

在MAXQ环境中编程

MAXQ2000微控制器的在线调试与程序装载功能与 IAR Embedded Workbench 开发环境相结合，为设计者提供C或汇编级的应用开发与测试手段。

MAXQ结构为应用编程者而开发。每个MAXQ微控制器都包含一个与微控制器内核紧密集成的硬件调试引擎。该架构中第一款芯片就是MAXQ2000，本文提供 IAR Embedded Workbench 与MAXQ2000评估板一起使用的实例与技巧。

MAXQ2000微控制器的在线调试与程序装载功能与 IAR Embedded Workbench开发环境相结合，为设计者提供C或汇编级的应用开发与测试手段。MAXQ2000基于硬件的调试引擎与自举装载过程在专用的JTAG端口运行，可以全面调试访问，而对系统资源的影响最小。

在线调试特性

与微控制器内核紧密集成的硬件调试引擎，用来控制MAXQ2000的调试特性。该调试引擎可以调用片上的固定用途ROM中的服务子程序，支持广泛的调试特性。

- 对集成程序闪存的读取访问。
- 对板上数据SRAM的读/写访问。
- 对16 x 16堆栈存储器的读取访问。
- 对所有MAXQ2000系统与外设寄存器的读/写访问。
- 单步(跟踪)程序执行。
- 最多四个地址断点，在代码存储器中特定的单元停止程序的执行。
- 两个数据存储器匹配断点，在数据存储器中特定单元被访问时停止程序的执行。
- 两个寄存器断点，在出现对特定系统或外设寄存器的写访问时(不能与数据存储器匹配断点同时使用)，并且写入寄存器的数据与特定值匹配时停止程序的执行。
- 密码匹配功能(用来解锁其余的调试功能)。

所有与调试引擎的通信都通过MAXQ2000专用的JTAG测试访问端口(TAP)接口完成，该接口与JTAG IEEE标准1149兼容。该接口由四个信号组成，由MAXQ2000端口引脚复用得到，定义如下：TMS(测试模式选择)——与P4.2复用；TCK(测试时钟)——与P4.0复用；TDI(测试数据输入)——与P4.1复用；以及TDO(测试数据输出)——与P4.3复用。

尽管JTAG TAP端口专门用于系统内调试与系统内编程，不过，一旦应用开发完成，传输JTAG TAP端口信号的四个端口引脚可以被释放用于其他目的。JTAG端口在复位后默认为激活状态，但是一旦运行，应用软件可以关闭端口，将四个相关的端口引脚用于其他用途。

所有与调试引擎的通信都通过MAXQ2000专用的JTAG测试访问端口(TAP)接口完成，该接口与JTAG IEEE标准1149兼容。

从MAXQ2000内核来看，JTAG接口与调试引擎异步工作。通过JTAG端口进行的通信不需要与MAXQ2000运行的时钟速率相同，不过对MAXQ2000来说，TCK频率的最大值被限定在1/8系统时钟速率以下。

在MAXQ2000执行代码时，可以通过调试引擎读取或写入断点设置。该模式被称为后台模式，在这种模式下，调试引擎独立于CPU内核之外工作。

为了完成其他操作，如存储器与寄存器的读写，调试引擎控制MAXQ2000的内核，并切换到执行固定用途ROM中的调试服务子程序。该模式被称作调试模式，该模式下，调试引擎中断了正常的程序执行。在这种情况下，用户应用程序被暂时挂起，等到调试函数执行完毕后继续执行，与中断服务子程序的处理方式相同。

由于在应用程序中JTAG TAP端口并不使用，因此组成JTAG端口的端口引脚可以被应用软件收回。调试功能所需的所有附加代码都位于固定用途ROM中，因此，调试功能消耗的系统资源只是少量的数据SRAM，以及一层程序堆栈(用来存储调用调试子程序的返回地址)。最多19个字节的数据SRAM(地址0x07ED至0x07FF)被保留用于调试服务子程序。若某个应用程序不使用在线调试，则这些数据SRAM单元可供应用程序使用。

通过 JTAG 进行集成闪存编程

JTAG TAP端口也可被用于自举装载功能，甚至在不使用调试功能时也有效。通过JTAG TAP接口设置3个配置位，然后从复位中释放MAXQ2000，将控制转移到位于固定用途ROM中内置的自举装载过程子程序。用来控制访问自举装载过程的配置位如下：

- SPE: 系统编程使能位(ICDF.1)。该位被置1时，MAXQ2000在系统复位后，执行固定用途ROM中的自举装载子程序。
- PSS[1:0]: 编程源选择(ICDF.3-2)。这几位的设置决定了自举装载过程的通信使用JTAG端口(PSS[1:0] == 00b)还是串口0 UART (PSS[1:0] == 01b)。

一旦这些位被置位并且MAXQ2000从复位状态中释放，固定用途ROM的自举装载过程通过选定的端口(JTAG或串口0 UART)开始与主机系统通信。在这两种情况下，使用的协议相同，并且都提供如下函数：

- 读取 MAXQ2000 的ID标题(识别固定用途 ROM 的版本)。
- 返回内部程序与数据存储器的大小。
- 读取、写入、验证与CRC校验集成闪存程序存储器。
- 读取、写入、验证与CRC校验内部数据SRAM。
- 密码匹配(解锁存储器读写指令)。

通过JTAG TAP接口设置3个
配置位，然后从复位中释放
MAXQ2000，将控制转移到
位于固定用途ROM中内置的
自举装载过程子程序。

当自举装载过程通过串口0 UART而不是通过JTAG端口通信时，必须使用JTAG接口将自举装载过程置为串行通信模式。不过，适当地设置SPE与PSS位，然后复位MAXQ2000(通过使看门狗定时器过期或使用外部硬件手段)，应用软件也可以在串行通信模式下调用自举装载过程。使自举装载过程被调用的方法必须由应用软件决定。

对于调试和自举装载功能的密码保护

基本的密码保护方案限制了对 MAXQ2000 的调试与自举装载过程函数的访问。必须先由主机系统提供该密码，才允许访问任何读取或修改存储器或系统与外设寄存器内容的函数。

密码长度为 16 个字或 32 个字节。密码的数值存放在内部闪存的 0x0010 至 0x001F 字节单元中。这些数值可以作为静态数组包含在应用程序中，或者只是存储在这些单元中的指令代码的数值。不管哪种方式，在装入应用程序时密码自动写入。若没有装入任何应用，密码的缺省值为所有的字都等于 0xFFFF。

即使密码未知，MAXQ2000 的内部闪存也始终可以通过自举装载过程擦除。这将有效地清除密码值 (所有字都变为 0xFFFF)，并允许其他编程或调试操作继续进行。在 32 字节密码数值没有匹配之前，密码保护简单地确保了不能从 MAXQ2000 中读取现有指令。

使用串口-JTAG 适配器模块

MAXQ2000 微控制器的集成开发环境 (例如 MAXIDE 与 IAR Embedded Workbench) 包含了支持与 MAXQ2000 JTAG 接口进行通信的软件库。不过，由于运行该软件的 PC 通常都不具备 JTAG 端口，因此，两个系统的接口需要一个硬件层。

MAXQ2000 评估板包含的串口-JTAG 适配器模块为这个接口问题提供了圆满的解决方案 (图 1)。PC 上运行的软件 (例如 IAR Embedded Workbench) 通过标准的 COM 串行端口与串口-JTAG 适配器模块通信。然后串口-JTAG 适配器模块与 MAXQ2000 的 JTAG 端口接口，将指令传输至自举装载过程或调试引擎。适配器模块不仅用来实现电平转换，也支持在不同电源电压范围运行的 MAXQ 微控制器，而且不需要 PC 为 JTAG 波形提供精确时序。

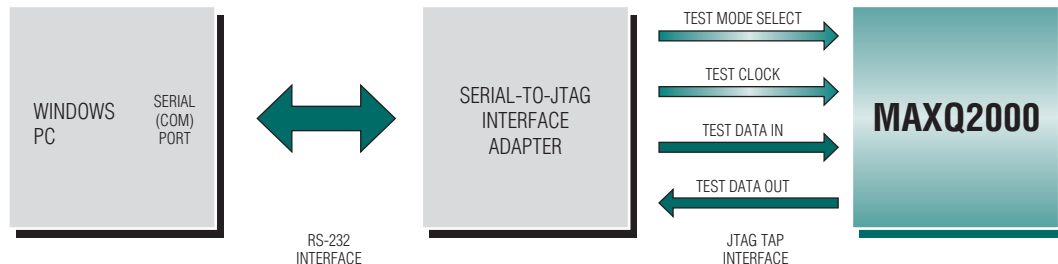


图 1. 串口-JTAG 适配器模块允许在 PC 上运行的软件访问 MAXQ2000 微控制器的 JTAG TAP 接口。

使用 MAXQ2000 评估板硬件

MAXQ2000 评估板为 MAXQ2000 微控制器提供了完整的硬件开发环境，包括如下特性。

- 为 MAXQ2000 内核与 VDDIO 提供板上电源。
- 可调的电源电压 (1.8V 至 3.6V)，可以用于 VDDIO 或 VLCD 电源。
- 为所有 MAXQ2000 信号与电源电压提供引出脚。
- 独立的 LCD 子板连接器
- 带 3V, 3.5 位静态 LCD 显示器的 LCD 子板

- 为串口0 UART提供包含流控信号线的全RS-232电平驱动器。
- 外部中断与微控制器系统复位按钮。
- MAX1407多用途ADC/DAC IC，与MAXQ2000 SPI总线接口相连。
- 1-Wire[®]接口，包括 iButton[®]接线柱与1-Wire EEPROM IC。
- 为端口引脚P0.7至P0.0的电平提供柱状图LED显示。
- 为应用程序的装入与在系统调试提供JTAG接口。

IAR Embedded Workbench 与MAXQ2000评估板以及串口-JTAG适配器模块相结合，可以完全访问MAXQ2000的基于JTAG的自举装载过程以及在线调试的特性。

为应用程序开发设置 MAXQ2000评估板与串口 -JTAG接口模块很简单。按照以下步骤简单地连接电路板。

- 1) 将5V DC稳压电源 (中间接线柱为正，±5%)插入串口 -JTAG电路板的电源插孔J2。
- 2) 将5V至9V DC电源插入 MAXQ2000评估板的电源插孔 J1。
- 3) 用一条直通式 DB9串行电缆连接串口 -JTAG电路板的 J1连接器与PC上的一个COM端口。
- 4) 用JTAG适配器电缆连接串口 -JTAG电路板上的 1 x 9连接器 P2与MAXQ2000评估板上的 2 x 6连接器J4。
- 5) 打开上述两个DC电源。
- 6) 在标准操作中，MAXQ2000评估板上的所有DIP开关都应处于OFF的位置。

使用 IAR Embedded Workbench进行应用程序开发

IAR Embedded Workbench开发环境为MAXQ2000提供了基于C语言或基于汇编语言的应用程序开发手段。使用上述对MAXQ2000评估板与串口 -JTAG适配器模块的硬件配置，IAR Embedded Workbench可以完全访问MAXQ2000的基于JTAG的自举装载过程以及在线调试特性。

IAR Embedded Workbench为MAXQ2000的应用程序开发提供了以下特性。

- 将编译完成的应用程序装入MAXQ2000的集成程序闪存中。
- 提供C语言或汇编语言级的单步(跟踪)程序执行。
- 显示代码、数据、硬件堆栈与固定用途ROM存储器。
- 调用堆栈跟踪。
- C语言或汇编语言级的断点设置。
- 显示与编辑所有MAXQ2000系统与外设寄存器。

创建与编译一个MAXQ2000项目

由于IAR Embedded Workbench包含对MAXQ2000微控制器系列的集成支持，创建一个新的MAXQ2000微控制器项目只需要几个具体的设置。

IAR Embedded Workbench 开发环境为MAXQ2000提供基于C语言或基于汇编语言的应用程序开发手段。

启动IAR后，从菜单中选择**File**，然后选择**New**。在**New**的对话框中选择**Workspace**，然后点击**Ok**。为该项目的工作空间键入一个新的名称(保存为“.eww”文件)，然后点击**Save**。

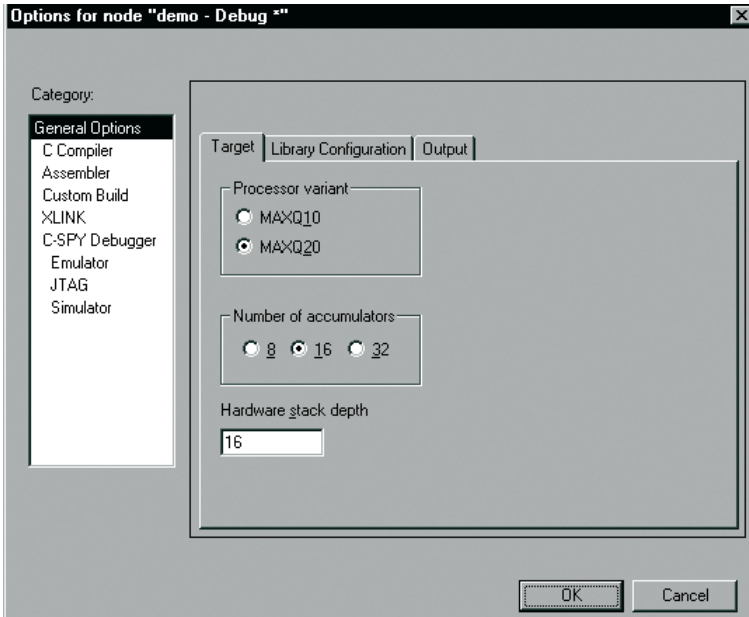


图 2. Options 对话框中的 General Options 部分允许用户指定处理器内核的类型 (MAXQ10/20), 可用累加器的个数, 以及硬件堆栈的深度。这里给出的是用于 MAXQ2000 的设置。

- 对于 MAXQ2000, Hardware stack depth 应当设定为 16。

在 Option 对话框中的 C-SPY Debugger 选择标签中, 应当为 MAXQ2000 选择以下设定 (图 3):

- 将 Driver 设定为 JTAG, 说明通过 PC 的 COM 端与串口 -JTAG 接口电路板连接。另两个可能的设置为 Simulator (用来与 MAXQ2000 软件模拟器一起运行) 与 Emulator (用来与 MAXQ2000 在线仿真器系统一起运行)。

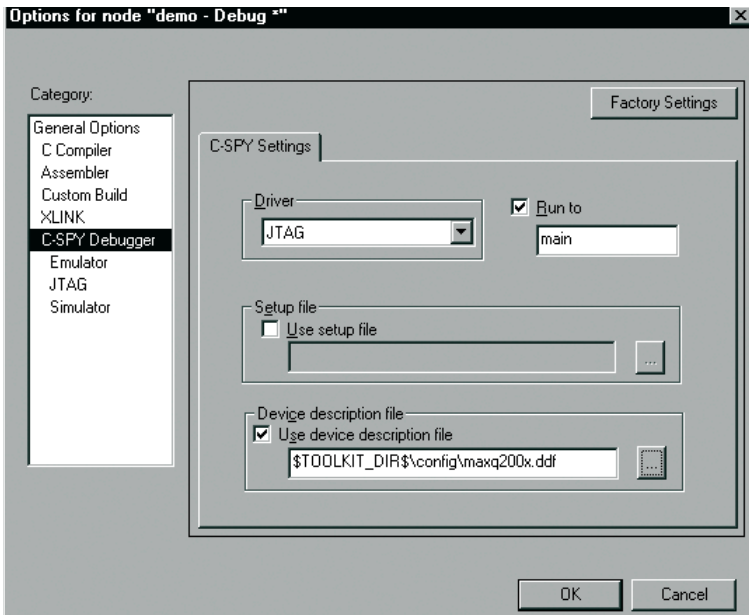


图 3. Options 对话框中的 C-SPY Debugger 部分允许用户指定调试对话。这里给出的是使用串口 -JTAG 适配器模块调试 MAXQ2000 的设置。

IAR 中的调试操作

启动调试过程后, Step Over (F10), Step Into (F11) 与 Step Out (Shift+F11) 可以用来跟踪项目的 C 语言代码。若要运行代码, 则从菜单中选择 Debug, 然后选择 Go, 或按 F5 键。

工作空间窗口打开后, 从菜单中选择 Project, 然后选择 Create New Project。对于新项目, MAXQ 工具链为默认状态。为新项目键入文件名 (保存为 *.ewp 文件), 然后点击 Create。

接下来, 从菜单中选择 Project, 然后选择 Settings。出现的对话框中显示的是新创建项目的设置, 如图 2 所示。

在 Options 对话框中的 General Options 选择标签下, 应当为 MAXQ2000 微控制器选择以下设置。

- 由于 MAXQ2000 具有 MAXQ20 型的内核, Processor Variant 应当设定为 MAXQ20。
- 对于 MAXQ2000, Number of accumulators 应当设定为 16。

- 应将 Use Device Description File 逻辑框选中。器件说明文件 (*.ddf) 应当是提供给 MAXQ2000 微控制器的文件 (maxq200x.ddf)。该文件定义了 IAR 环境使用的特定 MAXQ 微控制器的存储器空间与外设寄存器组。

在 Options 对话框中的 JTAG 部分, Command line options 区段包含了与串口 -JTAG 电路板连接的 PC COM 端口。图 4 所示为连接至 COM 端口 1 的选项设置。

为项目进行选项设置之后, 选择 Project, 然后选择 Add Files, 将 C 语言代码文件添加到项目中。添加了项目文件之后, 选择 Project, 然后选择 Make 来编译项目。接下来, 选择 Project, 然后选择 Debug 来启动调试过程。这样将编译后的项目通过 JTAG 接口下载, 并将 IAR 置为调试模式, 如图 5 所示。

将光标放置在源代码的指令行上，并点击工具栏中的 **Toggle Breakpoint** 按钮，就可以设定或清除地址断点。一次最多可以设置四个地址断点。

Memory 窗口可以显示 MAXQ2000 的 **Code** (内部闪存)、**Data** (内部 SRAM)、**Hw stack** (内部 16 层堆栈)，以及固定用途 ROM 存储器。存储器显示可以被设置为字节、字，或双字格式，也可以按 hex (对所有宽度) 与 ASCII (对字节宽度) 格式来显示。

Register 窗口用来显示 MAXQ2000 的系统与外设寄存器。它们按以下逻辑分组来显示：

- **CPU 寄存器**：累加器与累加器控制寄存器、数据指针与数据指针控制寄存器、指令指针、循环计数器，以及程序状态标志。
- **中断控制**：中断向量、模块屏蔽，以及标识寄存器。
- **周期**：显示已经执行的指令周期数。
- **并行端口**：用于 P0、P1、P2、P3 和 P4 的输入、输出以及端口方向寄存器。
- **外部中断**：用于外部中断的使能、边沿选择，以及标志寄存器。
- **定时器**：用于定时器/计数器 0 至 2 的寄存器。
- **串行端口**：用于 SPI 与串行端口的控制与缓冲寄存器。
- **乘法器**：与硬件乘法器模块相关的寄存器。

点击寄存器数值并键入新值，可以实现对可写寄存器的编辑。点击寄存器名称旁的加/减符号，可以实现寄存器中独立位或位区域显示的扩展或压缩。

结论

IAR Embedded Workbench 高级的、基于 C 语言项目的环境集成了 MAXQ2000 的底层调试接口，允许在 C 语言或汇编语言代码层上执行精细的调试。MAXQ2000 的内置的调试与在线编程的特性，以及对系统资源的底层影响，允许应用程序开发过程以及已完成项目的最终发布使用同样的硬件设计。

1-Wire 与 iButton 是 Dallas Semiconductor 的注册商标。

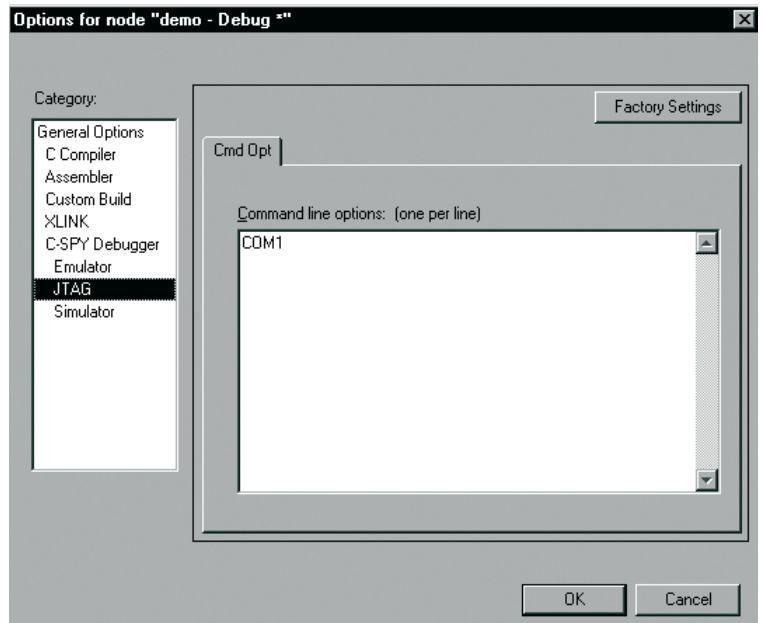


图 4. Options 对话框中的 C-SPY Debugger (JTAG) 部分允许用户改变专门用于串口 -JTAG 适配器模块的设置。这里给出的设置用于与 PC 端口 COM1 相连的串口 -JTAG 适配器。

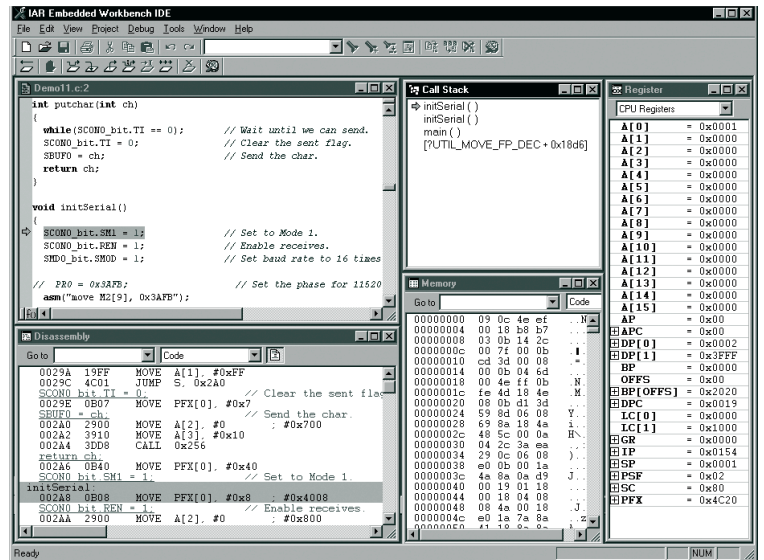


图 5. 由于使用了串口 -JTAG 适配器模块，IAR Embedded Workbench 可以在 MAXQ2000 上实现单步执行，并可以实现对片上存储器与寄存器数值的读取与修改。