



创建出色的32位微控制器应用的四步骤

作者
Rod Crawford

使用ARM RealView Microcontroller Development Kit (RVMDK)微控制器开发套件的四步式指南。

指南 1：选择正确的微控制器

摘要

为应用选择合适的基于ARM®内核的微控制器（MCU）有可能是一个复杂的令人望而却步的任务。目前基于ARM内核的可用微控制器数量超过125款，寻找一款符合相应外围设备组合和性能标准的微控制器需要投入大量研究。本白皮书描述如何使用Device Database®中的参数搜索引擎（与RealView® Microcontroller Development Kit微控制器开发套件相集成），为应用快速选择正确的MCU。此外，使用Device Database可以在RVMDK中为选定的MCU部件配置相应的工具。

概述

在现代嵌入式开发领域中，对领先竞争对手的需求，结合低成本高性能32位微控制器的出现，推动了这些部件在新产品中的广泛应用，由此取代过去的8位和16位微控制器。嵌入式开发人员在选择微控制器时，遵循四项主要标准：功能、可用性、成本和熟悉程度。本文中，我们讨论如何使用联机Device Database帮助开发人员为其应用作出正确选择。我们还演示Device Database如何与RealView Microcontroller Development Kit微控制器开发套件集成，使开发人员能自动配置他们的工具链，使用选定的MCU进行开发。

选择问题

今天，在选择具备所需外围设备组合和性能并且价格适当的MCU时，开发人员可选的余地很大。生产基于ARM内核的MCU的半导体公司数量超过16家，其

项	描述
部件描述	每个设备的简要描述，包括部件编号、内核和外围设备组合。
头文件	C语言和汇编语言头文件，它们描述可用于该MCU的SFR（特殊功能寄存器）。
示例代码	为选定MCU或MCU系列预先编写的代码片断和示例程序。
外围设备仿真	μ Vision®集成开发环境（IDE）完全仿真的片上外围设备列表，它集成到RVMDK中。
数据表	与MCU部件相关的数据表，可从Keil网站下载。
开发板	可用于选定MCU的评估开发板。
仿真器	可用于选定MCU的仿真器。
第三方软件	可在MCU上运行的软件，如实时操作系统和应用程序库。
顾问	熟悉此MCU的软件和硬件咨询公司列表。

中包括一些全球最大的MCU提供商。每家半导体供应商提供的MCU不只一款，且是针对特定市场领域的一系列MCU。在编写本文时，共有超过125款基于ARM内核的MCU，而且该数字每周仍在不断增长。

部分供应商提供一些在线搜索工具，使开发人员能对其产品组合中的MCU作出比较和对比。但是如果要比不同供应商的MCU，它就毫无价值了。我们真正需要的是一个工具，它使开发人员能对相同产品系列、不同产品系列和不同供应商的MCU作出比较和对比。这正是Device Database的切入点。

Device Database

Device Database是一个可搜索数据库，该数据库包含Keil™（ARM公司）软件开发工具支持的各种MCU的相关信息，其中包括RealView Microcontroller Development Kit微控制器开发套件支持的那些基于ARM内核的MCU。Device

Database有两个版本，一个是www.keil.com/dd上的在线版本，另一个则内建在RVMDK中。只有基于Web的版本提供参数搜索功能。

该数据库支持以下MCU信息：

除此以外，该数据库还包含指向FindChips.com网站的链接，其中列出了不同MCU的经销商。

数据库搜索

可按下列方式在Device Database中搜索MCU：

- 架构——显示支持特定处理器架构的所有MCU。例如，按ARM架构搜索将列出MCU内核中支持ARM架构的所有供应商。
- 供应商——显示特定半导体供应商生产的所有MCU。例如，将飞利浦作为供应商进行搜索将列出飞利浦生产的所有MCU系列的MCU。
- 参数——允许开发人员输入他们要查找的MCU的要求。

使用参数搜索强大功能

开发人员最感兴趣的可能是Device Database的参数搜索功能，它使开发人员能搜索整个MCU范围内的特定外围设



备组合和性能标准，而不会受到任何一家处理器供应商的搜索功能限制。使用这种搜索，开发人员可以迅速确定一个MCU部件或MCU系列，它们可提供最合适的性能、外围设备组合以及未来可扩展性。

参数搜索示例

为了展示Device Database参数搜索的一些高级功能，我们将使用以下示例。

示例：远程温度记录设备

我们要创建一个远程温度传感设备，它能持续记录来自传感器的信息并将它们存储在本地。现场使用时设备应由电池供电，如果崩溃则应有故障重新启动功能。通过USB连接设备时，它应该能通过USB转储其日志，并同步时钟以便进一步收集数据。基于这些条件，我们可以将MCU要求精简如下：

外围设备	参数
片上 RAM	>8K
片上闪存	>64K
A/D 通道	8+
断电模式	是
闲置模式	是
实时时钟	是
看门狗计时器	是
USB	是

如果我们将这些参数作为参数搜索输入Device Database，我们发现目前有两个来自不同MCU供应商、基于ARM内核的MCU系列，每个系列都有四个MCU部件符合上述要求。在不同MCU供应商范围内使用这种搜索将超过125种MCU的选择工作缩减到8种。此时，我们可以检查Device Database中各个部件项，并根据更多客观条件（如过去的经验、文档清晰性、应用程序库和示例代码的相关性，当然还有价格/批量）作出最终决策。

RealView Microcontroller Development Kit 微控制器开发套件配置

在上述示例中，我们已说明如何在

Web上将Device Database用作独立工具，根据特殊应用的MCU选择作出明智的决策。但Device Database的用途远不止此，因为在特殊应用项目的设置阶段中也可以使用它。RVMDK包含一个Device Database版本，在开发人员着手新项目时调用它。当开发人员在RVMDK的µ Vision IDE中指定Device Database选择的MCU时，将使用该MCU的数据库知识自动配置和定制专门针对该MCU的开发工具。这样可大幅缩短工具设置时间。自动化工具配置包括下列步骤：

仿真器设置

本步骤配置µ Vision仿真器为相应的MCU建模，包括默认时钟频率、指令集、寄存器集、内置ROM和RAM空间以及外围设备组合。

链接程序设置

本步骤构建一个链接程序可读内存映射或分散文件，它定义链接程序应将代码放到正确地址，并告诉MCU仿真器内存映射的布局（仿真器可写入非内存或只读内存）。

编译器和汇编器设置

本步骤设置编译器和汇编器，包括包含该MCU的正确头文件的路径。

将启动代码安装到项目中

本步骤预先将相关示例MCU启动代码的源代码安装到项目开始处，确保在重置后、执行应用程序之前将MCU置于明确定义的已知状态。

除此以外，自动配置可以帮助用户在第一次使用工具时设置几个实用默认值，其中包括调试编译、map文件的生成以及调用图等。µ Vision IDE中“目标GUI”的“选项”中可以显示所有这些相关选项，包括作为GUI生成的每个工具的GUI可更改选项以及命令行参数

集。图1显示，飞利浦LPC2148 MCU的目标配置及其片上内存映射和默认时钟频率的项目会从Device Database的MCU项中自动完成。除此以外，用户还可以在配置中添加其他配置选项，如其他板上内存位置，等等。

结论

在本指南开始处，我们声明在选择MCU时，嵌入式开发人员根据四个标准进行选择：功能、可用性、成本和熟悉程度。Device Database独特的功能使开发人员能根据这些标准作出明智的选择，他们可以：

- 在整个基于ARM内核的MCU范围内搜索特定功能。
- 快速确定经销商的MCU定价和可用性。
- 在熟悉的µ Vision开发环境中快速着手预配置的工作，以便为他们选择的特定MCU构建应用程序。

指南 2：为基于ARM内核的微控制器开发第一个应用程序

摘要

基于32位处理器的微控制器的强大功能使嵌入式开发世界得以向现有软件组件和行业标准高级语言转变。本指南描述ARM的RVMDK及其相关 RealView 实时库（RV-RTL）如何提供可配置的高性能软件平台，并以它为基础帮助您快速创建基于 ARM 内核的微控制器（MCU）应用程序。

介绍

基于32位ARM内核的微控制器提供的高性能可使用8位和16位微控制器无法提供充足支持的大规模、现成软件组件。嵌入式开发人员现在可使用高级语言（如ISO标准C和C++）编写其MCU应用程序。使用高级语言可以更轻松地支持运用现成软件组件，如实时操作系



统、文件系统和联网库。软件开发过程演变为将各种组件整合在一起，只需较少的组件内部知识或相应外围设备的知识即可快速构建嵌入式应用程序。

ARM RVMDK和RV-RTL提供一套软件组件，可满足现代嵌入式应用程序的诸多常见要求。此外，Device Database包含许多使用这些软件组件的示例程序，提供可改编为最终应用的高级模板应用程序。

目标配置

为嵌入式应用程序正确设置目标环境需要投入相当多的时间。如指南1所述，RVMDK可在项目开始阶段为选定微控制器添加特定启动代码，使开发人员的应用程序能在MCU处于明确定义的已知状态时开始执行。但是，特定MCU上运行的各个应用程序的内存要求和外围设备行为要求相差悬殊，提供的启动代码不可能符合开发人员需要的每种可行设置。

一种方法是开发人员使用一套标准的启动代码，逐步修改它直至达到MCU及其资源（如堆栈）需要的状态。虽然此方法肯定可以成功，但要耗费大量时间。RVMDK提供另一种方法，即使用配置向导手工修改示例启动代码，该向导提供映射到启动代码的MCU资源的高级视图。图1显示Atmel AT91M55800A微控制器外部总线接口上两个芯片选择寄存器的部分配置。开发人员可通过修改每项的值轻松启用和配置特定芯片选择寄存器。

此方法大大简化了启动代码的创建，因为完成配置向导时会自动生成启动代码。例如，完成芯片选择0（CSR0）项时会自动修改下表ARM汇编器常量声明相关的数值：

```
EBI_CSR0_Val EQU 0x010024A9  
EBI_CSR0_Val EQU 0x010024A9
```

该常量之后用于在启动代码中初始化外部总线接口。

配置向导的内幕

配置向导使用简单标记语言来定义用户在对话框中看到的内容以及用户的选择如何生成数字和文本项。为此，标记语言可以嵌入C、C++、汇编器源代码或任何文本文件中。这样可将配置向导用于使用数字字符串定义的任意配置。图2显示将配置向导用于配置目标系统堆栈大小的示例。图3显示该示例的底层标记语言。标记<h>和</h>定义堆栈项及其相关标签的起点和终点。<o>标记定义数字范围内的用户可编辑项。此示例中要编辑的实际项不明显，即标记（0x00000000）后的第一个数字字段。用户可通过在0后指定一个数字来定义字段偏移量。

软件应用程序库

基于ARM内核的MCU提供的高性能使它们能用于计算更密集的市场领域（相对于8位和16位MCU针对的其他市场）。成功填补这些市场空缺需要更复杂的软件。典型的8位应用程序可能包含一个实时操作系统（RTOS）和一些控制代码，而现代的32位MCU应用程序通常包含完整的通信栈，其复杂程度比低级RTOS高出一个数量级。

RV-RTL提供一套经常需要的软件组件，可以将它们轻松部署到基于ARM内核的MCU中去。该库包含一些为其支持的MCU的特殊外围设备专门优化的项，如TCP/IP网络栈、Flash文件系统、USB设备启动程序以及CAN设备驱动程序。开发人员可使用RVMDK中的配置向导快速配置组件。图4显示将配置向导用于配置USB2栈的示例。

RVMDK中的可配置对象包括RTX实时内核，RV-RTL中以源代码形式提供该

内核。这是一款全功能实时内核，提供许多现代RTOS中最常见的资源抽象，包括线程、计时器、队列、邮箱、信号量、互斥体、块池和事件标记。开发人员可使用内核来组合实时库中的组件，以便在目标系统上作为单独的通信任务来执行。

代码模板和应用程序示例

RVMDK包含几个针对各个受支持MCU的示例应用程序。开发人员可将这些示例用作他们的应用程序模板，无需深入了解其选定MCU即可着手开发。示例涵盖常见的初始开发要求，如设置中断、闪烁LED或将文本写入输出设备以及完成模拟数据采集和实时OS示例。

RV-RTL还随附几个示例应用程序，它们在现实世界典型情况下使用RV-RTL中的组件。它们可以用作整个嵌入式应用程序的基础。示例包括：

- 包含CGI脚本的嵌入式Web服务器
- SMTP 电子邮件通知
- Telnet
- USB 内存设备
- USB 人机接口设备
- USB 音频设备

结论

通过使用基于32位ARM内核的MCU开发微控制器应用程序，开发人员能使用之前8位和16位MCU无法实现的方式利用各种现成软件组件。基于ARM内核的MCU的强大实力和内存寻址功能无需手工编码汇编器或专用软件库。RVMDK和RV-RTL为开发人员提供了常见软件组件和示例库，可用作其应用程序的基础构建模块。使用内建的配置向导，可以在RVMDK的μVision IDE中为所需的应用程序轻松配置目标系统和软件组件。通过使用这些可配置组件并与实时内核相结合，开发人员可以更自信地将其应用程序迅速推向市场。

（未完待续）