工具。

### 3.1.1 免费平台 GNU

GNU 的调试工具包括编译器（gcc、g++）、二进制转换工具（objdump、objcopy）、调试工具（gdb、gdbserver、kgdb）和基于不同硬件平台的开发库。GNU 开发工具的主要缺点是采用命令行方式，用户掌握和使用比较困难，不如基于 Windows 系统的开发工具好用。GNU 工具的复杂性是由于它更贴近编译器和操作系统的底层，并提供了更大的灵活性。一旦学习和掌握了相关工具，也就了解了系统设计的基础知识，为今后的开发工作打下基础。GNU 的开发工具都是免费的，遵循 GPL 协议，任何人都可以从网上获取。

GNU 基于 ARM 平台的工具分别为 arm-linux-as (ARM 汇编器)、arm-linux-gcc (ARM C 编译器)、arm-linux-g++ (ARM C++编译器)、arm-linux-ld (ARM 连接器) 和 arm-linux-objcopy (ARM 二进制转换工具)。GNU 的所有开发工具都可以从 [www.gnu.org](http://www.gnu.org) 上下载, 基于 ARM 的工具可以从 [www.uclinux.org](http://www.uclinux.org) 获得。GNU 的编译器功能非常强大, 共有上百个操作选项, 这是这类工具让初学者头痛的原因。不过, 实际开发中只需要用到有限的几个, 大部分可以采用缺省选项。

GNU 工具的开发流程如下: 编写 C、C++ 语言或汇编源程序, 用 gcc 或 g++ 生成目标文件, 编写连接脚本文件, 用连接器生成最终目标文件 (elf 格式), 用二进制转换工具生成可下载的二进制代码。

### 3.1.2 RealView Development Suite 开发工具

RealView Development Suite (RVDS) 是 ARM 公司继 SDT 与 ADS1.2 之后主推的新一代开发集。RVDS 集成的 RVCT 是业内公认的能够支持所有 ARM 处理器, 并提供最好的执行性能的编译器。

RealView Development Suite 主要由 3 部分组成:

- RealView 编译工具 (RVCT);
- RealView 调试器 (RVD);
- RealView 内核模拟器 (RVISS)。

#### 1. RealView 编译工具 (RVCT)

RVCT 由一套工具连同支持文档和示例组成, 用于为 ARM 系列 RISC 处理器编写和编译应用程序。可以使用 RVCT 来编译 C、C++ 或 ARM 汇编语言程序。

开发工具主要由 armcc (ARM 和 Thumb C/C++ 编译器)、armasm (ARM 和 Thumb 汇编程序)、armlink (链接程序)、Rogue Wave C++ 库、支持库等组成。实用程序主要有 fromELF (ARM 映像转换实用程序)、armar (ARM 库管理程序), 可使多组 ELF 格式目标文件集中到一起并保留在库中。

#### 2. RealView 调试器 (RVD)

调试器主要包括以下部分。

- 多内核调试

RealView Debugger v1.6 为混合 ARM 和 DSP 调试提供了单一调试内核。该调试器完全支持断点的同步启动和停止、单步及交叉触发。

- 操作系统级调试: OS 认知

操作系统级调试主要包括暂停系统调试 (HSD)、暂停执行后询问并显示资源、访问信号和队列、查看当前线程和其他线程的状态、访问信号和队列、查看当前线程或其他线程的状态、自定义应用程序线程的视图等功能。

- 扩展的目标可见度 (ETV)

用户可以使用板芯片定义文件配置目标, 预配置文件通过 ARM DevZone 提供的客户/合作版或安装文件中得到。

- 高级调试设备

RVDS 提供标准调试视图和高级调试功能。RealView Debugger 在整个用户界面中支持 64 位“超长”型变量; 在 Call Stack 窗格中支持静态模型, 即非局部范围的静态变量; 强有力的命令行接口和脚本功能, 包括宏支持、从 ARM AXD 和 armsd 的转换以及记录以前操作的历史记录列表; RVD 可根据 Register、Watch 窗格或 Src 选项卡中变量或寄存器的内容找到并显示 Memory 窗格里的存储区域; 用户现在对 Code 窗口中的窗格和显示的调试视图有更大的控制权; RealView Debugger 提供在调试期间使用单个

Code 窗口显示系列数据视图的选项；Flash 编程模型作为标准组件提供；彩色存储器视图可根据存储器映射设置显示存储器类型。

- 跟踪、分析和配置

跟踪、分析和配置功能是 RealView Debugger V1.6 新增功能，通过跟踪调试许可证启动。跟踪支持适用于：

- ARM ETMv1.0 (ETM7 和 ETM9)，包括片上跟踪；
- DSP 模拟器；
- Motorola 56600 片上跟踪。

跟踪和配置功能可提供包括简单与复杂跟踪点以及数据过滤在内的全部跟踪支持：

- 查看源跟踪；
- 查看代码跟踪；
- 查看数据跟踪；
- 查看反汇编跟踪；
- 函数调用跟踪；
- 每个函数所花时间的配置报告；
- 按字段对捕获的跟踪数据进行分类的功能。

可以在源级视图和反汇编级视图中直接设置跟踪点。Memory 窗格提供相同的功能，以便可以选择要跟踪的存储区域，或者在特定存储器值发生变化时跟踪该值。

### 3. RealView ARMulator ISS

ARMulator ISS 是 ARM 内核指令集仿真器，可以提供指令精确的 ARM 处理器模拟，并能够使 ARM 和 Thumb 可执行程序在本机上运行。RealView ARMulator ISS 提供一系列模块，可以模拟 ARM 处理器内核和模拟处理器所有的存储器。

#### 3.1.3 ARM 硬件仿真器 Multi-ICE 与 Multi-Trace

Multi-ICE 是 ARM 公司自己的 JTAG 在线仿真器。Multi-ICE 支持 ARM7/9/10 和 Xsale 处理器系列。它通过 JTAG 接口连接到 TAP 控制器，支持多处理器以及混合结构芯片的在片调试，还支持低频或变频设计以及超低压核的调试，并且支持实时调试。

Multi-ICE 主要优点如下：

- 快速的下载和单步速度；
- 用户控制的输入/输出位；
- 可编程的 JTAG 位传送速率；
- 开放的接口，允许调试非 ARM 核或 DSP；
- 网络连接到多个调试器；
- 目标板供电或外接电源。

Multi-Trace 是一个嵌入式实时追踪模块。通过 ETM 接口（嵌入式追踪宏单元）与 ARM 处理器相连。Multi-Trace 包含一个处理器，因此可以跟踪触发点前后的轨迹（指令或数据），并且可以在不终止后台任务的情况下，对前台任务进行调试，在微处理器运行时改变存储器的内容。所有这些特性可使延时降到最低。

Multi-ICE + MultiTrace 是 ARM 实时调试解决方案之一。

#### 3.1.4 Multi 2000

Multi 2000 是美国 Green Hills 软件公司 (www.ghs.com) 开发的集成开发环境, 支持 C/C++/ Embedded C++/Ada 95/Fortran 编程语言的开发和调试, 可运行于 Windows 平台和 UNIX 平台, 并支持各类设备的远程调试。它包含完成一个软件工程所需要的所有工具, 这些工具可以单独使用, 也可集成第三方系统工具。Multi 2000 支持 Green Hills 公司的各类编译器以及其他遵循 EABI 标准的编译器, 同时 Multi 2000 支持众多流行的 16 位、32 位和 64 位处理器和 DSP, 如 PowerPC、ARM、MIPS、x86、Sparc、TriCore 以及 SH-DSP 等, 并支持多处理器调试。

### 3.1.5 Hitool for ARM

Hitool for ARM 由 Hitool International Inc 公司出品, 是目前比较新的 ARM 嵌入式应用软件开发系统, 主要包括 Hitool ARM Debugger、GNU Compiler (内建)、JTAG cable、评估板以及嵌入式实时操作系统 ThreadX 等。其中编译器模块可以替换成 ARM ADS Compiler 或 ARM SDT Compiler。

Hitool for ARM 开发工具具有以下特性。

- 近似 MS Visual Studio 的调试界面风格, 可以在 Windows 98/ME/NT 等多种 Win32 环境下运行。
- 具有优秀的工程管理器、源代码和二进制代码编辑器、字符串搜索引擎以及调试目标的自由拖放等功能。
- 支持汇编、C 以及 C++ 源码级调试, 不仅可以通过串口和并口进行本地调试, 也可以通过 TCP/IP 进行远端调试。
- 集成了 S-Record、Binary 和 Disassembly 格式的内存上下下载工具, Flash 编程工具。
- 支持多种常用的 Jtag Cable, 具备通过宏和脚本实现的自动化调试功能。

### 3.1.6 RealView MDK

RealView MDK 开发套件源自德国 Keil 公司, 是 ARM 公司目前最新推出的针对各种嵌入式处理器的软件开发工具。RealView MDK 集成了业内最领先的技术, 包括  $\mu$ Vision3 集成开发环境与 RealView 编译器。它支持 ARM7、ARM9 和最新的 Cortex-M3 核处理器, 具有自动配置启动代码、集成 Flash 烧写模块、强大的 Simulation 设备模拟、性能分析等功能, 与 ARM 之前的工具包 ADS 等相比, RealView 编译器的最新版本可将性能改善超过 20%。

下面将主要介绍 RealView MDK 的配置和使用。

## 3.2 RealView MDK 概述

RealView MDK 全称 RealView MDK 中国版开发套件, 该套件的产品模块包括  $\mu$ Vision3 IDE、启动代码生成向导、设备模拟器、性能分析器、RealView 编译器、MicroLib、RealView Real-time Library 实时库。

### 1. $\mu$ Vision3 IDE

$\mu$ Vision IDE 在全球拥有庞大的用户群, 超过 10 万开发工程师在使用 Keil 开发工具。不管是用 8 位、16 位 MCU, 还是改用 ARM 32 位处理器, 都可以使用  $\mu$ Vision IDE 进行编辑。编辑器的统一, 大大降低了开发门槛, 可以使工程师很方便地在新环境下开发。

图 3.1 显示了 RealView MDK 开发环境 IDE。

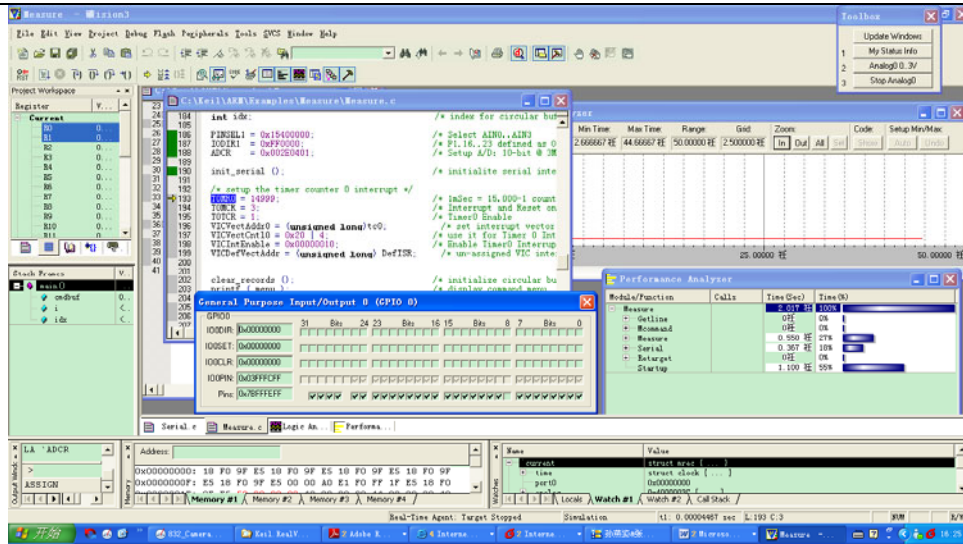


图 3.1 RealView MDK 开发环境 IDE

RealView MDK 开发环境 IDE 主要包括以下几个部分：

- 功能强大的源代码编辑器；
- 可根据开发工具配置的设备数据库；
- 用于创建和维护工程的工程管理器；
- 集汇编、编译和链接过程于一体的编译工具；
- 用于设置开发工具配置的对话框；
- 真正集成高速 CPU 及片上外设模拟器的源码级调试器；
- 高级 GDI 接口，可用于目标硬件的软件调试和 ULINK2 仿真器的连接；
- 用于下载应用程序到 Flash ROM 中的 Flash 编程器；
- 完善的开发工具手册、设备数据手册和用户向导。

## 2. 启动代码配置向导

$\mu$ Vision3 IDE 的启动代码配置向导将各个所需配置的功能模块以对话框方式展示，附加提示说明，可以帮助用户快速轻松地做出选择，生成完善的启动代码，免除手工写几百行汇编程序的痛苦。

图 3.2 显示了 RealView MDK 的启动代码配置向导。

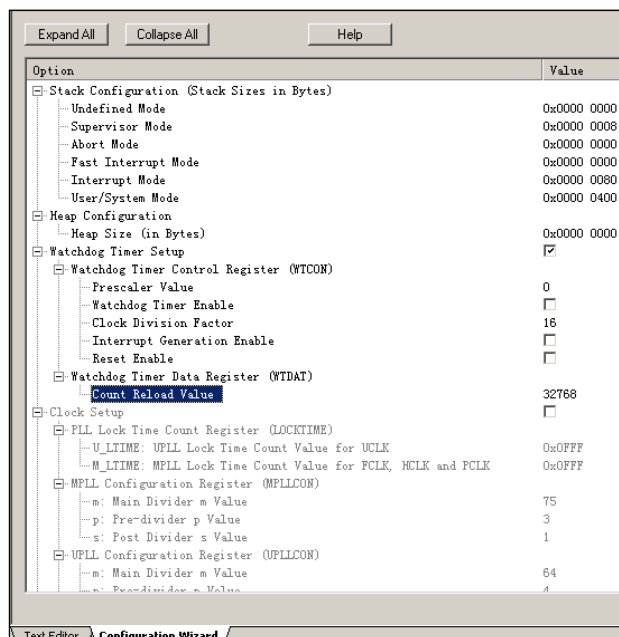


图 3.2 启动代码配置向导

### 3. 设备模拟器

μVision3 设备模拟器的功能强大，能模拟整个 MCU 的行为，使用户在没有硬件或对目标 MCU 没有更深的了解的情况下，仍然可以立即开始开发软件。其仿真功能包括：

- 高效指令集仿真；
- 中断仿真片内外围设备；
- ADC、DAC、EBI、Timers、UART、CAN、I<sup>2</sup>C、外部信号和 I/O 仿真。

图 3.3 显示了设备模拟器视图。

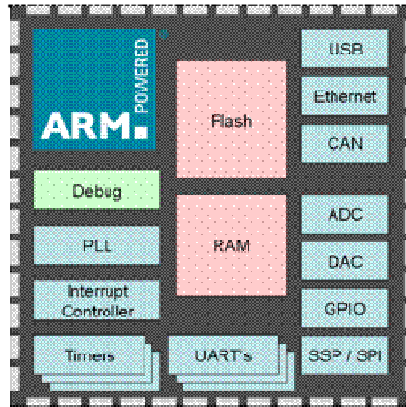


图 3.3 RealView MDK 设备模拟器视图

### 4. 性能分析器

性能分析器可实现如程序运行时间统计、被调用次数统计、代码覆盖率统计等高端功能，而这些功能对于快速定位死区代码、帮助优化分析等起了关键的作用。

图 3.4 显示了性能分析器视图。

Module/Function	Calls	Time (Sec)	Time (%)
BOOT_FLASH		2.000 秒	100%
ucos_ii		0 秒	0%
OS_SchedNew	0	0 秒	0%
OSTimeDly	0	0 秒	0%
OSTaskSuspend	0	0 秒	0%
OSTaskResume	0	0 秒	0%
OSSemPost	0	0 秒	0%
OSSemPend	0	0 秒	0%
OSSemCreate	0	0 秒	0%
OS_EventWaitListInit	0	0 秒	0%
OS_EventTO	0	0 秒	0%
OS_EventTaskWait	0	0 秒	0%
OS_EventTaskRdy	0	0 秒	0%
OSTimeTick	0	0 秒	0%
OSStart	0	0 秒	0%
OSIntExit	0	0 秒	0%
OSInit	0	0 秒	0%
OS_MemClr	0	0 秒	0%
OSTaskCreateExt	0	0 秒	0%
OS_TaskStkClr	0	0 秒	0%
OS_TCBInit	0	0 秒	0%
OS_Sched	0	0 秒	0%
OS_TaskIdle	0	0 秒	0%
os_dbg		0 秒	0%
os_cpu_c		0 秒	0%
os_cpu_a		0 秒	0%
os_thread		0 秒	0%
main		0 秒	0%
stm32f10x_it		0 秒	0%
timer		0 秒	0%
adc12		0 秒	0%
io		0 秒	0%
uart		0 秒	0%

图 3.4 RealView MDK 性能分析器视图

## 5. RealView 编译器 (RVCT)

RealView MDK 集成的 RealView 编译器 (跟 RVDS 使用一样的编译器) 是业界最优秀的编译器, 它能使代码容量更小、执行效率更高, 使应用程序运行更快、系统成本更低。

## 6. MicroLib

为进一步改进基于 ARM 处理器的应用代码密度, RealView MDK 采用了新型 microlib C 库 (用于 C 的 ISO 标准运行时库的一个子集), 并将其代码镜像降低到最小以满足微控制器应用的需求。MicroLib C 库可将运行时库代码大大降低。

## 7. RealView Real-time Library 实时库

RealView 实时库 (RealView RL-ARM) 是为解决基于 ARM MCU 的嵌入式系统中的实时及通信问题而设计的紧密耦合库集合。它可以非常方便地应用于所有 ARM7、ARM9 和 Cortex-M3 系列的处理器, 使得在 ARM 处理器上运行实时程序非常容易。

该实时库包含 4 个部分: RTX 实时内核、Flash 文件系统、TCP/IP 协议簇和 RTL-CAN (控制域网络)。

RealView 实时库可以解决嵌入式开发中的如下几个常见问题:

- 多任务 (可以在单 CPU 上管理几个工作或任务);
- 实时控制 (可以控制任务在既定时间内完成);
- 任务间通信 (可以实现系统中的任务间通信);
- Internet 连接 (通过以太网或串口 (Modem));
- 嵌入式 Web 服务器 (包括 CGI 脚本);
- E-mail 公告 (通过 SMTP)。

此外, RealView RL-ARM 还包括如下几个用于 RTX 实时内核与各种通信接口连接的驱动器。

- CAN 驱动: CAN 驱动可用于 STR71x、STR73x、STR75x、and STR91x 等设备上。
- USB 设备驱动: 基于标准 Windows 的 USB 设备驱动可用于 LPC2000 设备上。

## 3.3 RealView MDK 的使用

### 3.3.1 创建工程

$\mu$ Vision 是一个标准的窗口应用程序, 可以单击“程序”按钮开始运行。为了创建一个新的  $\mu$ Vision 工程必须做如下处理:

- 选择工具集;
- 创建工程文件;
- 选择设备。

#### 1. 选择工具集



μVision 可以使用 ARM RealView 编译工具、ARM ADS 编译器、GNU GCC 编译器和 Keil C ARM 编译器。当使用 GNU GCC 编译器或 ARM ADS 编译器时必须另外安装它们编译集。实际使用的工具集可以选择 Project→Manage”，在弹出的“Components,Environment and Books”对话框中选择“Folders/Extensions”页，如图 3.5 所示。

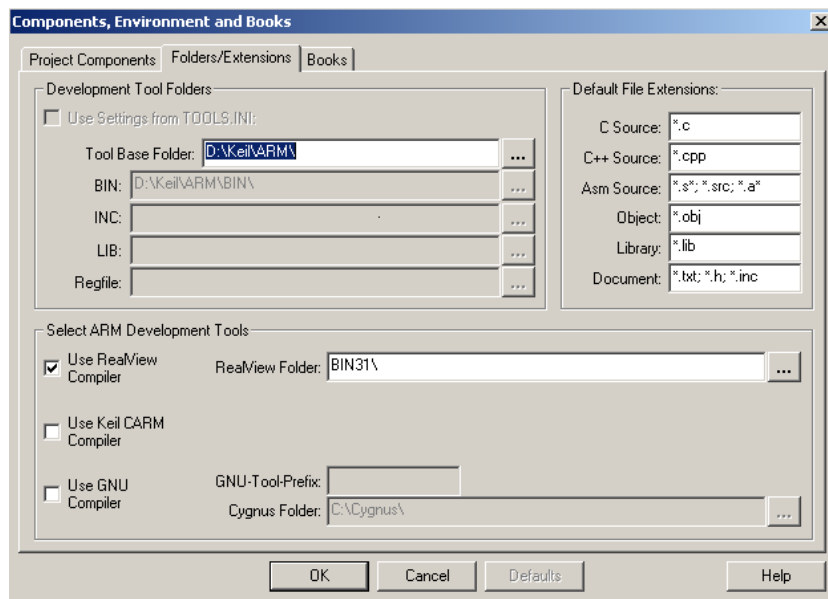


图 3.5 选择工具集

(1) “Use RealView Compiler”复选框表示本工程使用 ARM 开发工具。“RealView Folder”文本框指定开发工具的路径。

下面的例子显示了各种版本的 ARM ADS/RealView 开发工具的路径。

- μVision 的 RealView 编译器: BIN31\;
- ADS V1.2: C:\Program Files\ARM\ADSV1\_2\Bin;
- RealView 评估版 2.1: C:\Program Files\ARM\RVCT\Programs\2.1\350\eval2-sc\win\_32-pentium。

(2) “Use Keil CARM Compiler”复选框表示本工程使用 Keil CARM 编译器、Keil AARM 汇编器和 Keil LARM 链接器/装载器。

(3) “Use GNU Compiler”复选框表示本工程使用 GNU 开发工具。“Cygnus Folder”文本框指定 GNU 的安装路径。“GNU-Tool-Prefix”文本框指定不同的 GNU 工具链。下面是各种 GNU 版本的例子。

- 带 uclib 的 GNU V3.22: GNU-Tool-Prefix: arm-uclibc- Cygnus Folder: C:\Cygnus;
- 带标准库的 GNUARM V4: GNU-Tool-Prefix: arm-elf-Cygnus Folder: C:\Program Files\GNUARM\。

Keil 根目录的设置是基于 μVision/ARM 开发工具的安装目录的。对于 Keil ARM 工具来说，工具组件的路径是在开发工具目录中配置的。

## 2. 创建工程文件

单击 Project→New...→μVision Project 菜单项，μVision 3 将打开一个标准对话框，输入希望新建工程的名字即可创建一个新的工程，建议对每个新建工程使用独立的文件夹。例如，这里先建立一个新的文件夹，然后选择这个文件夹作为新建工程的目录，输入新建工程的名字 Project1，μVision 将会创建一个以 Project1.UV2 为名字的新工程文件，它包含了一个缺省的目标 (target) 和文件组名。这些内容在 Project Workspace→Files 中可以看到。

## 3. 选择设备

在创建一个新的工程时，μVision 要求为这个工程选择一款 CPU。当选择设备对话框显示了 μVision 的设备数据库后，只需要选择用户所需的微控制器即可。例如，图 3.6 显示了选择 STM32F101 微控制器，简化了工具的配置。

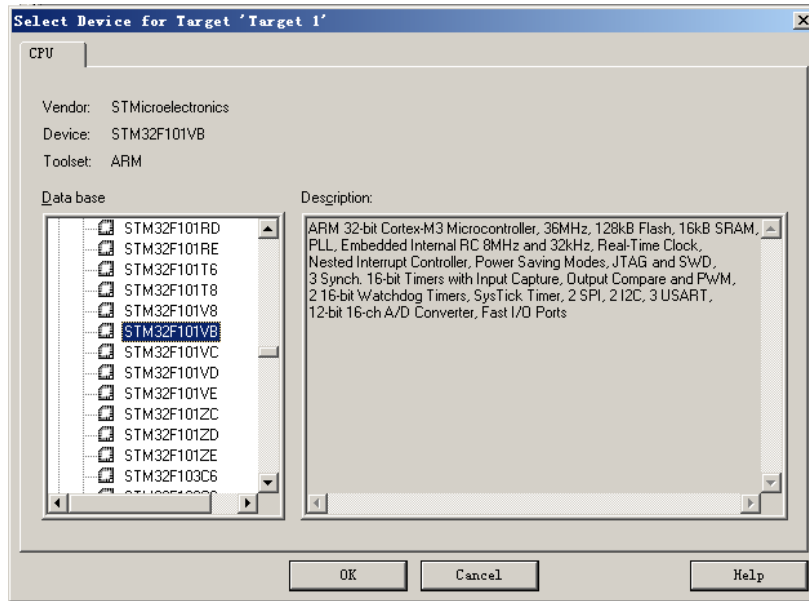


图 3.6 选择设备

创建源文件以后，就可以将这个文件添加到工程中。μVision 提供了几种方法将源文件添加到工程中。例如，在 Project Workspace→Files 页的文件组上单击鼠标右键，然后在弹出的菜单中选择“Add Files”菜单项，这时将打开标准的文件对话框，选择我们创建的.asm 或.c 文件即完成源文件的添加。

### 3.3.2 编译、链接工程

#### 1. 设置目标硬件的工具选项

μVision 可以设置目标硬件的选项。通过工具栏按钮或选择“Project→Options for Target”菜单项，打开“Options for Target”对话框，在 Target 页中设置目标硬件及所选 CPU 片上组件的参数。STM32F103 的一些参数设置如图 3.7 所示。

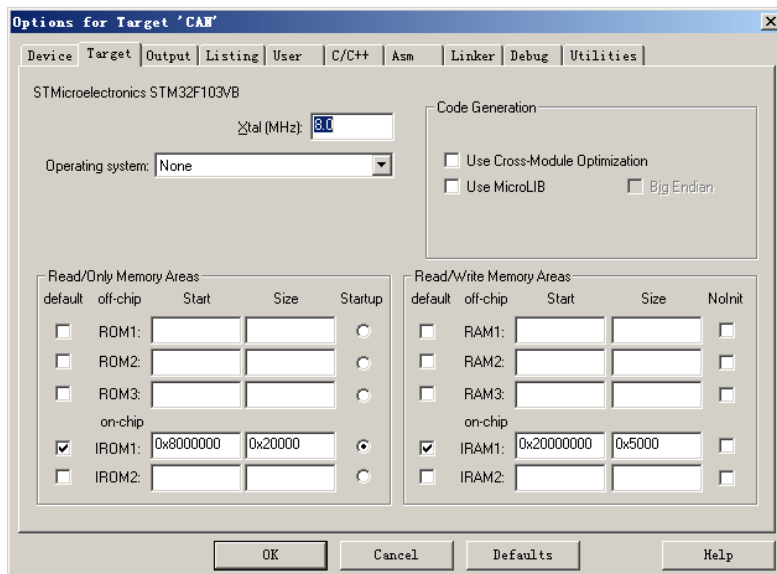


图 3.7 STM32F103 设置参数

表 3.1 描述了参数设置对话框各选项内容。

表 3.1 参数设置对话框选项内容

对话框项	描述
Xtal	设备的晶振 (XTAL) 频率。大多数基于 ARM 的微控制器都使用片上 PLL 产生 CPU 时钟, 所以一般情况下 CPU 的时钟与 XTAL 的频率是不同的
续表	
对话框项	描述
Read/Only Memory Area	配置片内、片外的 ROM 区地址以及大小
Read/WriteMemory Areas	指定目标硬件的片内和片外的 RAM 区地址以及大小
Code Generation	旋转产生 ARM code 还是 Thumb code

## 2. 增加连接控制文件

对于 GNU 和 ARM ADS/RealView 工具链来说, 链接器的配置是通过链接器控制文件实现的。这个文件指定了 ARM 目标硬件的存储配置。预配置的链接器控制文件在文件夹..\ARM\GNU 或..\ARM\ADS 中。为了与目标硬件相匹配, 用户可能会修改链接器控制文件, 所以工程中的那个文件是预配置的连接控制文件的一个副本。这个文件可以通过 Project-Options for Target 对话框的 Linker 页添加到工程中。

图 3.8 显示了链接配置文件对话框。

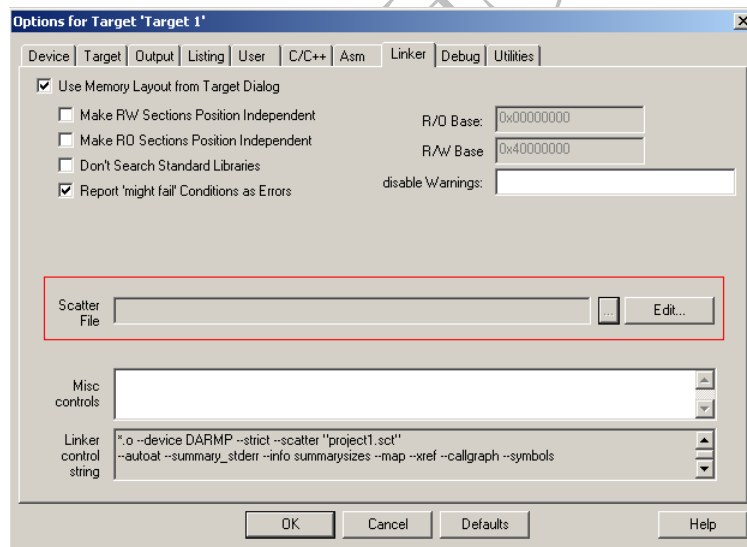


图 3.8 链接配置文件对话框

对于复杂的 memory layout 分配方式, 应该采用 scatter File 选项指定 Scatter 文件, 对于简单的工程, 直接指定 R/O 和 R/W 的基地址即可。关于 Scatter 文件的更多内容, 请读者参考 ARM 工具相关资料。

## 3. 编译链接

一般来说, 在新建一个应用程序的时候 Options→Target 页中的所有的工具和属性都要配置。单击“Build Target”工具栏按钮将编译所有的源文件, 链接应用程序。当编译有语法错误的程序时, μVision 将在 Output Window→Build 窗口中显示错误和警告信息。单击这些信息行, μVision 将会定位到相应的源代码处。图 3.9 显示了编译结果输出框。

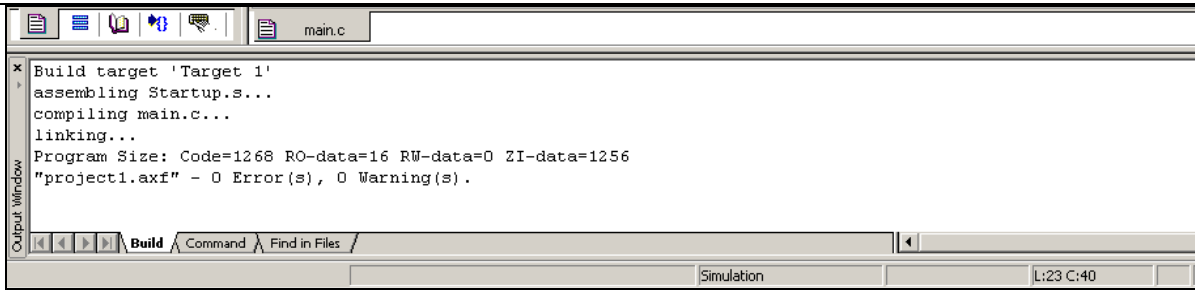


图 3.9 编译结果输出框

源文件编译成功产生应用程序以后就可开始调试了，单击“Debug→Start/Stop debug session”命令（Ctrl F5）即进入调试模式。

### 3.3.3 程序调试

进入调试模式之后，可以选择单步、全速运行；可以设置断点等常规的调试。所有有关调试的操作都可以在 Debug 菜单下找到。图 3.10 显示了程序进入调试模式时的界面。

当程序进入调试时，常用的调试手段有以下几种。

- 单步、全速运行程序，按快捷键 F10 进入单步运行程序；按快捷键 F5 进入全速运行程序。
- 对于各种模式下的寄存器，可以在左边的窗口中查看。可以查看 ARM 的 7 种模式下的寄存器。当处理器处于任何一种模式时，可以查看 Current 中所有的寄存器的值。处理器从一种状态改变到另外一种状态时，该模式下物理上独立的寄存器将会被用到。
- 设置断点：选中需要设置断点的行，然后按 F9 键即在该行设置断点，程序运行到此处就停止运行。
- 查看变量的实时值：对于局部变量，选择“View→Watch&Call Stack Window”，选择 Locals tab 就可以查看所有的 local 变量。对于全局变量，选择 Watch window 中的 Watch #1，加入你需要查看的变量就可以查看实时的全局变量的值。
- 外设模块仿真：选择 Peripheral→GPIO 可以看到每一个 GPIO pin 的实时状态信息。全速运行程序后，GPIO 的状态就开始按照程序的控制开始变化。

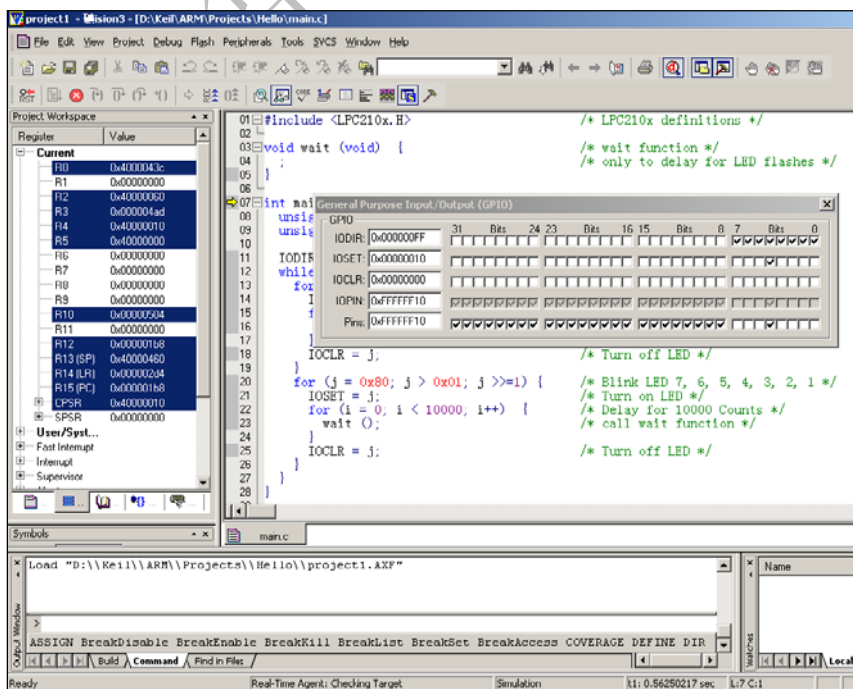


图 3.10 Simulator 程序调试界面

### 3.3.4 工程选项设置

在 Project-Options 对话框页中可以设置所有的工具选项。所有的选项都保存在  $\mu$ Vision 工程文件中，工程文件后缀为 “.Uv2”。

在 Project Workspace→Files 窗口单击鼠标右键，在弹出的菜单中可以设置文件夹或单个文件的不同属性。表 3.2 描述了各种选项对话框的功能。

表 3.2 选项对话框功能描述

对话框项	描述
Device	从 $\mu$ Vision 的设备数据库中选择设备
Target	为应用程序指定硬件环境
Output	定义工具链的输出文件，在编译完成后运行用户程序
Listing	指定工具链产生的所有列表文件

续表

对话框项	描述
C	设置 C 编译器的工具选项，例如代码优化和变量分配
Asm	设置汇编器的工具选项，如宏处理
Linker	设置链接器的相关选项。一般来说，链接器的设置需要配置目标系统的存储分配。设置链接器定义存储器类型和段的位置
Debug	$\mu$ Vision 调试器的设置
Utilities	配置 Flash 编程实用工具

## 3.4 仿真工具——ULINK2

ULINK2 仿真器是 ARM 公司最新推出的配套 RealView MDK 使用的仿真器，是 ULink 仿真器的升级版本。ULINK2 不仅具有 ULINK 仿真器的所有功能，还增加了串行调试（SWD）支持、返回时钟支持和实时代理等功能。开发者通过结合使用 RealView MDK 的调试器和 ULINK2，可以方便地在目标硬件上进行片上调试（使用 on-chip JTAG、SWD 和 OCDS）、Flash 编程等。

ULINK2 仿真器的技术规格如表 3.3 所示。

表 3.3 ULINK2 技术规格

特 性	规 格
RAM 断点	无限多个
ROM 断点（ARM7/9）	最多 2 个
ROM 断点（cortex-M3）	最多 6 个
执行断点（程序执行时设置的断点）	支持
Access 断点（ARM7/9）	最多 2 个
Access 断点（Cortex-M3）	最多 4 个
Trace 功能	无
Real-Time Agent	支持
JTAG 时钟	$\leq 10\text{MHz}$
JTAG RTCK 支持	支持

内存访问读写速度 (byte/s)	≈28KB	续表
<b>特 性</b>	<b>规 格</b>	
Flash 读写速度 (byte/s)	≈25KB	
单步执行速度 (指令/秒)	≈50	

### 3.5 RealView MDK 工程实例

本节将使用 RealView MDK 建立并调试 HelloWorld 应用程序。

#### 1. 建立 HelloWorld 工程

使用 Projctect→new μVision Project 建立新的应用程序工程，起名为 HELLO。  
图 3.11 显示了创建新工程对话框。

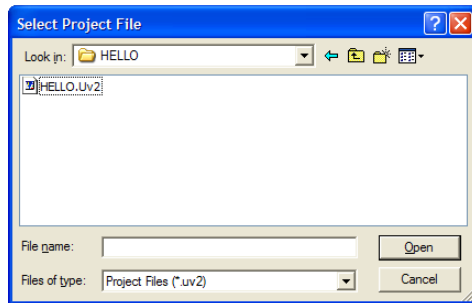


图 3.11 创建新工程——HELLO

#### 2. 编辑程序源文件

可以在 μVision 工程环境下添加、删除、编辑、查找源文件，如图 3.12 所示。

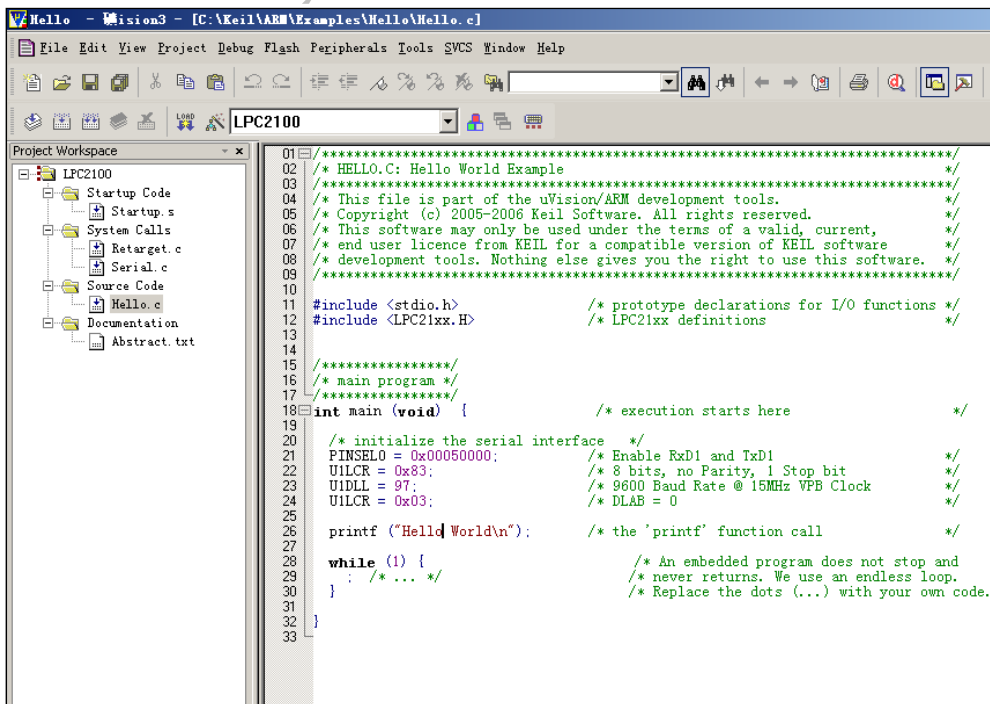


图 3.12  $\mu$ Vision 工程环境

### 3. 编译、链接工程文件

当工程编辑结束，可使用 Project-Build Target 命令编译工程。编译结果显示在 output 窗口中，如图 3.13 所示。

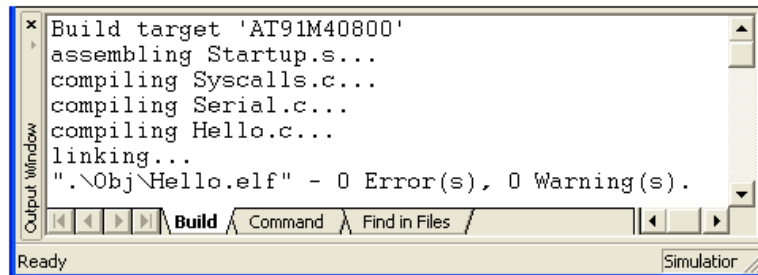


图 3.13 输出编译结果

### 4. 程序测试

$\mu$ Vision 工程环境下的  $\mu$ Vision debugger 可以对编译好的工程文件进行测试。在 Debug 目录下选择 Start/Stop Debug Session 命令开始/结束 Debug 测试。调试命令及其对应图标和说明如表 3.4 所示。

表 3.4 调试命令及其对应图标和说明

调试命令	工具栏图标	命令说明
串口输出		打开 Window#1 窗口，接收串口输入字符
连续运行		连续运行程序，直到遇到断点
停止程序运行		停止程序运行，观察程序输出结果

程序调试完整视图如图 3.14 所示。

### 5. 观察程序反汇编指令

调试过程中可以观察程序的反汇编，如图 3.15 所示。

### 6. 观察内存数据

当程序停在断点上时，可以通过内存窗口观察程序的内存数据，检测程序运行是否正常，如图 3.16 所示。本节完整演示了一个基于  $\mu$ Vision 的工程建立、编辑和调试的过程。使用  $\mu$ Vision 可以实现更多、更复杂的功能，如调试跟踪功能等，详细内容请参见 ARM RealView MDK 开发手册。

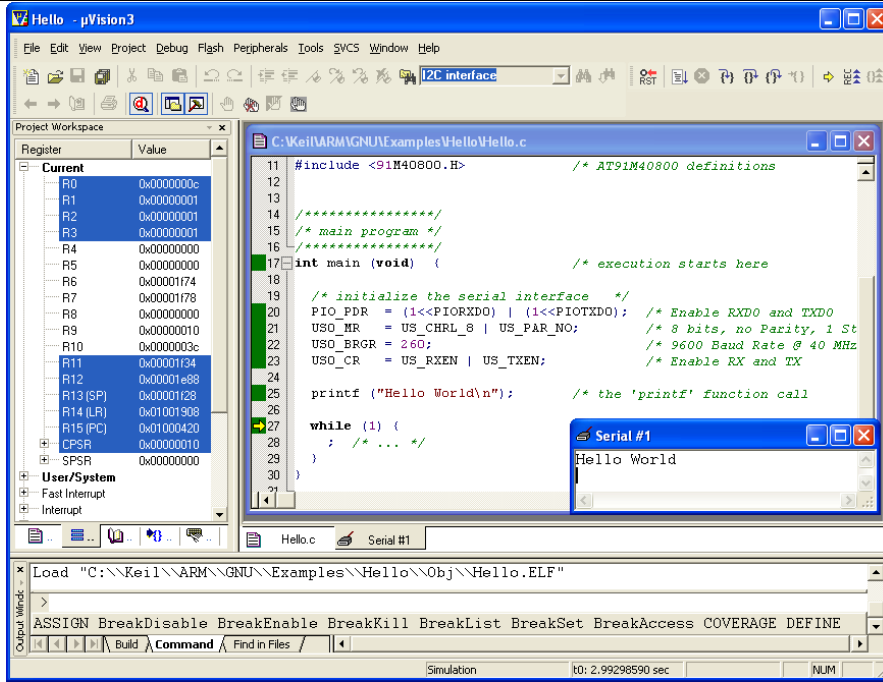


图 3.14 程序调试视图

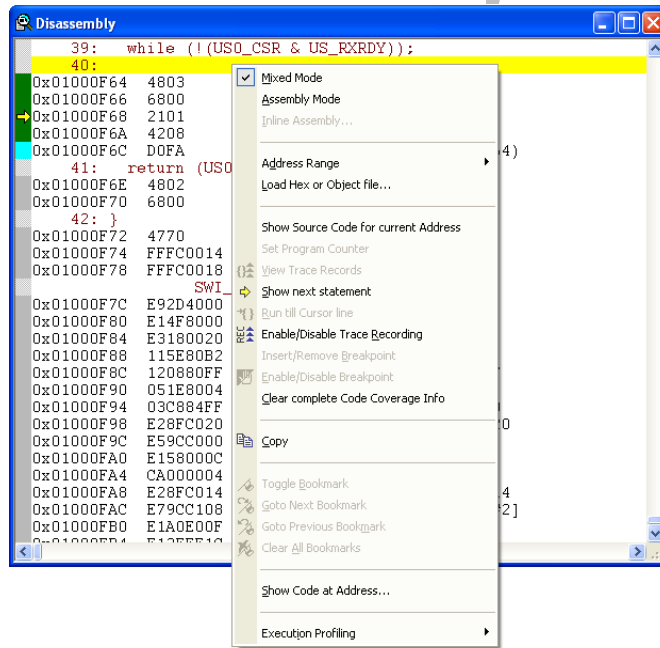


图 3.15 程序反汇编窗口

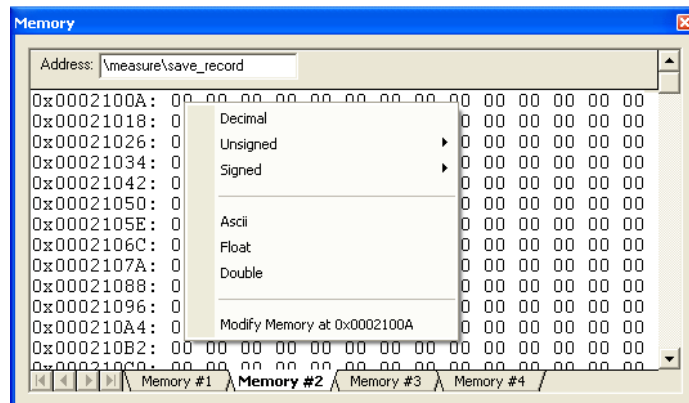


图 3.16 内存数据输出窗口



## 联系方式

集团官网: [www.hqyj.com](http://www.hqyj.com)      嵌入式学院: [www.embedu.org](http://www.embedu.org)      移动互联网学院: [www.3g-edu.org](http://www.3g-edu.org)

企业学院: [www.farsight.com.cn](http://www.farsight.com.cn)      物联网学院: [www.topsight.cn](http://www.topsight.cn)      研发中心: [dev.hqyj.com](http://dev.hqyj.com)

集团总部地址: 北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址: 北京市海淀区西三旗悦秀路北京明园大学校区, 电话: 010-82600386/5

上海地址: 上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区, 电话: 021-54485127

深圳地址: 深圳市龙华新区人民北路美丽 AAA 大厦 15 层, 电话: 0755-22193762

成都地址: 成都市武侯区科华北路 99 号科华大厦 6 层, 电话: 028-85405115

南京地址: 南京市白下区汉中路 185 号鸿运大厦 10 层, 电话: 025-86551900

武汉地址: 武汉市工程大学卓刀泉校区科技孵化器大楼 8 层, 电话: 027-87804688

西安地址: 西安市高新区高新一路 12 号创业大厦 D3 楼 5 层, 电话: 029-68785218

