

设计之源。加速。简化。

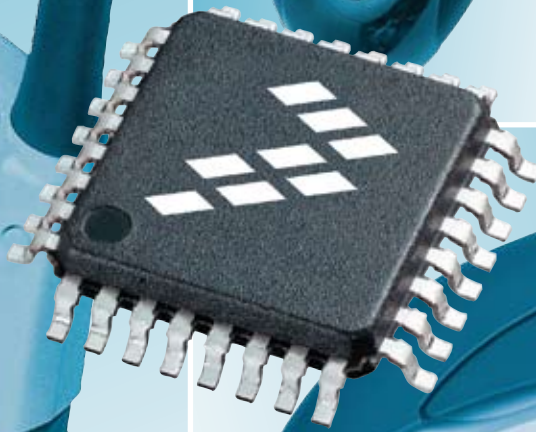
2007年夏第2期

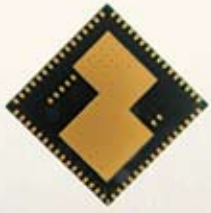
文档号: BR8BITBYNDBITS
Rev 1

超越比特位

目录

- 控制器联合体
- ZigBee® 与低成本无线网络
- 低功率考量





从零到实现ZigBee, 只需几分钟!

BeeKit™无线连接工具包

实际上,您只需30分钟就可以构建一个小型网络并开始运行。秘诀就是我们的开发软件环境及其高度直观的图形化用户界面(GUI)。它使您的团队可以在一个即时且集中的环境中进行协作——无论他们是在另一幢大楼里还是在另一个大陆上。在最终确定软件结构之前,团队成员可以自由而独立地进行试验和创新。不管是复杂的网状网络,还是简单的点到点网络,开发时间都可缩短数周。它以最快的方式将ZigBee设计推向市场。



从freescale.com/zigbee上获取您的BeeKit



超越比特位

第二期实现了界限的突破

欢迎阅读第二期《超越比特位》! 在一这期中，我们扩大了讨论的范围，涵盖了在飞思卡尔产品系列中采用的多种技术，包括8位微控制器(MCU)、32位ColdFire控制器、我们曾获大奖的16位数字信号控制器(DSC)及ZigBee® 无线技术等。此外，我们还增加了一个来自设计联盟合作伙伴的内容，以及飞思卡尔客户如何用我们的产品和服务取得成功的实例。

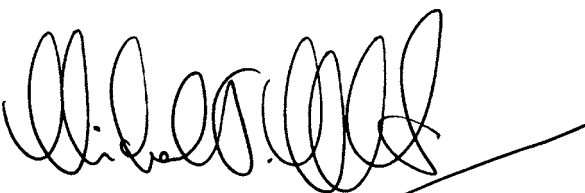
本期所包含的技术文章有一个共同的主题，那就是为您提供实际的方案和可以解决问题的参考资料。其内容对各个层次的技术人员都非常适合，而且我相信，不管是对于提高专业技术水平还是开发新设计，您都会发现它是有用的。

《超越比特位》第二期中的许多文章可帮助您设计成本敏感的应用。一些文章所讨论的激动人心的新技术可以帮助您达成设计目标。还有一些文章着重介绍了飞思卡尔的控制器联合体(Controller Continuum) — 行业的第一个也是唯一的8位和32位兼容架构的路线图。

无论您为工业、消费类还是汽车市场开发产品，我们所拥有的广泛产品系列必能满足您的需求，帮助您获得丰厚的投资回报。因此，请深入其中，您会发现飞思卡尔的创新技术是如何帮助您简化设计过程的。

本期与第一期相比有了很大的改进，其中很大一部分应归功于你们的重要建议。欢迎阅读这一期并继续提供宝贵的反馈意见。最重要的是，感谢您在现在或以后的设计中想到飞思卡尔。

顺颂商祺！



Mike McCourt
副总裁兼总经理
飞思卡尔半导体微控制器业务部



我们期望收到您的来信！

通过freescale.com/beyondbits发送您对产品看法、诀窍或问题。



超越比特位

目录

控制器联合体	6
Jeff Bock和Joe Circello	
成功之路	10
Robert Lewis	
低功率考量	13
Inga Harris	
超低功率无线设计	17
Radomir Kozub和Pavel Lajsner	
JTAG TAP的有效控制	22
David Baca	
加速创新	24
Lech Olmedo和Daniel Torres	
DSP56800E中的高速32位除法	27
David Baca	
马达控制及电源设计	31
Daniel Torres	
最新S08/RS08工具	36
Inga Harris	
ZigBee®与低成本无线网络	41
Pattye Brown	
针对成本敏感应用的微控制器	48
Dr. Martin Mienkina	
如何构建简单的无线应用	54
Juan Cazares	
飞思卡尔设计联盟计划	58
Gary Streber	
采用8位MCU的BLDC马达控制	60
Pavel Grasblum	
LCD驱动的诀窍	64
Ronald Gonzalez和Tatiana Orofino	
SMAC无线连接的简单实现	67
Gonzalo Delgado Huitrón和Jaime Herrero Gallardo	
DDR存储器的比较和综述	70
Joaquin Romo	
如何使用HCS08的TPM模块	74
Pavel Grasblum	
产品概览	82
Jeff Bock和Joe Circello	

控制器联合体

解决从8位到32位的过渡问题



长期以来，从8位到32位MCU的过渡对设计人员来说一直是一个严峻的挑战。许多因素都要求建立一个不同功能的处理器的统一平台；然而，现有的8位和32位的专业知识只能了解各自领域用户的需求。本文将介绍飞思卡尔如何认识这个过渡问题，修改硬件设计的

理念，并最终定义了具有兼容的封装、外围设备和工具的8位和32位新产品。这种硅芯片设计方法有望简化并加快大量的最终产品的性能升级，使客户能够更快地将产品推向市场。

性能路径

微控制器的应用范围非常广泛，从简单替换采用固态技术的传统机电功能，到已经数年发展、如今需要高级32位MCU的极其复杂的控制应用。虽然有几家公司能提供可以支持多种应用的单一解决方案，但没有哪家公司提供可以从8位到32位MCU实现轻松过渡的产品。目前，对低端8位MCU和中端16位MCU的需求正在迅速增长。如果这些应用需要更高的性能，答案可以在多种低成本32位架构中找到。然而，对于从8位和16位MCU进行移植的用户来说，他们会感觉32位架构有很大的不同。这通常需要新的硬件工具，因此移植就成了一项昂贵而耗时的的工作。

通过充分利用飞思卡尔在8位和32位MCU领域的丰富经验，我们定义了一种可帮助用户简化过渡过程，并重复使用软件和工具的方法。中级性能的MCU通常是根据价位、功耗、外围设备和/或特定应用所需的性能水平定义的，而不是根据比特位定义的。在大多数情况下，客户并不在乎比特位，他们只关心能否解决系统问题。一个8位到32位的联合体可以在这段空白上架起桥梁，让客户从当前的系统需求入手，同时规划未来的改进计划，且无需进行重大修改，也不必为先期的工具和软件再追加投资。

内核

8位与32位内核之间的无缝过渡从8位级产品开始。成熟的HCS08(S08)内核奠定了8位性能的基础并配备了丰富的外围设备。新近开发的RS08内核将S08中央处理器(CPU)的大小缩小了30%，并为联合体提供了更大的产品范围。该内核的目标是用电子控制代替机电设备的低成本应用，在管脚数较少的封装中配置不到16K的闪存，而且所用的指令更少，编码更紧凑、更高效。这种体积更小、更高效的内核充分利用了为广阔S08应用开发的外围设备。

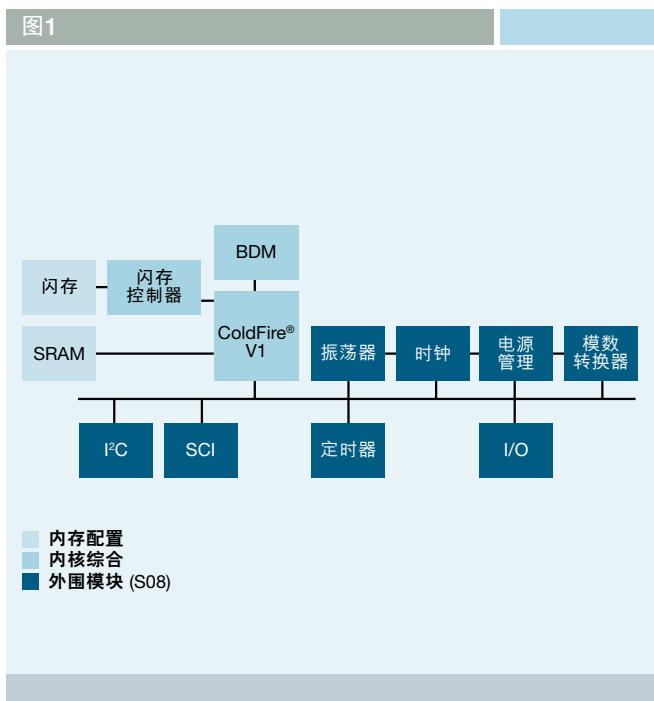
在高端产品方面，先进的32位ColdFire V1内核可以提供转换链条上缺失的环节。V1平台由V1内核、本地内存(闪存和SRAM)、后台调试模块(BDM)和总线桥组成。总线桥提供内核的本地高速总线与辅助外围设备模块之间的互连，它还是将一种协议转换成另一种协议的总线接口。

一般情况下，V1平台部分的运行速度是外围设备模块的两倍。除了提供总线协议转换功能外，总线桥模块还可以用作不同时钟区之间的接口。

采用成熟、易用的8位外围设备，加上16位内部架构和32位内核，8位和32位产品系列可以方便地连接起来，为工程师提供设计灵活性。

从8位迁移到16位MCU的决定是基于多种考虑的。根据现有的处理技术，32位V1内核的价格与16位内核大致相同。然而，可提供的性能却要高得多。

图1



通用外围设备

32位的V1 ColdFire通过IPS总线连接到8位S08外围设备模块。总线桥模块使V1平台部分能用高达外围设备模块2倍的速度运行。

可互换的内核

8位S08内核和32位V1 ColdFire内核都可与众多的S08外围设备一道使用，甚至可以封装在引脚兼容的封装中。单引脚的BDM使8位和32位MCU可以使用同样的CodeWarrior Development Studio开发套件。

兼容的外围设备、封装和工具

多种因素促成8位与32位架构之间的无缝移植。进程从内核开始，并涉及外围设备、封装以及开发工具。

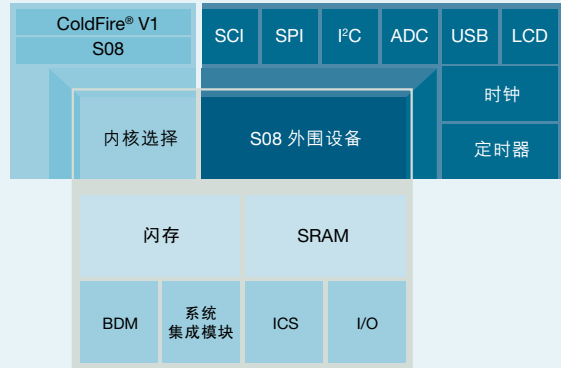
从10多年前首次实施以来，ColdFire的指令集架构(ISA)发生了很大变化，从最初只在一定程度上支持8位和16位操作数，面向32位操作数的优化体系演变成扩展并改进8位和16位操作数处理支持的架构。对于V1类的应用，这对从8位和16位应用移植的用户尤其重要。在处理器内核层，指令集是否能直接解决涉及8位和16位操作数的高效处理问题，是非常重要的。

V1内核微架构实施了大量的优化，以提高那些处理8位和16位操作数的指令的性能。V1不仅支持这些指令类型，它们在V1通道中的执行也是优化过的。在有大量数据被用作8位和16位操作数的应用中，对这些指令的支持和快速执行时间可以实现大幅度的性能改进。

“控制器联合体”可以充分利用现有外围设备的知识产权(IP)。外围设备模块连接通用总线的结构可以确保S08 IP的完全重复使用。结果，不管是飞思卡尔还是用户在8位产品上的投资，在移植时都会得到保持，包括外围设备驱动软件的重复使用。虽然可能还需要考虑定时和其他一些问题，但这种方法确实可以使现有的外围设备驱动软件继续发挥作用。

后台调试模式(BDM)曾是保持相同引脚数的困难之一。传统ColdFire中的BDM是采用三针串行接口(时钟、数据输入、数据输出)实现的，还用第四根针来提供断点功能。这种BDM支持连续插入读/写寄存器和读/写内存命令的功能，还可以运行和停止处理器等。由于S08用于更低端应用中，管脚数也非常有限，所以传统ColdFire内核的调试功能被重新映射到针对V1的单引脚接口上，与S08完全一致。有关调试的改变的第二个部分与ColdFire内核对实时追踪的支持有关。

图2



对于V1来说，必须修改实时追踪操作。CPU可以输出处理器状态信息、地址和调试数据等。然而，V1不像V2和其他ColdFire MCU一样立即输出这些信息，而是采用了一个PST(处理器状态)和调试数据追踪缓冲器。利用这个方法，用户就可以设置开始和停止条件来记录追踪端口发出的信息。这些信息被收集、压缩并载入追踪缓冲器中。然后，用单引脚的BDM就可以读取跟踪缓冲器中的数据了。这是一个非常有用的概念，由此可以捕捉跨越数百个机器周期的活动，从而保留了追踪功能，只是把它们映射到采用单引脚接口的技术上了。

8位到32位过渡的最大挑战是工具。主要的开发工具必须是可互换的，并且在移植到32位时，要给8位用户提供同样简洁的外观和感觉。采用一种新版本的CodeWarrior®开发环境软件，用户可以用完全相同的工具、连接线和CodeWarrior工具，直接将8位S08拿掉并用32位ColdFire V1内核替换。用户只需简单地重新编译一下用C语言开发的代码即可。

用户的工作有何不同

利用飞思卡尔的“控制器联合体”，起初采用S08微控制器的客户可以比以前更轻松地从8位架构过渡到32位架构。通过过渡，用户可以获得远远高于8位S08内核的32位内核的处理功能。以相同速度运行时，32位V1估计可提供高达8位S08架构十倍的性能。与RS08相比的性能改进幅度就更大。

如此大幅度的性能改进可开辟新的应用领域，而且可以改变一个公司规划其长期产品路线图的方式。这样一种基于简便易用的32位架构的方法，非常适合那些寻求轻松获得32位性能的8位用户。

这种“联合体”的方法对那些寻求降低功耗和应用成本的32位用户同样有效。V1采用了先进的低电压、低功率流程，不管是在待机还是运行模式下都能提供一个非常低功耗的32位MCU。此外，它的功率密度预计将是独一无二的，可创建很多新的应用。

由于ColdFire V1 MCU产品具有高达S08器件10倍的性能，因此有可能在某个应用中大幅提高产品的性能而不必提高频率。对于那些对EMC敏感的应用，如家电和洗衣机，这可能会是一个重要的优势。我们已经与很多客户讨论并验证过这些方法。一旦设计人员在设计过程中看到采用这种新方法的可能性，他们会立即想到可以应用于未来的设计的新途径。

千里之始

通过对客户的反馈意见采取行动，“联合体”构建了扩展的8位S08以及32位ColdFire。这种反馈导致了大量的改进，包括指令集、硬件(包括调试功能)和这些器件的开发工具，这一切使向上过渡变得十分简单。此外，飞思卡尔正在推出Flexis™系列——首批管脚、外围设备和工具都兼容的、处于控制器联合体的“连接点”上的8位和32位微处理器。MC9S08QE128和MCF51QE128分别采用S08和ColdFire V1内核，是Flexis系列中将这种兼容性路线图变成现实的首款产品。

在大部分嵌入式控制设计中，很少是只有一个重要因素来左右设计决策。决策一般是根据许多较小的，但也很重要细节的累加来制定的。8位到32位的“控制器联合体”可以为系统设计人员提供独特的性能，让他们可以采取“穷尽一切”的理念来优化功率、大小、性能、外围设备使用和开发工具等等。此外，利用无缝地向上或向下过渡的功能，该“联合体”可给公司提供长期规划的能力，进而改变许多行业的产品路线图。

参考文档

飞思卡尔“控制器联合体”

www.freescale.com/files/abstract/overview/TSP8773_CONTINUUM.html

飞思卡尔Flexis™ MCU系列

www.freescale.com/flexis

Jeff Bock是飞思卡尔MCU部门的全球产品营销经理。Jeff主要负责针对消费和工业领域的产品的启动和推广活动。他在半导体行业拥有10多年的经验。

Joe Circello是飞思卡尔半导体TSPG产品部的MCD处理器/平台设计师，负责通道组织、控制结构和性能分析。他在摩托罗拉/飞思卡尔工作超过15年，一直担任MC68060和ColdFire系列产品的首席设计师。

成功之路

嵌入式处理器和C



这个项目就像其他项目那样开始——有一个合理的营销计划、功能规范以及可行的时间表。如果不是发生了那件不可避免的事，一切都会按计划运行。可预测的变化范围...“如果可能，我们需要加上它；倘若我们加上了，我们就会真正改进产品，它不会很难的。毕竟，它只是软件。”

且慢，让我们回顾一下刚开始的情况——项目实施之前。项目很简单，选择一个频率，根据用户选择的设置将它转换到另一个频率，然后在某种条件下输出。采用一个简单的8位微控制器就可以实现这一点，但我们想要额外的调试功能。当时908只有一个断点，因此我们选择了HCS12系列。我们不想用汇编语言编程，因此C语言似乎是唯一合理的选择，也是一个好方法。汇编器可能会很好用，但是由于命名问题和寄存器寻址错误引起的编程一致性标准的管理，以及对调试结果的管理造成的开销不值得我们冒额外的风险。

接下来，我们开始寻找良好的ANSI编译器。我们考察了CodeWarrior工具包，但考虑到项目的大小以及要执行的有限功能，我们认为把大笔的预算花在工具包和全是开关

和选项的大堆文档上，似乎不太合理。这意味著仅仅在了解交互式开发环境(IDE)上就要花费大量的时间。

因此我们开始考虑免费的第三方C编译器。免费的编译器似乎足够稳定，然而很明显，我们需要大量的专业知识来设置系统以进行编码和链接，而我们唯一可以得到的支持就是去网上论坛。当时，我们没有足够的时间和背景知识来处理如此之多的未知风险，因此我们只能将注意力放在看起来比较好的第三方替代产品上。论坛中的用户对整个ANSI C编译器的稳定性和支持赞赏有加，因此我们决定使用所选的产品。

有了编译器，下一步就是调试器。购买2万美元的在线仿真器(ICE)一定是很爽的，然而它看起来与我们300美元的编译器和预算太不相称了，因此我们很快排除了它。我们开始寻求不需要“在线仿真器”的产品。用评估模式运行了这个产品几天之后，我知道我们做了一个正确的选择。该产品拥有出色而合理的输出接口、稳定而且没有冲突、意外停机、崩溃、锁死或其他常见的问题。我们相信，这种产品将会非常好用。

在开始之前，我们还需要将另外几种工具结合起来。我们需要一个多版面编辑器、能跟踪修改的源代码保存库以及用于交叉参考和符号查找的源代码分析系统。这个用于源代码分析的开源产品是一个令人惊异的免费产品，具有许多功能，但遗憾的是没有相关文档。

现在，您可能看到了我们以前未能看到的前进方向。我们有效地构建了自己的IDE，因此也要负责集成和支持每一个方面。因为没有说明书，所以我们需要搞清楚如何让它运行。事实证明，这不是一件小事。

正如所料，让编译器和调试器运行花了很长时间。调试器是好的，但这个编译器是新版本的，因而在将目标代码正确写入闪存，以及正确地匹配分页寄存器的读写方面存在一些问题。不幸的是，问题发生时，编译器供应商正在度假。最后，我们得到了来自调试器团队的良好支持。他们找出了编译器中的不一致性，等到那个供应商回来时，编译器已经变得可以正确载入闪存了。

有一段时间一切都运行得很好，代码正在设计，有关命名规则、函数调用等方面的编码标准也或多或少地制定出来了。然后，第一次变化出现了。我们要去掉简单的BCD开关，用一个4行16字符的LCD和一个选择不同选项的按键来替代。（“什么选项？BCD开关没有选项，它只有开关！它们只是开或关，只在启动时读一下！我们需要显示和开关...”）

现在我们必须编写子程序来建立液晶控制寄存器、过滤地址和数据、编写定位字符的所有原代码，构建字符行，还有大约40个其他与显示器有关的子程序。做完这些之后，我们还要增加一个中断处理程序来运行5路开关。另外还需要一些子程序来跟踪菜单中光标处在哪一行。现在有了一大堆的功能，每个功能中有很多行，因此我们需要一个菜单来跟踪选项。

代码变长了，然而在下一个要求出现之前还是可管理的，这就是用户保存选项的方法。我的回答是：“如果您使用BCD开关，您就不需要保存任何东西，您只要看开关就可以了。”这个建议被拒绝了，然后我们就开始考虑采用串行数据连接。但是这样就需要一个PC应用程序。我们用面包板搭了一个串行连接电路，并用先进的第四代工具做这个PC应用程序，这之后我才知道串行传输本质上就极易出错，完全不适合我。如果失去同步，系统将会挂死；或者如果它错过一个字节，一切都会乱套并且做怪异的事情。唯一的解决方案是大量的握手软件。通过

大量工作之后，我们决定放弃使用与PC的动态连接，另外采用一个带FAT文件系统的SD卡。一切也从此开始恶化。

SD卡将目标代码的输出增加了约20KB，并且出现了大量的问题。调试器开始显示代码会跑飞，进入完全无关的区域。在看似随机的时间，运行程序会跳到无关的函数。我们整天整天地编写代码来试图跟踪可能的堆栈问题，还有更多时间花在试图捕捉可能的错误中断上，结果都没有成功。最后，在调试工具的作者的帮助下，我们发现由于函数引用的顺序，以及引用函数时是内嵌式的而不是链接式的，编译器生成了错误的符号表和链接。编译器供应商为我们提供了一个变通方法，我们的业务恢复了运行，尽管有些不尽如人意。我们艰难地等待著永久性的解决方法，然而至少我们能够继续前进。

接著出现了字符显示“太滥”的说法，认为这降低了产品的价值——因此我们需要图形显示。无需非常奇特。它可以是单色的，只要比我们现在的字符显示精细和时髦就好。我的答复是：“BCD开关既时髦小巧、又颜色丰富”，结果被认为既没有建设性也没有用处。

由于添加了带有字体表的图形子程序，目标代码又增加了20KB，代码执行过程中也出现了更多问题。编译器似乎在设置分页寄存器上遇到了的问题。编译器供应商已在尽其全力解决这一问题来支持这个产品了，然而论坛上的交流帖子显示，他的公司将大量的时间花费在为另一家制造商的产品开发新编译器上了。

这是压垮骆驼的最后一根稻草；我们最后不得不承认，如果要继续使用我们开发出的代码库，就必需一个能确保稳定性的产品。唯一的选择就是重新考虑使用CodeWarrior工具包。我们不会天真地认为CodeWarrior IDE没有自己特定品牌的问题，然而我们的确知道，只有像飞思卡尔这样生产处理器并提供工具包的大型跨国公司，才会拥有非常强大的动力来保持工具包的最新性和可操作性。

因此，我们购买了完整版的CodeWarrior IDE，但没有购买完整版的Processor Expert™ Bean。该版本使我们可以构建各种规模的代码库，它还拥有一些基本的Processor Expert目标代码。当时我们对Processor Expert beans一无所知。看起来似乎只要掌握了CodeWarrior工具包就可以开始使用了。

我浏览了一些飞思卡尔网站上的培训课程，过程似乎没有我最初设想的那么令人生畏。接著，我开始转换代码。

我没有调节过任何编译器或连接程序开关，只是使用默认设置。

大约两天后，我们完成了代码的编译和链接，没有出现任何错误。由于我们曾经使用的编译器是兼容ANSI的，因此几乎没做什么修改。主要的修改在于文件系统的命名、那些与CodeWarrior中定义端口和ECT配置的文件相冲突的Pragma声明和构件。原来的文件系统使用了几十个ANSI库预留的函数，如fopen()和fclose()。它们很容易修改，只要重新命名函数调用为fopen_imn即可。Pragma的修改非常直观也非常容易。

我们加载了代码并开始运行，令我们惊异的是它运行得十分顺畅——第一次就这样。不仅如此，由于CodeWarrior IDE是一种优化编译器而我们的旧编译器不是，因此CodeWarrior代码容量要小30%，而且明显快得多。在我们以前的代码库中所遇到的所有问题和Bug都消失了。而且还有一个惊喜，我们的老朋友(那个非在线的仿真器)调试器可以用CodeWarrior的ELF输出运行。因此我们既可以使用CodeWarrior调试器，也可以使用以前选择的工具进行调试。目前我们使用的是X-Gate处理器，因此我们需要更经常地使用CodeWarrior调试器了。

以事后之见，您可从中学到什么？尽管我做大型控制系统项目已经有30多年了，但我仍然陷入了我经常警告别人注意的小陷阱中。如果不用我的钱，我会购买最好的工具和所有其他必需的东西来确保以最小的风险尽快完成项目。如果用我的钱(或部分)，我就会花时间挑选较低成本的工具，因为我相信我的经验能使它很好地运行；而且在一定程度上的确是这样。然而当涉及其他人员时，有可能出现你无法控制项目结果的情况。即使以非常低的费率计算我的时间，也比购买CodeWarrior工具支付的费用要高出好多倍。此外，这还不包括情感方面的投入或项目延误造成的损失。有时，大量投资购买工具可能不可行。然而，有了免费的支持高达32KB的程序大小的CodeWarrior工具包，以及优化得很好的编译器，我们可以构建以前不太合适的很多项目。然后，一旦项目足够大的话，应该就会有能够负担额外开支的资金了。

如果您依靠的是一个拥有较小客户群的小供应商，这个供应商可能会受到他们不能控制的因素的影响。例如，主要员工的流失、疾病或收入的变化都有可能影响这些供应商的交付能力。

这可能不是一个在所有项目中都值得考虑的因素。这是否影响您的决策过程取决于最终的用户。如果您的产品是同一类型，而且您可以在本地通过变通方法迅速地支持客户，那么可能根本就没有风险。您可能从未遇到过编译器和连接程序的Bug，或是因为您的代码库较小，或是过去的项目经过了全面的测试，所以新代码可轻松完成调试和隔离。然而，如果您是构建的是一个将用于实时控制系统或大规模生产的产品，一旦在最终用户方发现错误，会造成大规模的收入损失以及信任度的丢失。这种类型的项目需要即时响应，所以重新考虑考虑一下您的“保险策略”可能是一种比较明智的做法。也就是说，当您为经过证明、有支持的产品支付额外费用时，您可以升级支持问题并得到响应。这可能不能立即解决问题，但总比被告知“我们之前从没遇到过这种问题：看看论坛上是否有人能提出相关建议”要好得多。那是一种危机中的茫然无助的感觉。

我是否提到了我们穿过Processor Expert beans的朦胧地带的经历？没有？那么，下次再说吧。

Robert Lewis是iMn MicroControl公司的工程师。他拥有电子工程专业理科学士学位和硕士学位，专业是基于微处理器的系统。

消费和工业应用的 低功率考量

消费和工业市场一直要求新产品设计具有更高的电源效率。每一代新的便携式产品都需要提供更长的电池寿命，而大型产品(如白色家电)则需要更高的能效比来保持竞争力。在许多情况下，用电子器件替代机械部件可以改善电路板的功耗，而另外的软件控制可以精细地调整应用的功率性能。

产品器件的发展使工程师可以使用新的设计理念和技术来克服早期产品的功率使用限制。由于以下两个方面的原因，微控制器(MCU)成了实现功率降低创新的关键：第一，MCU本身就消耗功率；第二，MCU控制消耗功率的其他系统器件。因此，MCU制造商需要将极低功耗的特性结合到他们最新的产品设计中。

本文着重描述了飞思卡尔采取哪些措施来满足这一市场需求，同时还介绍了降低功耗所采用的方法和增加的特性。最重要的是，它介绍了如何在应用设计中充分利用这些方法和特性。

通过在最新的低功耗8位MCU即MC9S08QE128上运行一个通用应用，重点介绍节能的实现。本文将通过逐一开启S08QE128上的每种节能特性来显示各个特性对应用功耗的影响。

功率与性能

系统设计人员总是要在适合其应用需求的功耗(Amp)和性能(MIPS)之间寻求平衡。一般而言，器件的时钟频率越高，它们所消耗的能量就越多。在大部分时间都在运行的应用中，以系统能够容忍的最低速度运行的解决方案就是最高效的方案。这种系统的一个实例就是电机控制器。如果其主要功能是通过模数转换器(ADC)通道转换信号，那么根据Nyquist定律，ADC的时钟频率就不能低于最小采样频率，这样就限制了MCU的最低运行速度。

在许多情况下，应用程序会通过尽可能快地执行预定任务，从而使MCU快速回到极低功耗的状态来实现整体功耗最低的目标。例如烟雾探测器，仅需每5秒钟唤醒

一次，进行读取、作出决策然后回到休眠状态。这种方法在电池供电的设备中很常见，此类系统在大约99%的时间里都处于休眠状态。

为了实现这种系统，飞思卡尔结合了快速时钟切换和快速唤醒程序来帮助应用设计人员实现其功耗目标。

烟雾探测器案例分析

图1和图2分别显示了烟雾探测器的结构框图和功率与时间关系示意图。MCU有一路来自烟雾枪的主要输入，还有两路输出：报警器和发光二极管(LED)。MCU内的实时时钟(RTC)模块可以定期从“停止2”模式(稍后描述)中唤醒设备，以便使运放读取并处理数据，并决定LED和报警器的状态。然后，MCU重新进入“停止2”模式以降低功耗。HCS08在“停止2”模式下的电流略低于ColdFire v1的同等部分，但S08 CPU的内置数据处理功能也较弱。图2对比了两种器件的功率和时间方块图，帮助您选择最适合您的应用需求的解决方案。在任务执行速度最为重要的应用中，MCF51QE128 ColdFire V1器件可能是更好的解决方案。这两种产品间的管脚兼容性意味著从一种策略到另一种策略的迁移可以迅速而轻松地完成，因为它们可以使用相同的电路板、软件和工具。

图1: 烟雾探测器结构简图

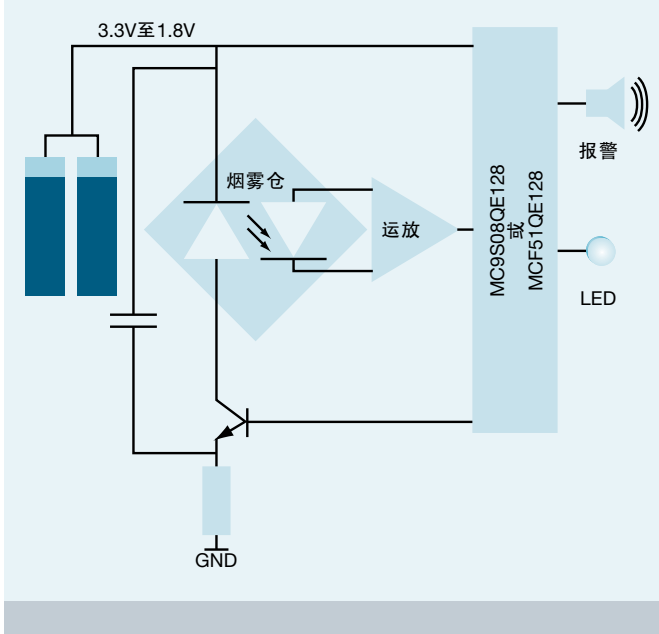
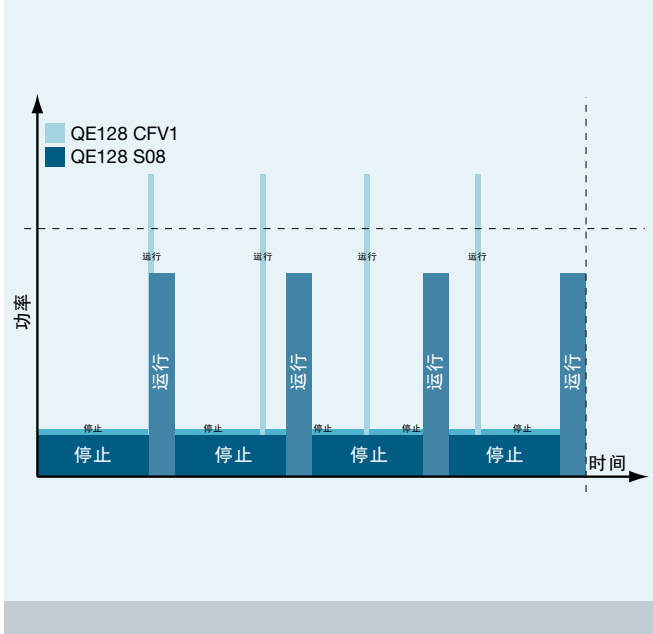


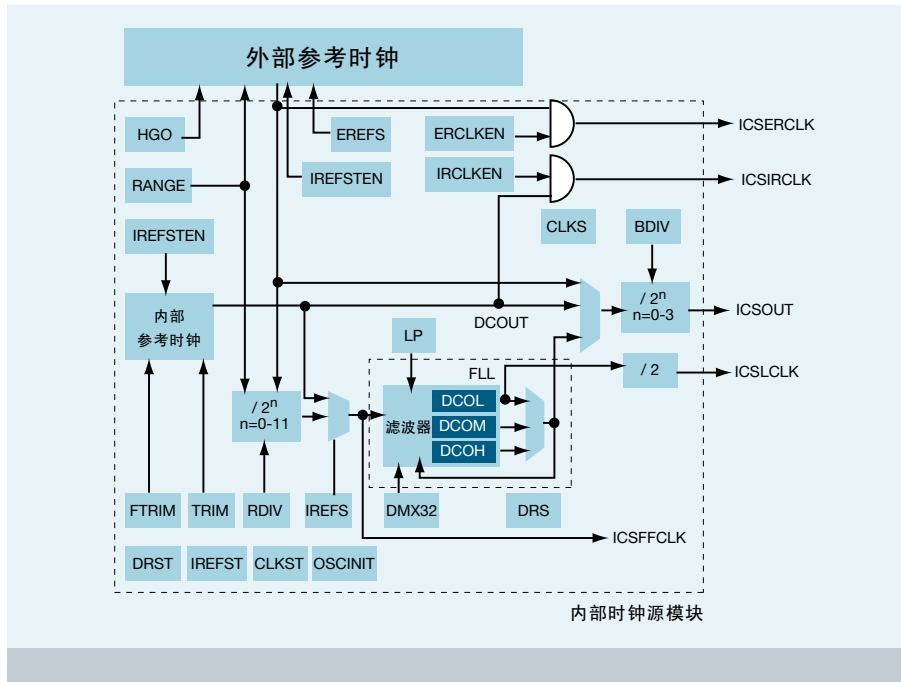
图2: 烟雾探测器应用功率与时间关系示意图



时钟选择

QE128的心脏是图3所示的内部时钟源模块(ICS)。该模块使设计人员可以选择从32 kHz至16 MHz的外部参考时钟(ERCLK)或可调整的从31.25 kHz至39.06 kHz的内部参考时钟(IRCLK)。ICS的核心是锁频环(FLL)模块，它可以将其输入时钟频率倍频至最高50 MHz。FLL的输入时钟，也称为FLL参考时钟，必须在31.25 kHz至39.06 kHz范围内，这样FLL才能正确运行。使用内部FLL参考时钟时非常简单，您要做的只是将频率调整到适当的范围。外部参考时钟的频率可高达16 MHz，并且可以由参考分频器(RDIV)分频，它可以设置为1到1024。如果需要低频总线，那么FLL可以旁路。第二个分频器模块，即总线分频器(BDIV)可以对时钟信号进行1、2、4或8分频，再将其输出到ICSOUT信号。MCU总线时钟为ICSOUT除以2。

图3: 内部时钟源核心



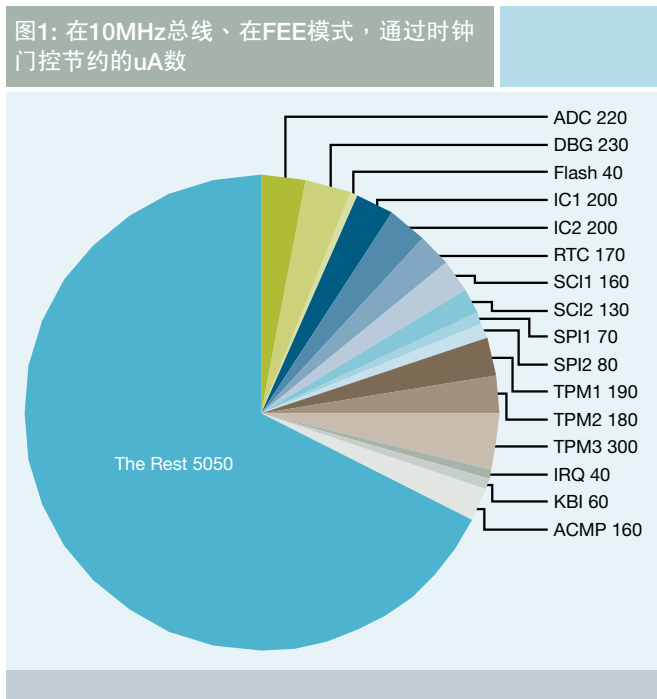
ICS还控制一个独立的1kHz超低功率振荡器(LPO)，它可以供实时时钟(RTC)和看门狗(COP)使用，其使用符合EN60730的标准。EN60730是针对家庭或类似应用的自动化电气控制标准。

表1显示了MC9S08QE128上的每个模块可以使用的时钟源。表中出现的其它时钟有OSCOOUT(到外部时钟的直接通路)和XCLK(进入FLL模块的信号，是内部振荡器或经过RDIV后的外部时钟源)。

	ICSOUT	IRCLK	ERCLK	LPO	其他
ACMP	/2				
ADC	/2		•		
COP	/2			•	
CPU	/1				
Flash	/2				
I ² C1	/2				
I ² C2	/2				
RTC		•		•	OSCOOUT
SCI1	/2				
SCI2	/2				
SPI1	/2				
SPI2	/2				
TPM1	/2				XCLK或外部时钟源
TPM2	/2				XCLK或外部时钟源
TPM3	/2				XCLK或外部时钟源

这款MCU的灵活性的关键是不同模块可以同时使用内部振荡器和外部振荡器的时钟源。这意味着一些模块可以比其它模块运行慢一些，从而可降低功耗。

MC9S08QE128的设计中采用了与时钟有关的另一种节能特性。系统时钟门控寄存器1和2(SCGC1和SCGC2)使用门电路打开(1)或关闭(0)时钟源，从而控制各单元的时钟门控电路，如定时器、ADC、IIC、SCI、



调试模块、闪存、IRQ、键盘中断模块(KIM)、模拟比较器、RTC和SPI(串行外围设备接口)。通过门电路关闭未使用模块的时钟，可以在MCU的“运行”和“等待”模式(后文介绍)下节约宝贵的微安数(uA)。这种特性对于MC9S08QE128等器件尤其重要，因为它集成了大量的通信模块和定时器。

仅在需要时打开模块的门电路，在作业完成时关闭门电路，这是一种重要的节能方法。在写入寄存器后时钟门将立即开启或关闭。然而，这种方法也有一些弱点需要考虑。

- 复位后，所有的时钟门都是开启的，因此为了保持低功耗，时钟门应尽快关闭。
- 对时钟门已关闭模块的寄存器的写操作无效。
- 为了避免错误操作，比较好的做法是在时钟门关闭前禁用该模块，并在时钟门开启后重新初始化该模块的寄存器。

图1说明了通过关闭各模块的时钟门可以实现的节能效果。这些数字仅表现了某个测试环境中的情况，飞思卡尔半导体不对这些数字的准确性负责。

操作模式—时钟

如上所述，ICS连到了所有消耗功率和器件控制的外围设备上。这使设计人员能够根据需要进行功率降低的程度和性能水平。同样重要的其它因素包括：

- 使用低频率、低增益的外部振荡器比使用内部电路更加节能
- 通过旁路FLL或PLL(锁相环)，由于降低了频率从而可降低功耗，同时旁路FLL本身也可以降低功耗。
- 在MC9S08QE128 MCU中禁用而不是旁路FLL/PLL可进一步降低功耗。

注意，在PLL和FLL之间进行选择时，FLL是功率最低的选项，尽管它的长期准确性较低。

MC9S08QE128 MCU中的ICS有6种运行模式：FEI(FLL启用的内部时钟模式)，这是缺省模式；FEE(FLL启用的外部时钟模式)；FBI(FLL旁路的内部时钟模式)；FBE(FLL旁路的外部时钟模式)；FBILP(的功耗FBI模式)和FBELP(低功耗FBE模式)，它们通过ICS控制寄存器2中的LP位来控制。FBILP和FBELP可以降低振荡器轨道上的电压摆幅，从而降低整体功耗。

运行模式—CPU

在系统设计的适当阶段选择正确的运行模式可以实现甚至突破产品功耗目标。最新一代的MCU产品能提供多种模式选择，每种方式又有不同的优点和缺点，以往的简单选择“运行”模式还是“低功率”模式的日子已经过去。设计人员必须选择退出路径、唤醒次数和寄存器保持等等。除了常用的“运行”模式，MC9S08QE128还有另外5种节能模式。其中的“等待”、“停止3”和“停止2”等三种模式对于S08内核而言并不陌生，在其它内核中使用也很常见，只是名称有可能不同。

在“等待”模式下，CPU完全关闭以降低功耗。MCU系统时钟和整个电压调节功能将保持。根据总线频率的不同，等待模式可将运行模式电流降低30%~60%，而且任何中断都允许立即退出等待模式。

“停止”模式暂停系统时钟并使稳压器进入待机状态。

在MC9S08QE128 MCU中，“停止3”模式在两种停止模式中消耗的能量最高，但好处是能够使唤醒时间缩短到仅仅6微秒(us)。“停止3”模式有三种版本：

- 启用BDM
- 稳压器工作
- 精简型

与“等待”模式不同，退出“停止3”模式仅限于从RTC、低电压检测/低电压报警(LVD/LVW)、ADC、带内部参考电压的模拟比较器(ACMP)、IRQ、SCI、KBI和复位。“停止2”是MC9S08QE128 MCU中最低功率模式的名称。这种方式下，内部电路断开电源，但仍维持RAM的内容和引脚状态。主要的不同之处在于，只有三种方式可以退出这种模式，即复位、IRQ和RTC，且MCU在退出时产生复位，效果与上电复位相同。

MC9S08QE128器件中有两种新的节能模式：“低功率运行”(LPR)和“低功率等待”(LPW)。LPR与常规“运行”方式相比降低功耗的方法是将稳压器置于待机状态。要进入这种模式，必须运行FBELP模式，即功率最低的ICS模式。必须禁用低电压保护系统、MCU内部带隙电压以及内置在线仿真器/后台调试控制器(DBG/BDC)模块，因为无论总线频率是多少，这三个模块都将消耗大量功率。由于稳压器处于待机状态，我们建议在此方式下使用所有可能降低功耗的方式。

S08产品采用SST许可使用的SuperFlash技术。该信息可能会有所变化，决策时应使用最新参考手册。

“低功率等待”(LPW)模式只能从LPR模式进入，因此对LVD电路、调试和ICS模式的限制是一样的。与LPR相比，MCU在LPW模式下通常可以将电流消耗量降低50%。

表2对比了不同CPU运行模式的主要特性。

表2: CPU运行模式对比表

	典型总线频率	典型 I _{dd}	退出源	退出时间
运行	8 MHz	5 mA	无	无
低功率运行	16 kHz	25 μA	LP位清零	无
等待	8 MHz	2 mA	任何中断	立即
低功率等待	16 kHz	4 μA	任何中断	立即
停止 3	无	450 nA	RTC, LVD/ LVW, ADC, ACMP, IRQ, SCI, KBI	6 μs
停止 2	无	370 μA	RTC, IRQ 或RESET	6 μs + 73总线 时钟

有助于节约功率的其他MCU特性

MC9S08QE128 MCU具有可以在低至1.8V电压下进行重新编程的闪存，这可以在便携式应用中延长电池使用寿命。MC9S08QE128器件上的三个端口都具有“置1”、“清零”和“翻转”功能，这可以加速引脚操作，缩短代码执行时间，从而可以迅速重新进入低功率模式。器件的所有输出引脚都具有斜率控制功能，当驱动外部器件时，可以通过减慢瞬变过程来降低功率并控制驱动强度。

结论

通过仔细考虑模块基础上的时钟源、时钟分布和时钟模式，以及在恰当的时间和恰当的时间长度使用恰当的CPU模式，通过我们的MCU可以实现针对现代消费和工业产品的严格功耗限制。不是只有时钟和CPU模式是可以降低功率或延长电池寿命的MCU功能，在应用指南AN3460中进一步讨论了其他一些简单方法，可以从freescale.com上下载。

总之，优化的MCU设计可使系统功耗降低一半。飞思卡尔所采用的功能都是概念简单且易于使用的。功能的多样性意味着所有类型的应用，无论是电池还是电源供电的，都可以从中获益。这些功能很容易理解、合成和使用，为您提供了一个实实在在的工具，可以有效地处理您的系统中最严格的功率问题。

Inga Harris于2000年毕业于Strathclyde大学，拥有电子学和电子工程专业的荣誉学位。她所从事的飞思卡尔的技术市场和工程应用工作主要面向消费和工业市场，以8位MCU系列为主。

用于实时通信的 超低功率无线设计

我们的设计团队被要求开发一个用于演示三轴加速计(飞思卡尔MMA7260QT)的无线系统。这样做的目的是为了用无线连接代替最初RD3112MMA7260Q (STAR)演示板中使用的串行电缆,使用最新的飞思卡尔技术来将加速计数值从传感器中实时地传输到PC中。

无线演示板的主要特性包括:

- 非常小的尺寸(硬件和软件)
- 低功率
- 电池供电
- 20米的无线覆盖范围
- 印刷线路天线
- 实时响应
- 低成本
- 可用作参考设计

初级评估之后,我们清楚地发现主要的设计挑战是低功率和低电流消耗。该演示的目的就是为加速计测量值提供每秒约30次的传输频率。因此在使用有限的电源时每个微安秒(μAs)都不容忽视。

只知道传感器电路板所用的电池而不知道将采用哪种无线技术会也是一个难题。起初考虑的是AA和AAA碱性电池。容许的电流是足够的,然而电池的实际尺寸与项目的其他部分相比就显得过大。由于该演示的目标之一就是演示飞思卡尔的极低功率技术,因此最后决定使用“纽扣”型的锂电池。常见的220 mAh CR2032 3V纽扣电池带有合适的电池固定架,从外形来看是非常理想,而且可提供充足的电池寿命来进行成功的演示。为了强调采用低功率电池作为电源,传感器电路板考虑使用透明罩甚至不使用盖罩。

找寻合适的无线技术

下一个设计选择是确定最理想的低功率无线技术,将数据从加速计传输到计算机。最后选择了2.4 GHz工业、科学和医用(ISM)频段,因为它是全球都接受的,而且能够充分利用小型电路板和低成本(印刷线路)天线。

3种知名的标准无线通信技术可在2.4 GHz ISM频段内运行,即Wi-Fi、蓝牙和IEEE 802.15.4技术。Wi-Fi可能最有名的,可提供最快的通信速率,而且非常适合计算机连接的部分。然而它不是为超低功率运行和长期电池寿命而设计的。蓝牙技术的确更适合低功率、电池供电的设备,它设计用于为电话、耳机和其他手持消费设备提供近距离(10m或更少)的中速无线连接。另一方面,IEEE 802.15.4技术最初就是为小型电池供电的设备提供数月甚至数年的低速率近距离(20m至100m)的无线连接而设计的。它的初衷就是低数据速率(如传感器和控制网络中),而且超低功耗的表现卓越。由于这些原因,基于802.15.4的无线传输技术被选择用于无线加速计演示设计,我们团队将其命名为ZSTAR。

演示方案概述

该演示使用ZSTAR传感器模块和(一般通过USB接口)连接到PC的802.15.4无线数据收集器模块。飞思卡尔有许多演示模块可用作数据收集器。ZSTAR演示方案采用的模块是一个无线传感器电路板,它使用了飞思卡尔的3种主要器件:

- 三轴加速计MMA7260QT,也称为“Triax”
- 16脚8位微控制器(MCU)MC9S08QG8
- MC13191 802.15.4收发器(无线调制解调器)

所有这些器件都能够支持超低功率运行。为了确保合理的电池寿命,必须充分利用所有这些器件提供的强大的低功率运行模式。

表1: 调制解调器、MCU和三轴加速度计各状态下的电流消耗

调制解调器运行模式	描述	典型电流消耗
关闭	整个器件断电	0.2 μ A
休眠	保持RAM/寄存器内容，数字IO保持其状态。参考时钟振荡器关闭。	1 μ A
半休眠	晶振参考时钟开启。保持RAM/寄存器内容，数字IO保持其状态。输出时钟可用。定时器可用于唤醒功能。	35 μ A
空闲	参考时钟振荡器运行。可快速过渡到接收和发送模式。	500 μ A
接收	接收模式或CCA	37 mA
发射	发射模式	30 mA
MCU运行模式	描述	典型电流消耗
停止1	所有I/O脚自动转换到默认的复位状态。MCU的大部分内部电路断电。RAM内容丢失。	475 nA
停止2	保持RAM。所有I/O脚控制信号维持原状态。唤醒等同于复位启动。	600 nA
停止3	保持所有内部寄存器和RAM内容，保持I/O脚状态。唤醒等同于中断服务。	700 nA
等待	CPU核心时钟停止，所有启动的外围设备正常运行。	1 mA
运行	所有外围设备关闭，CPU内核时钟为32.768 kHz，源自32.768 kHz晶振。	95 μ A
运行	所有外围设备关闭，CPU内核时钟为约32KHz，源自内部振荡器(无锁频环)。	150 μ A
运行	所有外围设备运行，CPU内核时钟开启(8MHz)。	3.5 mA
三轴加速度计运行模式	描述	典型电流消耗
睡眠	无输出	3 μ A
运行	三轴加速计运行	500 μ A

低功率考量

影响功耗的几个重要因素：

- MCU、Triax和调制解调器的省电模式须在整個通信周期的每个部分进行处理(指将一批加速计数值发送给PC的时间段)
- 每秒发送的数据包的数量(即通信周期的速率)
- 每秒交换的数据包的数量(802.15.4协议对每个数据包进行确认，但是为了降低功耗，传感器每进行8次传输才发送一个确认)
- 数据包长度(协议必须简单而轻便，具有最少的开销和最小数据长度)

表1总结了可用的器件状态，综合介绍了这些器件可提供的运行模式。

为实现低功率而平衡系统

传感器电路板的运行必须从功耗角度考虑。由于所发送数据包的数量和大小是一致的，没有变化，因此功率的处理必须着眼于工作周期的各个部分。

A. 低功率或通讯之间的安静时间

由于电路板具有MCU时钟模式灵活性，低功率考量可在精确定时和非精确定时之间的作充分的选择。

1. **精确的接收/发送定时(基于晶振)** — 发送/接收操作都可精确定时。此外，接收窗口可以尽可能的窄，从而节约电流。

- a. MCU带32.768 kHz晶振、调制解调器为关闭模式。在等待下次通信时，MCU低速运行。在需要高速运行的时间段，MCU切换至锁频锁环(典型总线时钟

为8 MHz)。该解决方案的缺点是从关闭到空闲模式的过渡需要10至25ms的时间，用于稳压器和时钟振荡器的启动。此外，调制解调器RAM的内容与GPIO状态也会丢失，因此每次必须重复进行所有的配置。它需要两个晶振(调制解调器为16 MHz，MCU为32.768 kHz)。

总体功耗：200 nA (调制解调器) + 95 μ A (MCU)

- b. MCU带32.768 kHz晶振、调制解调器处于休眠模式，表现于1a) 的情形类似。从休眠到空闲模式的过渡时间稍短(8至20ms)。该模式的优点是RAM的内容与GPIO状态可以保持。该模式与前述相同使用两个晶振。

总体功耗：1 μ A (调制解调器) + 95 μ A (MCU)

- c. MCU由内部振荡器提供时钟，或由外部提供(通过调制解调器提供的CLKO输出)，调制解调器处于半休眠模式。若MCU提供内部振荡器，当命令MCU退出半休眠模式时，它可以快速响应。MCU复位并配置调制解调器。调制解调器在休眠模式下也可以提供参考时钟。MCU将CLKO编程至所需频率并等待CLKO稳定。CLKO稳定后，MCU将时钟切换到外部时钟源。调制解调器中长时间延时的精确定时是由周期性地从半休眠模式中退出的时间来完成的，而此时MCU处于睡眠状态。在这种结构中，设计人员可充分利用自动启动的定时器触发事件，如：

- 激活接收模式
- 激活发射模式
- 激活空闲模式

此外，接收帧在接收期间会进行“时间标记”，从而可以在各种不同事件间精确定时。微控制器可一直处于低功率模式。与MCU使用独立晶振的方法相比，这种情况下的功耗稍高，但它只需要一个晶振(调制解调器晶振)。

2. 非精确数据包定时

- a. MCU定时由内部振荡器(RTI)控制，调制解调器处于关闭模式。这种情形的缺点是通信定时不是那么精确。MCU定时由不能提供精确定时的RTI内部振荡器控制(仅30%的精度)。因此接收窗口必须更宽，从而对功耗产生负面影响。只需一个晶振。

总体功耗：200 nA (调制解调器) + ~1 μ A (MCU睡眠模式 + RTI开启)

- b. MCU定时由内部振荡器(RTI)控制，调制解调器处于休眠模式。这种情形与a相似。不同的是到空闲模式的过渡时间更短，并且不需要重新初始化调制解调器。

总体功耗：1 μ A (调制解调器) + 1 μ A (MCU睡眠模式 + RTI开启)

- c. MCU由内部振荡器提供时钟，以最低频率运行(4kHz)，调制解调器处于关闭模式。

MCU总线时钟由内部参考时钟驱动。具有2%的精度(调整之后)，足以满足许多应用的需求。调制解调器为关闭模式，只需要一个晶振。

总体功耗：200 nA (调制解调器) + 150 μ A (MCU以4 kHz内部时钟运行)

- d. MCU由内部振荡器提供时钟，以最低频率运行(4kHz)，调制解调器处于休眠模式。该情形与c相似；调制解调器处于睡眠模式。

总体功耗：1 μ A (调制解调器) + 150 μ A (MCU以4 kHz内部时钟运行)

表2

模式	MCU时钟	精确性	调制解调器模式	消耗
1a	Xtal 32768 Hz	100 ppm	关闭	95 μ A
1b	Xtal 32768 Hz	100 ppm	休眠	96 μ A
1c	调制解调器的时钟	20 ppm	半休眠	36 μ A
2a	RTI 时钟	30%	关闭	1.2 μ A
2b	RTI 时钟	30%	休眠	2 μ A
2c	内部时钟 4 kHz	2%	关闭	150 μ A
2d	内部时钟 4 kHz	2%	休眠	151 μ A

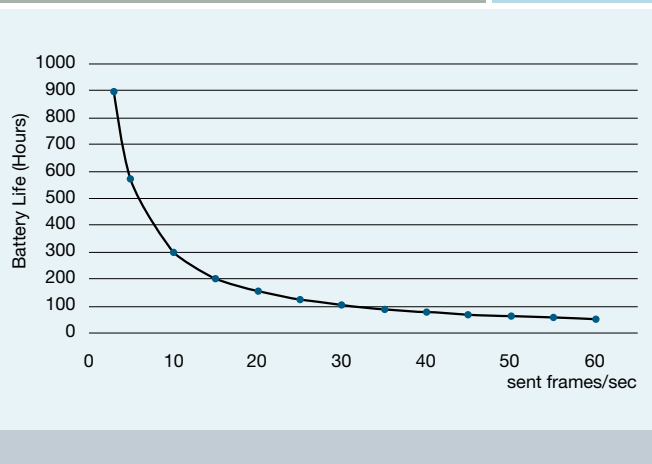
现在让我们看看总体功耗的另一部分。

B. 发送和接收事件

正如调制解调器发送/接收操作模式的电源电流所示，调制解调器的发送和接收状态对功耗也有很大的影响。因此，数据包的长度需要进行高度优化，而且必须仔细选择通信速率。

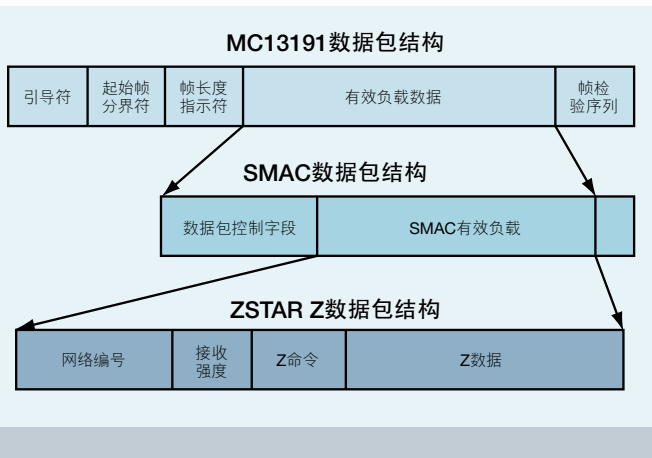
在下面的几张图中，您可以看到通信速率和数据有效负载长度对电池寿命的影响。图1显示了电池寿命小时数与每秒发送的帧数的关系。我们必须将速率折衷为每秒30个包，以得到足够的采样速率和合理的电池寿命。

图1: 电池寿命与数据包/秒的关系



数据包的长度对电池寿命也有很大影响。这就是我们为什么只发送三轴加速计的原始数据的原因(请参考《ZSTARRM参考手册》第5.3章中的数据包的格式及长度，以了解更多详情。)

图2



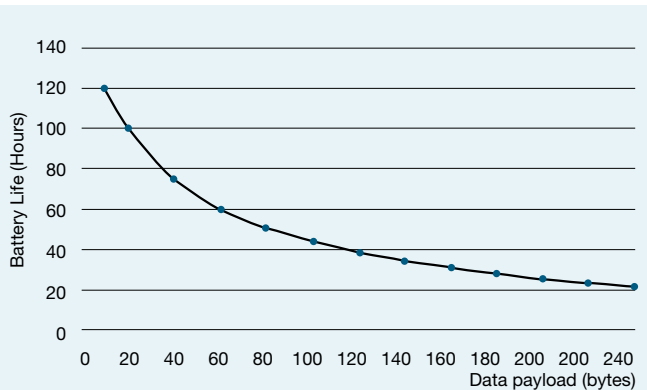
符合IEEE 802.15.4的标准数据包有4个字节的引导符、1个字节的STD(起始帧分界符)、1个字节的FLI(帧长度指示)、2字节的FCS(帧检验序列)以及有效负载数据(最多125字节)。因此，帧开销为8字节或 $8 \times 32 = 256 \mu s$ ，最长的有效负载发送时间为 $125 \times 32 = 4000 \mu s$ 。那么数据包发送时间为：

$$\text{总发送时间} (\mu s) = 256 + (\text{有效字节} \times 32)$$

我们传感器电路板的标准SMAC数据包有17字节(包括2字节的IEEE 802.15.4控制字段包)。还有两个分别需要 $144 \mu s$ 和 $10 \mu s$ 的所谓“预热”(从空闲到发送过渡)和“冷却”(发射到空闲过渡)时期。

下图显示了数据包长度和电池寿命的关系。很明显，数据包长度应尽可能小。

图3: 电池寿命与数据有效负载的关系



这个图也只是理论上的，因为实际上还有另一个物理限制。电源(锂电池)只能提供非常有限的持续电流，因此如果传输较长的帧，电池电压将明显下降。在ZSTAR的设计中，使用了一个大容量钽电容器($470 \mu F$)来改善电源对接收或发送引起的电流峰值的响应。即使如此，较长的数据帧(大于约50-100字节)还是会导致电压大幅下降。

电容计算：我们规定发射期间(1 ms)容许的电压下降为100 mV。

$$\Delta u = \frac{1}{C} \cdot i \cdot \Delta t$$

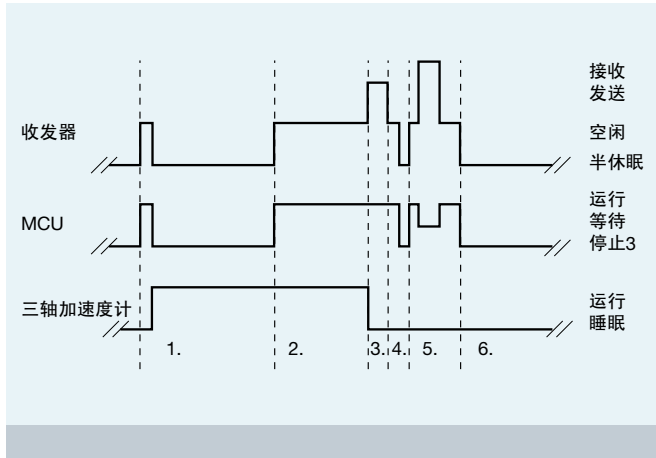
$$C_{min} = i \frac{\Delta t}{\Delta u} = 30 \text{ mA} \cdot \frac{1 \text{ ms}}{100 \text{ mV}} = 300 \mu F$$

最后，我们从相同封装系列中选择了 $470 \mu F$ 电容器。

如何协同运行

传感器电路板卡设计得非常小巧而且具有高性价比，因此我们在调制解调器中只使用一个晶振。同时，我们需要精确的通信定时，所以用调制解调器休眠模式进行通信定时，而MCU使用内部时钟(在工作期间)，否则MCU就处于停止3模式。传感器电路板与USB接收头之间的通讯如下所述。

图4



一个数据包的发送过程

1. 该阶段开始时，调制解调器从停止3模式中唤醒MCU。MCU将Triax从睡眠模式下唤醒(控制Triax的SLEEP脚)。然后Triax的输出必须稳定，然后再进行测量。在这段时间内，调制解调器再次进入定时的半休眠模式，同时MCU也进入停止3模式。
2. 当Triax准备好进行测量时，调制解调器再次唤醒MCU，然后MCU测量Triax。MCU获得数据后Triax切换回睡眠模式，数据包形成并放入调制解调器的RAM缓冲器中。
3. 调制解调器通过射频发送数据包。
4. 为了检测连接错误(传感器电路板和USB接收头之间)，必须等待USB接收头的响应。然而，因为USB接收头需要一定的时间来处理接收到的数据并形成

响应数据包，所以MCU再次将调制解调器设置为短时间的半休眠模式。

5. 这段短时间过后，MCU醒来并在规定的接收窗口内接收USB接收头的响应。然后，调制解调器被设置为半休眠模式，直到一个周期结束，同时MCU进入停止3模式。
6. 在两个连续数据包之间，MCU一直处于停止3模式，Triax切换到睡眠模式，调制解调器处于休眠模式。同时，其时基会测量到收到下一数据包的时间。

在评估该通讯方式的过程中我们发现，每次发送后的接收会大幅增加电流消耗。由于这不是要求实现的或必需的，因此USB接收头的响应接收窗口只是每8次才打开一次。在其他情况下，发送结束之后传感器电路板立即进入休眠模式(跳过状态4和5)。

总结

我们对ZSTAR项目的最初目的(使用有限的电源传输实时数据)进行了评估。系统设计的各个方面(硬件架构和软件实施两方面)被经过研究和优化以实现最佳的应用性能。三轴加速计的XYZ采样值，加上内部温度与电池电压水平值，使用微型锂CR2032电池每秒发送30次，可达到100小时的运行时间。系统的平均电流消耗估计为2.5 mA，实际测量也证实了这种预期。

总之，ZSTAR的设计表现了一个采用MC1319x调制解调器系列和高性价比的少管脚8位微控制器(MC9S08QG系列)实现的高性价比方案，用于3轴加速度计(MMA7260QT)的无线感测。

Radomir Kozub是飞思卡尔的应用工程师，在公司任职已近4年。他拥有电气工程专业硕士学位。加盟飞思卡尔之前，Kozub致力于开发各种测量应用。

Pavel Lajsner是飞思卡尔的应用工程师，在公司任职已超过11年。他拥有电气工程专业硕士学位。

JTAG TAP的高效控制

JTAG TAP转换变得更轻松

介绍

飞思卡尔的许多微处理器都带有JTAG端口——一种兼容IEEE 1149.1标准、用于执行边界扫描和调试嵌入式应用程序的端口。早在20世纪80年代中期，Joint Test Action Group (JTAG)就开始着手IEEE 1149.1标准的制定，后来IEEE接手了这项任务。因此，符合IEEE 1149.1标准的测试接入端口就按最初发起标准化的这个机构命名，通常被称作JTAG端口。

现在，JTAG端口主要用于边界扫描、编程和调试目的。所有符合IEEE 1149.1标准的JTAG具体实现并不只限于提供该标准规定的功能，其功能可以自由扩展。例如IEEE-ISTO 5001论坛™标准在定义嵌入式处理器调试接口时就以JTAG为基础。

IEEE 1149.1标准把通用测试接入端口(TAP)定义为访问JTAG功能的物理层。此外，该标准还定义了TAP控制器。该控制器负责控制用于访问JTAG功能及其他可能包含的电路(如调试模块)的信号。TAP控制器是一种同步有限状态机(FSM)，可以响应TAP引脚的变化。为了访问特定的JTAG功能，您必须使TAP控制器在各状态间变化。如需了解有关各个TAP控制器各状态的作用及其他JTAG技术规范的信息，请参见IEEE 1149.1标准。

本文介绍了一种简单而高效地沿著最短路径将TAP控制器从当前TAP状态跳转到其他状态的算法。编程人员一般是通过编写一系列操作TAP输入引脚的指令来跳转TAP控制器的状态。因此，为所有软件模块编写跳转代码，同时还要协调JTAG和其他相关功能，极易变成一件棘手、乏味而且容易出错的任务。我们的算法采用一个例子程序将TAP控制器引导到希望的状态，从而免除了一系列的TAP控制指令。因此，您只需要调用该程序，如GoTo(dstate)程序，TAP控制器就能跳变到希望的状态dstate。

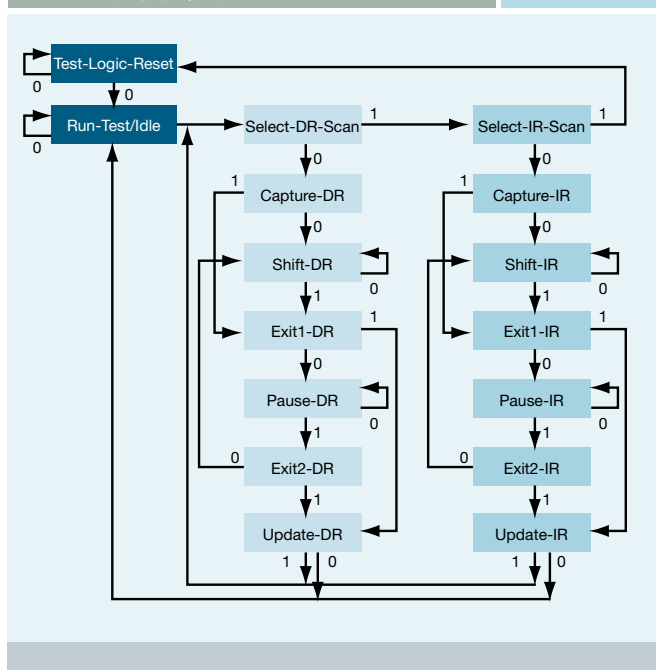
跳变算法

在TAP控制器状态机中(如图1所示)，如果由TAP的TMS引脚上的逻辑信号电平(0或1)进行使能控制，并且TAP处于转换的初始状态，那么在时钟(引脚TCK)的上升沿之后就会产生一次转换。图1中的箭头0和1代表使能控制信号的逻辑电平。

图1所示的框图可以看作是由顶点和有方向的边沿组成的图表，其中顶点为TAP控制器的状态，有方向的边沿为转换。寻找任何两个顶点之间的最短路径是一个经典的已解问题。在TAP状态机中，最短路径很明显，可以手动找到。因此，我们需要一个高效的机制来保存最短路径，这样以后就可以照做而不需再次寻找了。

为保存最短路径，我们可以利用以下推论。假设状态 s_1 到状态 s_n 之间有一条最短路径，并且该路径经过状态 s_k 。这样该路径为 $s_1, \dots, s_k, \dots, s_n$ 。那么，路径 s_k, \dots, s_n 就是状态 s_k 和 s_n 之间的最短路径。如果该路径不是最短路径，那么

图1: TAP控制器状态图



我们就可以缩短 s_i 和 s_n 之间的路径，这样最初那条路径就不是最短路径，这就产生了矛盾。

因此，对于任意从 s_i 到 s_n 之间的最短路径，只保存路径上紧跟著 s_i 状态之后的下一个状态就足够了。存储各最短路径的下一个状态的信息库被称作路由表，类似于网络中对路由信息的定义。在网络中，每个节点都保存下一个相邻节点的向量，与此不同，我们的路由表是一个矩阵，因为状态转移是由TAP接口控制的。TAP路由表如表1所示。

通过最短路径将TAP控制器转移到希望状态的算法非常简单。假设当前的TAP状态保存在变量`tap_state`中，而二维矩阵Routing-Table保存表1的内容。GoTo(`dstate`)函数的运行如下所示：

```
GoTo(dstate)
{
    while(tapstate != dstate)
        step(RoutingTable[tapstate][dstate]);
    return;
}
step(tms)
{
    将TMS引脚设置为tms (通常在并行端口上);
    在TCK引脚上生成时钟脉冲;
    tap_state = next_state(tap_state,tms); //
更新当前状态
}
next_state(tap_state,tms) 函数使用另一个表格，它给出每个给定的TMS引脚值的状态的下一个状态。
```

实现

我们使用C++实现上面介绍的算法。一个称为JTAG的Singleton类控制TAP控制器的信号，保存当前状态并将TAP控制器转移到希望的状态。当这些类启动时，路由表及其他表格被载入内存，TAP控制器复位到test-logic- reset初始状态。JTAG Singleton接口提供一种叫作AssertTAPState(`tapstate`)的功能来确保当前的TAP控

表1: TAP路由表

状态	Test-logic-reset	Run-test/idle	Select-DR-scan	Capture-DR	Shift-DR	Exit1-DR	Pause-DR	Exit2-DR	Update-DR	Select-IR-scan	Capture-IR	Shift-IR	Exit1-IR	Pause-IR	Exit2-IR	Update-IR
Test-logic-reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Run-test/idle	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Select-DR-scan	1	1	x	0	0	0	0	0	0	1	1	1	1	1	1	1
Capture-DR	1	1	1	x	0	1	1	1	1	1	1	1	1	1	1	1
Shift-DR	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
Exit1-DR	1	1	1	1	0	x	0	0	1	1	1	1	1	1	1	1
Pause-DR	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Exit2-DR	1	1	1	1	0	0	0	x	1	1	1	1	1	1	1	1
Update-DR	1	0	1	1	1	1	1	1	x	1	1	1	1	1	1	1
Select-IR-scan	1	1	1	1	1	1	1	1	1	x	0	0	0	0	0	0
Capture-IR	1	1	1	1	1	1	1	1	1	1	x	0	1	1	1	1
Shift-IR	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
Exit1-IR	1	1	1	1	1	1	1	1	1	1	1	0	x	0	0	1
Pause-IR	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
Exit2-IR	1	1	1	1	1	1	1	1	1	1	1	0	0	0	x	1
Update-IR	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	x

制器状态为`tapstate`。如果当前状态不一样，该功能会将TAP控制器转换到正确的状态。

JTAG Singleton通过操纵连接到嵌入式处理器上的JTAG物理接口的并口引脚来转变TAP控制器状态。通过并口控制JTAG引脚的方法由另一个称为JTAGPort的Singleton实现，它只能由JTAG访问。

结论

我们介绍了可以用一个很小的路由表来保存任意两个TAP控制器状态间的最短路径，以此方法可以省去编写冗长乏味的一系列转变TAP控制器状态的指令。考虑到当前计算机的速度和能够在并行端口上实现的最大频率(约1MHz)，调用函数或功能以及查看TMS引脚值产生的相关开销可以忽略不计。此外，JTAG端口通讯速度受处理器时钟频率的限制。

David Baca是飞思卡尔的应用程序工程师，在飞思卡尔工作近两年。他拥有电气工程和工商管理专业硕士学位，目前正在攻读人工智能博士学位。在加盟飞思卡尔之前，Baca曾参与过NASA和美国空军等机构赞助的研发项目。

利用ColdFire® MCF5223x嵌入式控制器 加速创新

介绍

在当今这个迅速发展的世界里，节省时间以及让我们生活的各部分都处于控制之中的要求变得日益重要。技术在影响世界的变化方式方面继续发挥著重要的作用。在家中，几乎每一个设备都内置某种技术，而这种技术本身还在不断创新。连接就是一个很好的例子，说明技术总是在不断变化，提供新的解决方案来使我们的生活变得更轻松、更快捷和更高效。

设计挑战

以太网为人类打开了一个充满无限可能的新世界。它使我们能够从全球任何地方接入全球网络，让我们能够随时著手进行日常活动和家务。借助以太网连接，人们在办公室就可以通过计算机远程控制家里的温度或照明。

随著低成本具有以太网功能器件的面世，房屋和楼宇的自动化功能现在也已开发。高端家电，诸如与以太网连接的冰箱、报警系统、摄像头、门窗锁、灌溉系统和空调系统等，只是一部分说明该技术如何使生活更轻松的实例。

飞思卡尔半导体致力于使生活变得更轻松、更安全，这正是我们推出大量新型低端32位ColdFire设备来实现家用电器、工业流程和安全系统的远程控制的原因。飞思卡尔让开发工程师可以灵活地从一系列不同的ColdFire嵌入式控制器产品中选择合适的32位微控制器。

飞思卡尔解决方案

以太网控制可用于许多应用，如家用电器和工业控制。它们都采用以太网作为主要的通信协议，同时使用一个能实现应用程序和连接任务的控制器。飞思卡尔提供的新型低成本ColdFire MCF5223x系列器件非常适合这类控制和连接需求。

MCF5223x概述

ColdFire MCF5223x系列建立在过去曾广泛应用68 KB系列的基础上，是新型低端ColdFire器件中的一种。为了确保安全的连接，MCF5223x系列ColdFire器件是一种单芯片解决方案，提供具有以太网接口的32位控制。它包含有10/100快速以太网控制器(FEC)、以太网物理层(EPHY)、CAN 2.0B和加密加速单元(CAU)，它采用Version 2 ColdFire内核进行安全硬件加密，从而以合理的成本提供出众的性能(60MHz时可达57个Dhrystone 2.1 MIPS)。MCF5223x嵌入式控制器可为设计人员提供一系列合适的外围设备和内存空间(高达32 KB的SRAM和高达256 KB的闪存：10万次擦/写周期，10年数据保持)，适合作为紧凑型的具有以太网功能的平台，可减少开发时间和成本，帮助您的产品更快上市。

图1: MCF5223X结构框图

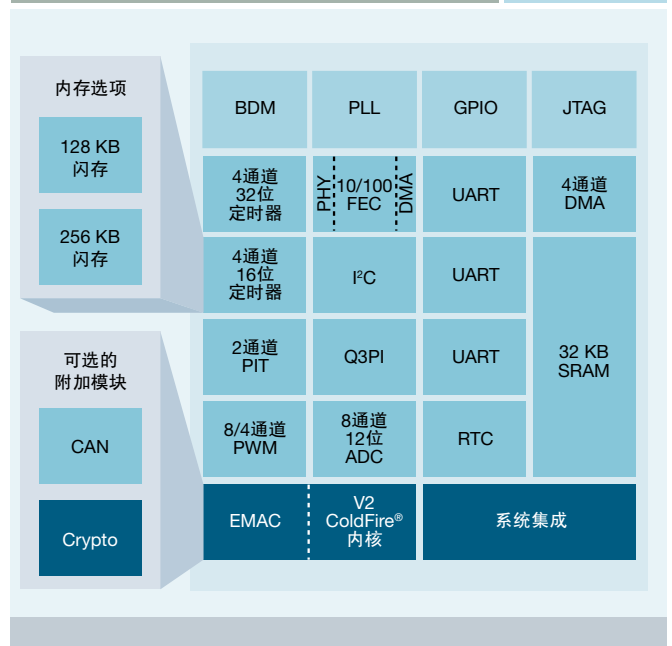
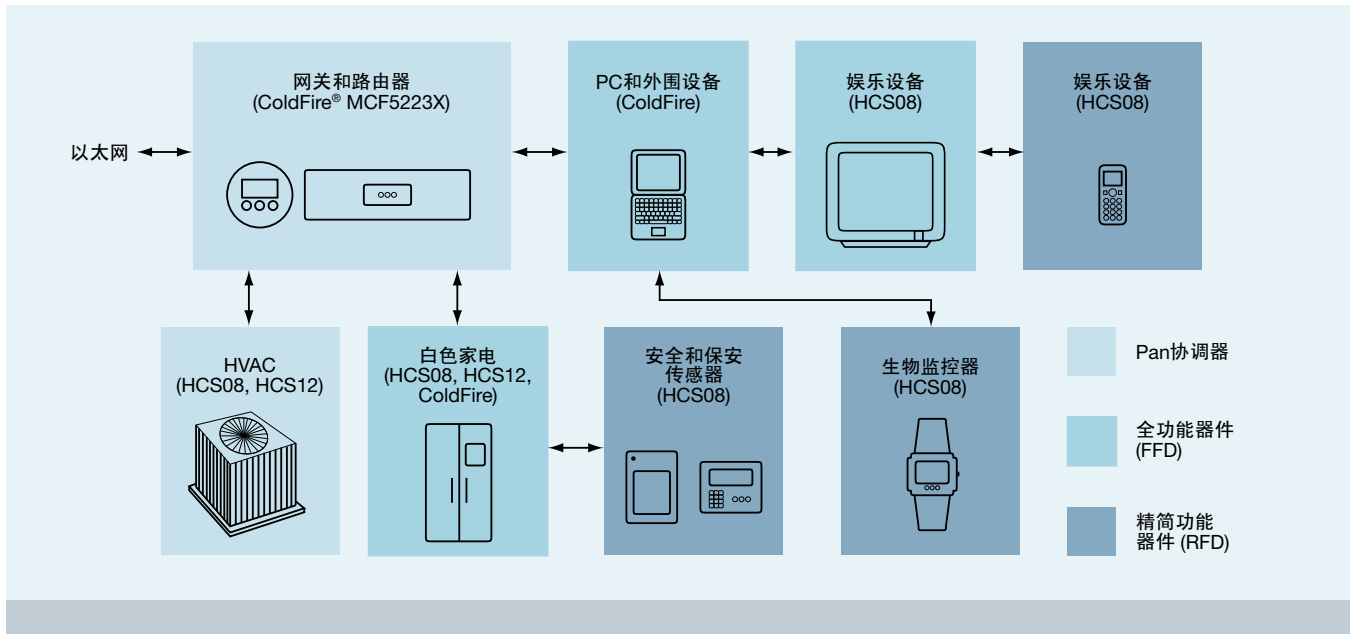


图2: 典型的ZigBee®网络



MCF5223x系列集成了标准的ColdFire外围设备，包括三个用于中远程连接的通用异步接收器/发送器(UART)、一个集成电路互连(I²C)和一个队列式串行外设接口(QSPI)，用于所连外围设备的系统内通信。

在数字控制方面，MCF5223x系列提供以下外围设备：4个带有DMA功能的32位定时器通道、4通道/16位或8通道/8位PWM发生器、2个用于报警和倒计时的周期中断定时器(PIT)、4通道DMA控制器、8通道12位ADC，多达73个通用I/O以及系统集成模块(如PLL和COP)。

这些器件为现有的8位和16位用户拓宽了移植桥梁，如果这些用户需要具有32位性能、快速以太网功能和高级工具支持的低成本入门级32位解决方案。飞思卡尔通向高端32位运行和连接功能的ColdFire路线图可以帮助客户轻松而快捷地移植其应用程序。

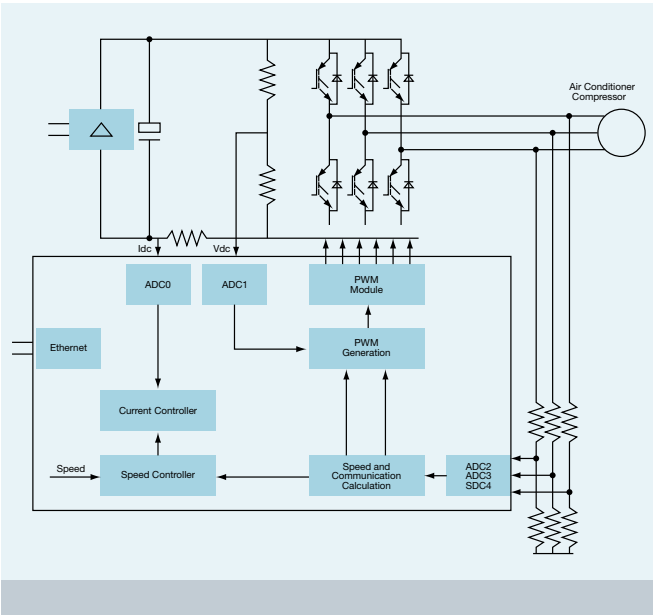
ColdFire MCF5223x作为ZigBee®系统的主机

想像一下，如果您能从PC上控制家中的照明、传感器或门锁该有多好。其实这类网络控制没有听起来那么复杂—只需要无线通信和连接到以太网的网关就可以实现。飞思卡尔同时提供无线解决方案和控制器解决方案。M13192器件是在2.4 GHz频段上运行的收发器，

误包率(PER)为1%时的接收灵敏度为-92 dBm，电源为2.0-3.6V(采用片上稳压器)，逻辑接口1.7-3.3，使用一块锂电或两块碱性电池，带有完整的RF收发器数据调制解调器—天线输入，全数据包输出，通过最低频率为4MHz的标准SPI实现数据和控制接口。MC13192还带有802.15.4的MAC。

收发器可通过SPI轻松连接到MCU上。收发器可连接到锁扣传感器上、灯光断路器上或任何希望被监控或控制的设备上。这些设备应当带有标签，而且作为设备与连接到MCF5223x控制器的中心收发器进行通讯。MCF5223x设备可以连到Web，可以对每个设备进行监控，甚至在需要时还可以采取行动(使用InterNiche的免费协议栈)。图2显示了一个典型的家庭ZigBee网络。

图3: 具有以太网功能的空调控制器



ColdFire MCF5223x用作网关进行监控和诊断

监控某个工业线路并将结果自动显示到内部网络上也是一项常见任务。实时监控可以提高时效，系统化地将数据库中的信息写入系统中而无需人工干预。这样就可以在生产线上进行远程过程控制，如果遇到问题，技术支持团队就可以决定采取哪些纠正动作。

ColdFire MCF5223x用于空调控制和监控

不管是对于家用电器还是工业应用，空调的控制和监控都是一个关系到舒适度和生产力的问题。MCF5223x是一种理想的器件。空调通常由专用于电机控制应用的DSP或DSC进行控制，而MCF5223x除了可以完成这些器件的控制功能外，还添加了远程监控的以太网连接功能。MCF5223x具有PWM模块、定时器和一个12位ADC，这些正是调节电机控制速度和扭矩的合适的外围设备。它还包括以太网控制功能和物理层，另外还有协议栈可对此器件提供诊断、监控和远程控制等全部功能。

结论

在工业、商业甚至消费市场上连接32位控制的应用正在变成一项必然需求，而不只是可有可无的选项。飞思卡尔ColdFire MCF5223x系列可以提供一种单芯片解决方案，通过以太网接口实现32位控制，从而有助于确保安全的连接。这些高性价比的嵌入式控制器可提供合适的外围设备和内存空间，帮助构建紧凑的以太网平台来缩减开发时间和成本，帮助您的产品更快上市。

Lech Olmedo是飞思卡尔RTAC的市场部应用程序工程师。他为ColdFire系列产品提供支持并参与ColdFire和Linux (uClinux)应用程序的连接解决方案开发。他参与的其他项目包括8到16位汽车应用移植等。

Daniel Torres是飞思卡尔半导体的应用工程师。他在数字信号控制器、ColdFire处理器和8位MCU领域具有丰富经验。他主要从事电机控制和电源管理领域的工作。

DSP56800E中的高速32位除法

最小化不复原除法算法

介绍

处理器内核DSP56800E包含多个32位寄存器。执行32位整数除法看起来应当是件简单的任务，但是，如果您不是二进制除法的专家，您可能需要更多支持。DSP56800E使用一种叫作DIV的指令来执行除法运算。该指令代表对除数结果进行1位的迭代除运算。该指令以及16位除法算法的描述可以参见DSP56800E参考手册。

DIV采用32位的被除数和16位的除数，在16次迭代后生成16位的商和16位的余数。迭代运算使用2个32位的操作数，其中16位除数位于操作数的高16位。这种用来计算商和余数的算法称作“不复原除法算法”(NrDA)。

如果我们要将NrDA用于32位除法，我们就需要64位的操作数。在一个DSP56800E这样的16位处理器上用64位操作数实现NrDA是很麻烦的，因为它需要对64位算法的支持。还有一种选择是采用著名的牛顿—拉普森迭代法，这种方法只需要32位算法。但这种方法采用除法，因此需要大量的反复才能得到所需的精度。此外，我们不能确保所有的运算对(被除数和除数)都能得到正确的运算结果，除非对几乎所有可能的输入对都进行过测试。

最小化不复原除法算法(MNrDA)可以用32位运算中高效低实现32位不复原算法。由于DSP56800E中的DIV指令也采用32位数字作为除数，所以我们能高效地实现NrDA算法。

本文将介绍以下内容：

- 介绍基本的复原除法算法
- 说明源自恢复算法的不复原除法
- 介绍如何消除NrDA中不必要的迭代，以实现最小版本的NrDA
- 给出处理器内核DSP56800E实现最小化不复原除法的实例

不复原除法算法

整数除法符号

整除可以定义如下：

$$N = Q \cdot D + R$$

其中，

- N表示分子(被除数)
- D表示分母(除数)
- Q表示商
- R表示余数

为了方便，我们假定N、D、Q都是正数。根据定义R是正数，且 $0 \leq R < D$ 。此外，每个操作数都可以拆解成单独的位，形如 $Q = \sum_{j=0}^{m-1} Q_j \cdot 2^j$ 。

复原除法算法

不复原除法算法(NrDA)源自复原除法。复原算法从分子中连续减去分母得到余数，直到余数进入相应的范围。现在，我们可以得到M位整数复原除法的算法。

余数计算的代数等式为：

$$R = N - (D \cdot Q)$$

我们将Q拆分为位

$$\begin{aligned} R &= N - D \cdot \sum_{j=0}^{m-1} Q_j \cdot 2^j \\ &= N - D \cdot \sum_{j=0}^{m-1} Q_j \cdot D \cdot 2^j \end{aligned}$$

这可以看作从N中减去 $D \cdot 2^j$ ，对于所有 $Q_j = 1$ 。在这个减法运算中，只有当余数仍为正数($Q_j = 1$)时才能成立。所以，我们可以将复原算法公式表示如下。我们设：

$$R(m) = N \text{ 其中 } m \text{ 代表整数大小(用位表示)，如16位和32位。}$$

然后我们计算 $R(j)$ 和 Q_j ，这时 $j = m - 1 \dots 0$ 。如下所示：

$$Z = R(j + 1) - D \cdot 2^j$$

$$\text{如果 } Z \geq 0 \quad Q_j = 1, R(j) = Z$$

$$\text{否则} \quad Q_j = 0, R(j) = R(j + 1)$$

计算出 $R(0)$ 后， $R(0)$ 就是实际余数， Q 包含得出的商。

为了执行复原算法，我们需要临时变量 Z ，而且我们必须要在减法得出负值的那一步复原余数。这是因为我们在同一歩内计算 Q_j 和 $R(j)$ 的值。不复原算法就不是这样了，因为它从 Q_{j+1} 计算 $R(j)$ 。

不复原除法算法

在不复原除法算法中，我们将负数结果用于下一次迭代。为了用公式表示这个算法，我们首先从复原算法入手。假设我们在第 k 步得到一个负余数。

$$R(k) = R(k + 1) - D \cdot 2^k$$

显然 $Q_k = 0$ ，因此在下一步我们应当计算：

$$R(k - 1) = R(k + 1) - D \cdot 2^{k-1}$$

由于 $Q_k = 0$ ，所以我们采用 $R(k)$ 的结果，用加法表示为

$$R(k - 1) = R(k) + D \cdot 2^{k-1}$$

$$= R(k + 1) - D \cdot (2^k - 2^{k-1})$$

$$= R(k + 1) - D \cdot 2^{k-1}$$

因此，我们在不复原算法中用公式表示这个迭代，如下所示：

$$R(j) = R(j + 1) - D \cdot 2^j, \text{ 当 } Q_{j+1} = 1$$

$$R(j) = R(j + 1) - D \cdot 2^j, \text{ 当 } Q_{j+1} = 0$$

$$Q(j) = 1, \text{ 若 } R(j) \text{ 为正数。}$$

$$Q(j) = 0, \text{ 若 } R(j) \text{ 为负数。}$$

我们将上面的两个等式组合起来将迭代步骤的代数式表达为：

$$R(j) = R(j + 1) + (1 - 2 \cdot Q_{j+1}) \cdot D \cdot 2^j$$

开始算法时，我们首先假设 $R(m) = N$ ， $Q^m = 1$ 。在完成 $m - 1$ 迭代后，我们得到 $R(0)$ 。

$$R(0) = N - (2^{m-1} - \sum_{j=0}^{m-1} (1 - 2 \cdot Q_{j+1}) \cdot 2^j) \cdot D$$

$$R(0) = N - (2^0 - \sum_{j=0}^{m-1} Q_j \cdot 2^j) \cdot D$$

显然， $R(0)$ 不是正确的实际余数。因此，在算法结束时，我们需要通过另一次计算来获得余数。

$$R = R(0) + (1 - Q_0) \cdot D$$

最后，我们可以将不恢复算法概括如下：

$$R(m) = N, Q_m = 1 \text{ (初始状态)}$$

对于 $j = m - 1$ 到 0 之间的数

$$R(j) = R(j + 1) - D \cdot 2^j, \text{ 当 } Q_{j+1} = 1$$

$$R(j) = R(j + 1) + D \cdot 2^j, \text{ 当 } Q_{j+1} = 0$$

$$Q_j = 1, \text{ 若 } R(j) \geq 0$$

$$Q_j = 0, \text{ 若 } R(j) < 0$$

结果

$$R = R(0) + (1 - Q_0) \cdot D$$

不复原算法要求转换 D $m - 1$ 次。特别是32位除法， D 必须在左移31位。因此，执行不复原算法需要64位算法。总之， m 位除法需要 $(2 \cdot m)$ 位算法来执行不复原算法。

最小不复原除法算法

最小不复原除法算法(MNrdA)可以用 m 位运算执行 m 位数的不复原算法。

命题1，如果 n 、 d 和 q 分别是 N 、 D 和 Q 的最高有效位的数值，那么 $n - d \geq q$

命题1证明：对于 m 位整数，我们有：

$$N = Q \cdot D + R$$

$$\sum_{j=0}^{m-1} N_j \cdot 2^j = \sum_{j=0}^{m-1} D_j \cdot 2^j \cdot \sum_{j=0}^{m-1} Q_j \cdot 2^j + R$$

$$2^n + \sum_{j=0}^{n-1} N_j \cdot 2^j = (2^d + \sum_{j=0}^{d-1} D_j \cdot 2^j) \cdot (2^q + \sum_{j=0}^{q-1} Q_j \cdot 2^j) + R$$

马上就得到 $n - d \geq q$

推论1 在整数除法中，以 N 为分子， D 为分母。 n 和 d 分别作为 N 和 D 的最高有效位的数值。那么对于所有的 $j > n - d$ ， $Q_j = 0$ 。

推论1证明：这个推论是命题1的直接结果

如果我们将推论1应用到不复原除法算法中，我们可以直接跳到 $R(n-d)$ 的运算，因为前面的迭代得出商位为0。

$$\begin{aligned} R(n-d) &= N - D \cdot 2^{n-1} + \sum_{j=n-d}^{m-2} D \cdot 2^j \\ &= N - D \cdot 2^{n-d} \\ &= N - 2^{n-d} \cdot (2^d + \sum_{j=0}^{d-1} D_j \cdot 2^j) \\ &= N - (2^n + \sum_{j=0}^{d-1} D_j \cdot 2^{j+n-d}) \end{aligned}$$

由于第一次计算得出的部分分数是 $R(n-d)$ ，我们只需要对 D 转换 $n-d$ 次，就能有效地将 D 的最高有效位移动到位置 n 。由于 $m-1 \geq n$ ，我们只需要 m 位来存储转换后的 D 。

即使在最坏的情况下($n=m-1, d=0$)，MNRDA也没有减少迭代数量。既然定时分析通常基于最坏案例执行次数，我们只需假设 $n=m-1$ 即可。因此，MNRDA的执行过程如下：

1. 计算 D 的前导零： l_0
2. 将 D 左移 l_0 位
3. 执行不复原除法算法，从 $R(l_0)$ 开始，假设 $Q_j=0$ ， $j=l_0+1 \dots m-1$



在DSP56800E采用MNRDA实现带有符号数除法

DSP56800E上的DIV指令代表16位复原除法迭代。这个指令采用32位被除数和16位除数。除数会被左移16位，在每次迭代中的除数右移，而不是将被除数左移。 Q_j 结果在每次迭代中都会移入。因此，在执行16次迭代后，余数 $R(0)$ 包含在被除数的高16位中。在代数上可以表示如下。一个迭代步骤为：

$$\begin{aligned} R' &= 2 \cdot R'(j+1) \pm D \cdot 2^n \\ &= 2^{n-j} \cdot R(j+1) \pm D \cdot 2^n \\ &= 2^{n-j} \cdot (R(j+1) \pm D \cdot 2^n) \\ &= 2^{n-j} \cdot R(j) \end{aligned}$$

最后得出：

$$R'(0) = 2^n \cdot R(0)$$

在我们的例子中， $n=16$ ，因此 $R(0)$ 包含在高16位中。商占据了低16位。

为了执行完全的32位除法，我们给DIV指令提供32位的除数并左移 $n-d$ 位，这里 n 和 d 分别是被除数和除数的最高有效位。

$$\begin{aligned} R'(n-d) &= R(n-d) \\ &= N - D \cdot 2^{n-d} \\ R'(n-d-1) &= 2 \cdot R(n-d) - D \cdot 2^{n-d} \\ &= 2 \cdot (R(n-d) - D \cdot 2^{n-d-1}) \\ &= 2 \cdot R(n-d-1) \\ &\vdots \\ R'(n-d-j) &= 2^j \cdot R(n-d-j) \\ &\vdots \\ R'(0) &= 2^{n-d} \cdot R(0) \end{aligned}$$

基于这个原因， $R(0)$ 包含在高 $m-1-(n-d)$ 位中，商包含在低 $n-d$ 位中。如果我们假设 $n=m-1$ ，那么 $R(0)$ 位于高 d 位中，其LSB在第 $(n-d)$ 位上。为了得到商，我们对DIV的目的操作中进行掩码操作。在我们的例子中， $m=31$ ，而且我们假设实际的除法中被除数和除数都是正数。我们采用MNRDA实现四象限除法以得到商。由于从MNRDA中得到余数的复杂度几乎等同于乘以商，再从被除数中减去，所以我们就不计算余数了。这种算法最差情况下的执行时间为60个循环。

```

; A = 有符号的被除数 (分子)
; B = 有符号的除数 (分母)
; A1:A0 / B1:B0
; 程序使以下寄存器 : A, B, D, X0, Y

tfr    B,Y    ; Y=den

; 商符号
eor.l  A,B    ; B的MSB保留符号位

; 准备R(q)和den
abs    A      ; abs (num)
abs    Y      ; abs (den), 因此我们能够计算R(q)
        ; DIV接收负数的S, 但是对于
        ; R(q)计算, 我们对减法进行硬编码

; 移动后的分母和第一个位
clb    Y,X0   ; 计算前导零
asll.l X0,Y   ; Y=den*2^q
sub    Y,A    ; A=R(q), Carry=1, 这时Y>A

; 计算位q; 1, 这时R(q)>=0
bftstl #8,A2  ; 如果是正数, 设置Carry

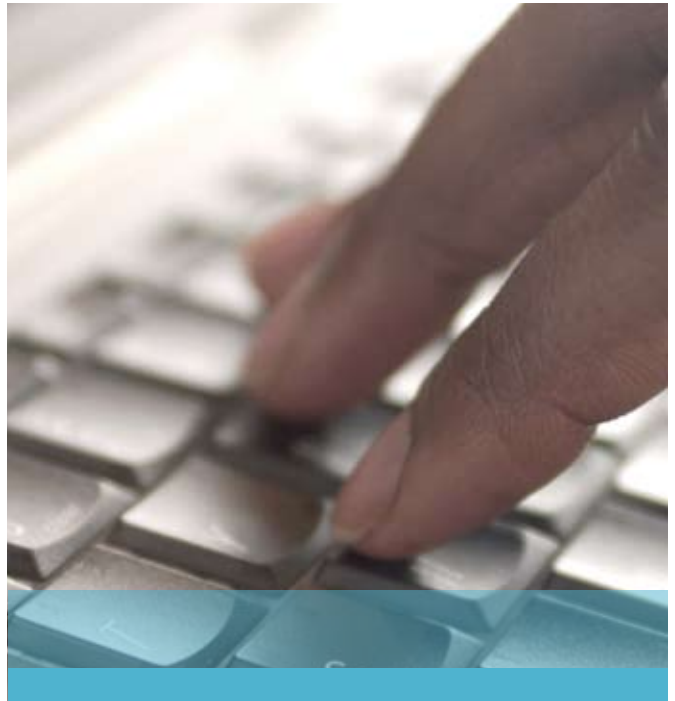
; 除以环数
rep    X0     ; 计算位q-1...0
div    Y,A    ; Carry拥有最低有效位

rol.l  A      ; 最后Carry必须进来
        ; DIV会破坏最高位

; 屏蔽位0...q-1
eor.l  Y,Y    ; 清除Y
add.l  #2,Y   ; 对于X0=0的情况, 我们计算位0
asll.l X0,Y
dec.l  Y      ; Y包含掩码
and.l  Y,A    ; 应用掩码

; 应用有符号的位
brclr #8000, B1, Qpositive
nop
neg    A
Qpositive:

```



结论

在本文中，我们介绍了带DSP56800E内核的飞思卡尔处理器能够执行全精度的32位除法。由于DSP56800E DIV指令能采用32位数作为除数，因此我们能够通过最小化不复原除法算法(MN_rDA)来实现32位除法。DIV指令最初设计用于16位不复原除法。在该除法中，除数的首次移动自动设置为全16位。

与32位除法不同，在16位除法中采用MN_rDA会增加最坏情况下的执行时间，因为计算和执行初始除数移动会或多或少地耗费掉减少的迭代次数。另一方面，对于32位除法，我们超过了16位处理器的寄存器规格，因此MN_rDA是最佳选择。这种方法的另一个好处是迭代次数减少的实际效果，与纯粹的不复原除法算法需要32位迭代的因素之间的平衡。

David Baca是飞思卡尔的应用程序工程师，在飞思卡尔工作了近两年的时间。他拥有电子工程和工商管理硕士学位，目前正在攻读人工智能博士学位。在加入飞思卡尔之前，Baca曾参与过NASA和美国空军等机构赞助的研发项目。

采用BLDC、PMSM和SMPS的电机控制和电源设计

介绍

现在，大部分帮助我们生活得更轻松、更舒适的设备都需要极性运动的控制，诸如洗衣机、冰箱、风扇、空调、电动工具和搅拌器等等。

所有这些设备都需要消耗能量以产生机械运动，而有效利用能源的途径取决于控制系统、电气机构的设计及控制算法等。作为人类，我们面临的一个最大挑战是如何有效地利用能源。在解决这个问题过程中，人们将大部分精力集中在机械运动控制系统上。因此，许多节能方面的进步是通过改进电机控制技术、结构设计、材料和制造精确度来实现的。

早在几年前，人们就已经开发出了更高效的控制技术，但执行这类复杂算法和计算所需的CPU成本较高，不能满足成本敏感市场(如家电市场)的需求。这种情况在最近几年已经发生了变化，成本更低而且具有执行这些复杂的控制算法所需的所有功能的高性能数字信号控制器已经面世。

实现节能的另一个开发领域是功率转换。功率转换系统用于将电能从一种形式转换为另一种形式，在此过程中，由于系统的固有能耗、拓扑结构的效率、控制技术以及所使用的电子元器件，必然会产生一定的能量损失。大部分功率转换控制是由模拟电路实现的，但新的节能法规提出的要求越来越高，使得模拟控制系统越来越难以达到这些要求。

微控制器(MCU)和DSC的使用为此开辟了新的前景。现在，借助数字控制技术和高性能、低成本的数字信号控制器实现的复杂算术计算，功率转换系统的效率达到98%是完全可行的。

机械运动控制设计中的难点

在机械运动控制中会使用多种电机，包括无刷直流电机(BLDC)、有刷换向永磁直流电机、线性电机和步进电机。

系统工程师不但需要选择正确的电机来完成机械动作，同时还必须选择适当的控制环路结构来满足系统的机械和电子时变响应的要求。控制环的调节通常在电子驱动装置的设计阶段进行。

由于不同的电机对电子驱动装置有一系列不同的设计要求，开发人员可能需要处理大量的设计变量。此外，由于电动机的感性特点，它容易造成电磁干扰(EMI)、射频干扰(RFI)和具有破坏性瞬间高能量，因此，电机本身也使电子驱动装置的设计变得更为复杂。这类设备的设计不但要避免电磁干扰(EMI)和射频干扰(RFI)，同时还必须能够承受瞬间过电压和过电流的情形。

BLDC电机已广泛用于许多应用领域。BLDC电机不带换向器，因而比直流电机更可靠。BLDC电机同时也在许多方面优于交流感应电机。BLDC电机通过转子磁体生成旋转磁通，具有很高的效率，因而它们一般用于高端家用电器(如冰箱、洗衣机和洗碗机)、高端水泵、风扇和其它需要较高可靠性和效率的设备中。

由于BLDC电机的结构非常牢固，所以被广泛应用于泵、风扇和压缩机等应用中。这些应用的一个共同特点是它们不需要位置信息，只需要速度信息，而且只需要控制速度。BLDC电机的使用不需要复杂的控制算法。

在BLDC电机中，必须知道转子的位置，才能提供相电压对并控制其电压值。如果使用传感器检测转子的位置，那么检测到的信息必须传送到控制单元中去。

这就需要与电机建立额外的连接，可是这在有些应用中是无法接受的。有时候可能无法建立与位置传感器的物理连接。有时，位置传感器和布线所产生的费用可能是无法接受的。虽然物理的连接问题可以通过在电机内部集成驱动器的方法加以解决，但大量的应用由于低成本的特点，需要无传感器的解决方案。

永磁同步电机(PMSM)可以用一个与电源频率同步的恒定速度进行旋转，而不受负荷或线路电压的影响。电机运行可以保持恒定的、与电源频率同步的速度，只要转矩不超过电机极限运行值。因此，PMSM是高精度定速驱动的理想选择。

3相PMSM是一种永磁电机。它能产生非常高的功率密度、非常高的效率和极好的响应，所以能适应机械工程领域中最复杂的应用。另外它还具有很强的过载运行能力。PMSM基本上不需要维护，因此可以确保最高效的运行。

精确的速度调节使PMSM成为特定工业过程的理想选则。PMSM的速度/转矩特性非常适用于直接驱动大马力、低转速(rpm)的负载。

同步电机能够以更高的功率因数运行，因此能提高整个系统的功率因数，进而能消除或减少功率因数的损失。功率因数的提高还可以减少系统及电机终端的压降。

PMSM舍弃了激励线圈，而且转子的转速与定子磁场的转速率相同。PMSM的这种设计可以减少转子铜损，与传统的感应电机相比可以产生极高的峰值效率。

PMSM的功率重量比也高于感应电机。

功率电子和微电子领域的进步使PMSM可用于高性能的驱动，而这在过去是只能使用直流电机的。

功率转换设计中的难点

电源的主要作用是不论电网的情况如何，都能向负载提供经过调节的稳定的能量。由于具有很高的效率和功率密度，开关模式电源(SMPS)被广泛用于办公设备、计算机、通信系统和其它应用中。

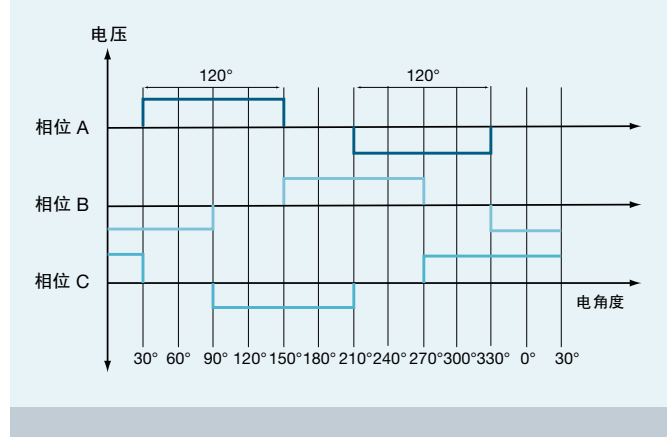
由在数字信号控制器(DSC)上运行的软件进行完全数字化控制的SMPS比混合模拟器件和处理器控制的方案具有更多的优势，包括可编程性、适应性、更少的元器件数量、设计重复使用性、处理的独立性、高级调整功能和更好的性能等。

采用完全数字化的控制后，SMPS系统变得更加灵活，能够实现复杂的控制算法，从而提高效率并降低成本。基于控制器的SMPS系统将功率电子技术与高性能数字信号处理能力相结合，提供了一种全新的功率电子设计方法，还提供了SMPS系统经常需要的高级控制和通信功能等。

无刷直流电机的控制原理

BLDC电机是一种电气旋转机械，具有类似于感应电机的经典三相定子。其转子表面装有永磁体。它也称为电子换向电机。转子上没有电刷，换向在特定的转子位置上以电子方式完成。定子通常由磁铁薄片制成。定子相位绕组插入凹槽中(分布式绕组)，或作成绕在磁极上的线圈。由于气隙磁场是由永磁体产生的，所以转子磁场恒定不变。永磁体的磁化及其在转子上的分布是经过选择的，可以确保反电势(定子绕组中由于转子运动而感应的电压)的波形是梯形的。这样就可以采用具有矩形波形的直流电压来生成低转矩纹波的旋转磁场。

图1: BLDC电机的三相电压系统



这种电机的每个相位可以有对磁极。每个相位的磁极对的数量决定了电气旋转与机械旋转之间的比率。例如，图中所示的BLDC电机每个相位上有三对极对，表示三次电气旋转对应于一次机械旋转。

由于所施加的矩形电压很容易产生，所以电机控制和驱动变得简单。但是必须知道转子的位置在某个角度，才能让所施加的电压与反电动势对齐。反电动势与换向动作对齐是非常重要的。只有这样，电机才能作为直流电机以最高的效率运行。因此，控制和实现的简单使BLDC电机成为低成本、高效率应用的最佳选择。图1描绘了加在三相BLDC电机上的波形。

永磁同步电机的控制原理

借助诸如矢量控制等复杂的控制方法，PMSM可以提供与高性能四象限直流驱动电机相仿的控制功能。

矢量控制是永磁同步电机的一种先进控制方法，这种方法利用电机磁场定向的原理来控制磁通量、电流和电压的空间矢量。可以建立一个坐标系，将矢量分解成磁场产生的部分和转矩产生的部分。这样，电机控制器(即矢量控制器)的结构几乎与分励直流电机完全相同，从而简化了永磁同步电机的控制。过去开发这种向量控制技术的目的是在同步电机中达到同样出色的动态性能。

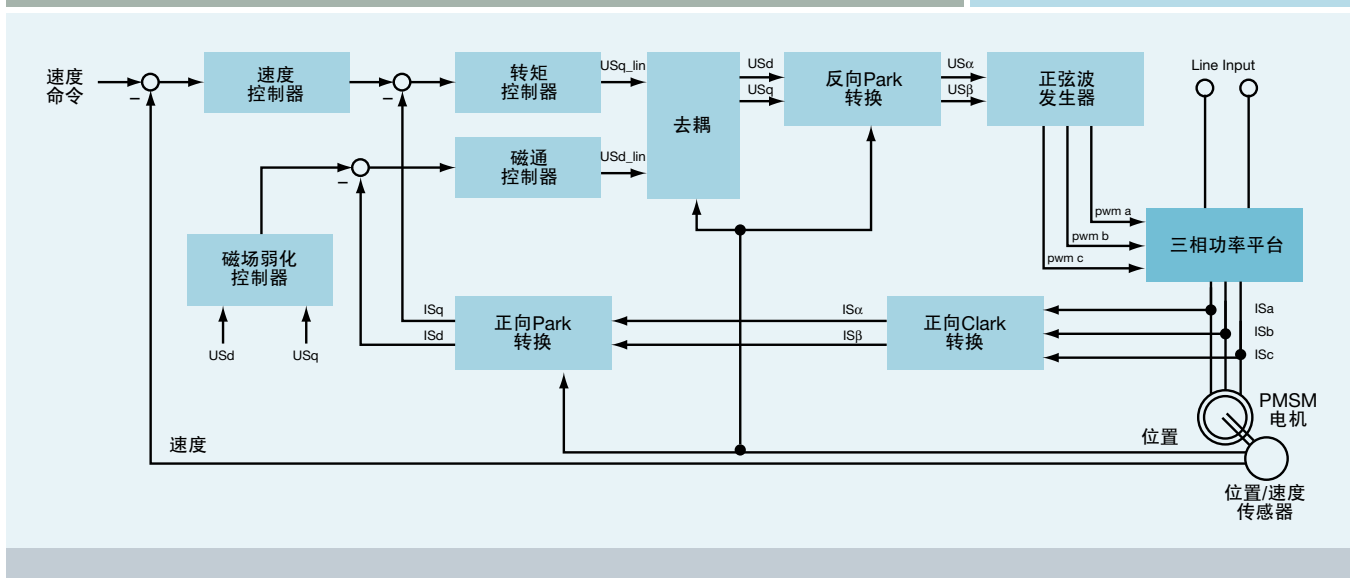
在此方法中，必须将定子电流分解为的产生磁场的部分和产生转矩的部分，以便分别控制磁通量和转矩。要做到这一点，必须建立与转子磁场关联的转子坐标系。该坐标系通常称为“d,q坐标系”。将转子坐标系转换成定子坐标系通常需要很强的CPU运算性能。

图2: 给出了CPU在矢量控制技术中需要执行任务的方框图

要执行矢量控制，必须做以下步骤的工作：

- 测量电机的数值(相电压和电流)
- 用Clarke转换将它们转换成两相系统(α 、 β)。
- 计算转子磁通空间矢量的大小和位置角度
- 用Park转换将定子电流转换成d,q坐标系
- 定子电流的转矩(i_{sq})和磁通量(i_{sd})分量由控制器分别进行控制
- 用去耦模块计算定子电压空间矢量的输出值
- 通过Park逆转换将定子电压空间矢量从d,q坐标系转换回固定于定子的两相系统
- 用正弦调制生成三相输出电压

图2: 永磁同步电机向量控制示意图



56F8013 DSC的优点和特性

飞思卡尔的MC56F801x DSC系列在单个芯片上结合了DSP的计算功能和MCU的控制功能，非常适合数字电机控制。这些混合控制器提供多种专用外设，如脉宽调制(PWM)模块(组)、数模转换器(ADC)、定时器、通信外设(SCI、SPI、I²C)、内置闪存和RAM等。

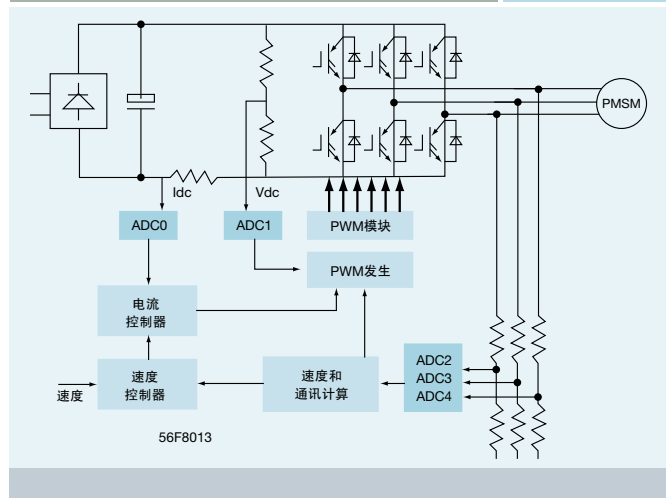
- 32 MHz运行频率时高达32 MIPS
- 片上内存包括高速易失性和非易失性组件
- 16KB编程闪存
- 4KB 统一数据/程序RAM
- 闪存加密功能防止未经授权访问其内容
- 闪存保护防止意外修改
- JTAG/增强型片上仿真(EOnCE)，提供无干扰实时调试
- 4个36位累加器
- 16位和32位双向桶型移位器
- 具有特殊寻址模式的平行指令集
- 提供硬件DO和REP循环
- 3个内部地址总线
- 4个内部数据总线
- 支持MCU类型的软件堆栈
- 控制器类型的寻址模式和指令
- 单周期16 x 16位平行乘-累加器(MAC)
- 已证明能采用比竞争性架构更小的内存面积，提供更多控制功能

机械运动控制的推荐方案

BLDC系统结合了交流和直流电系统的优点。与有刷直流电机不同，BLDC系统采用的电机一般是一种具有梯形反电动势波形的永磁交流同步电机，而且用电子换向取代了直流电机中的机械电刷。尽管这种控制方法在相位换向过程中会产生转矩跳变，但它还是能满足以控制转子速度为目标的大部分应用的需要。

反电动势波形为正弦波的PMSM电机也可应用于BLDC系统。但是，定子磁通和转子磁通之间的矢量角度应保持在60°和120°电角度之间。运行期间会产生转矩波纹，但平均转矩可维持恒定，从而能满足大部分低端应用程序的要求。图3显示了既可用于实现PMSM矢量控制，也可用于实现BLDC电机控制的方框图。

图3: PMSM/BLDC电机控制方案通用方框图



反电动势的过零信息可用于确定转子的位置，以实现正确的换向，决确定开通哪个功率晶体管以获得最大的电机转矩。成本最低且最可靠的反电动势过零信息的采样方法是将电阻网络采集的反电动势信号发送到ADC输入端或GPIO。在有传感器控制结构中，转子每转动60°电角度换向一次。这意味着只需6个换向信号就足以驱动BLDC电机。此外，有效的控制还需要Bemf相与电源相位之间的同步，要确保Bemf只在不供电的60°区间内过零一次。

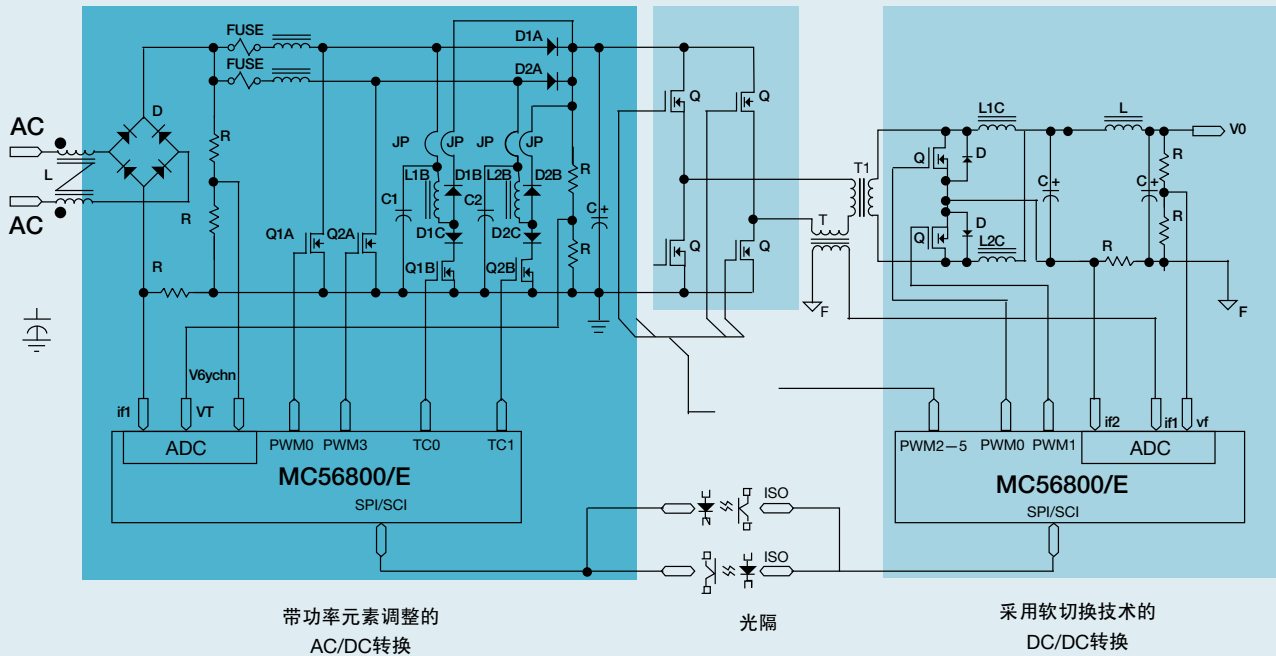
由于在某一时刻定子线组中只有两个有电流，所以这两相电流是相反的，而第三相位电流为0。既然知道三个定子电流总和等于0(星形绕组定子)，就可以计算预期的Bemf瞬间波形。三个定子端的电压之和是中心点电压(Vn)的三倍。各个Bemf在每次机械旋转中过零两次。由于Bemf的数值计算比较方便，再由于56F8013的信号处理能力，因此才能够得到与换向有关的6个必需的信息。

开关模式电源的推荐方案

一般的SMPS AC/DC系统包含两大部分：前端是带功率因数调整(PFC)的AC/DC转换器；后端是一个全桥DC/DC转换器。AC/DC系统采用交错型PFC升压控制结构，包括一个全桥整流器、两个平行交错升压PFC电路和两个辅助开关，以实现主开关的过零切换(ZVS)。采用ZVS算法可减小器件的压力，并提高效率，还可以在设计中省去反向恢复输出二极管。DC/DC转换器采用以软件实现的ZVS移相全桥控制结构，带有倍流整流器。这样不仅减小了滤波电感的尺寸，还提高了效率。

图4是一个基于56800/E的开关模式电源(SMPS)的电路

图4: AC/DC方框图



框图。整个系统由两个56800/E芯片进行控制。前端芯片负责PFC系统的全面控制，包括两个主开关和两个ZVS开关。后端芯片负责DC/DC移相全桥转换器的全面控制，包括4个主开关和2个同步整流器。软件中用于PFC和DC/DC转换器的功能包括：电源系统中的2个数字PI调整功能，所有开关的控制、软启动、前端PFC正弦参考数字信号的发生、通讯、电源保护和监控功能。

设计参考手册DRM070

“采用MC56F8013实现三相BLDC电机的无传感器控制”
飞思卡尔半导体

设计参考手册DRM077

“采用56F8013器件实现PMSM和BLDC的无传感器控制”
飞思卡尔半导体

参考资料

www.freescale.com

应用笔记AN1931

“采用DSP56F80x实现三相永磁同步电机的矢量控制”

飞思卡尔半导体

引用笔记AN3115

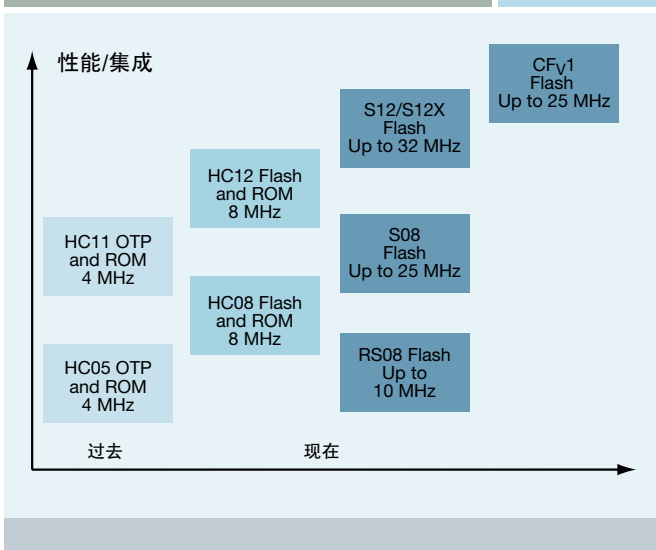
“采用56F8300数字信号控制器实现数字AC/DC开关模式电源”

飞思卡尔半导体

Daniel Torres是飞思卡尔半导体的应用工程师，在数字信号控制器、ColdFire控制器和8位MCU领域拥有丰富的经验。他主要从事电机控制和电源管理。

最新的S08/RS08工具 新一代SPYDER已推出

图1: 内核路线图



飞思卡尔半导体是全球领先的嵌入式半导体设计和生产商，产品广泛应用于汽车、消费、工业、网络和无线市场。过去几年时间里，我们越来越重视为大众市场和专业市场提供工具。随著我们新增了电子商务网站、馈赠样品、免费编译器和调试器、以及经济高效的硬件工具等，设计人员现在很容易接触到飞思卡尔的微控制器系列。SofTec Microsystems™推出最新的工具，2007年价格仅为29美元的USBSPYDER08证明了我们提供小型、快捷、经济高效的硬件工具的承诺，助您的设计启动。

上个世纪80年代推出的HC05和HC11 8位内核目前已经广泛应用到市场的各个主要领域。在90年代末又推出了HC08 (8位)和HC12 (16位)内核，它们现在更受某些特定的消费和工业应用客户的欢迎。2000年初，HCS12 16位内核和HCS08 (S08) 8位内核引进了一项重要功能，即后台调试模块(BDM)。该模块使开发过程变得更轻松。图1是飞思卡尔内核路线图的一部分。

最新推出的RS08(精简的HCS08内核，通过减少指令和其他降低成本的方法使物理体积减小)和8位S08微控制器包含单线后台调试接口，支持片上非易失性内存的在线编程和复杂的无干扰调试功能。该模块有助于开发经济高效、使用方便的工具。同样的BDM连接还出现在刚刚

推出的32位微控制器ColdFire V1内核产品上(如需了解更多信息，请参见www.freescale.com/flex)。

2005年，freegeeks.net为HCS12微控制器社区提供了一种名为TBDML(Turbo BDM Light)的开放源代码工具。在12个月的时间内，该产品的下载次数就达1454次，我们意识到它对工程师非常有用。现在，我们为8位MCU也提供了同等具有BDM功能的工具，可以通过两种形式得到：

- 针对S08的开放源代码BDM (OSBDM) — 详细信息可见于飞思卡尔论坛(www.freescale.net/forums)。这种自制工具的物料单(BOM)成本不到10美元，可以购买由小公司预先制作的产品。
- USBSPYDER08开发工具包支持MC9S08QG、MC9S08QD和MC9RS08KA 8位微控制器。这套工具的设计支持飞思卡尔未来的8位、8管脚MCU。

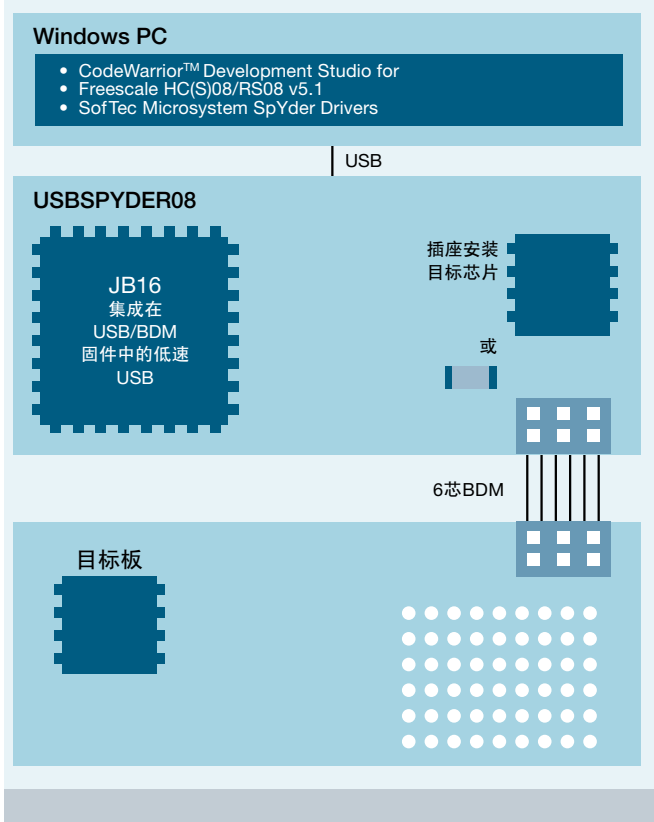
OSBDM和USBSPYDER08在本质上是一样的。它们是您的开发环境(基于Windows PC)和目标MCU之间的接口，如图2所示。这些工具的主要目标是提供更便宜、更方便的设计，供爱好者选用。

USBSPYDER08是飞思卡尔新的USB-to-BDM开发工具。如果您不熟悉BDM，它与新近推出的8位和16位MCU上使用的debugWIRE或JTAG类似。

这些芯片上的后台(BKGD)引脚提供一个单线的后台调试接口，连接到片上调试模块。如需了解这些调试模块的更多信息及使用方法，请参见S08或RS08说明书的开发工具章节。这个接口是单线的，通常是一个6芯的接插件，称为BDM端口，用来和目标板的接口。如图3所示。

BKGD脚的主要功能是用于后台激活模式命令和数据传说的双向串行通信。复位时，这个管脚用来选择是启动进入后台激活模式，还是启动用户的应用程序。此外，这个管脚还要求发出定时同步响应脉冲，以便开发工具主机确定正确的后台调试串行通信时钟频率。后台调试控制器(BDC)命令从主机连续发送到目标S08或RS08 MCU的

图2: 开发环境

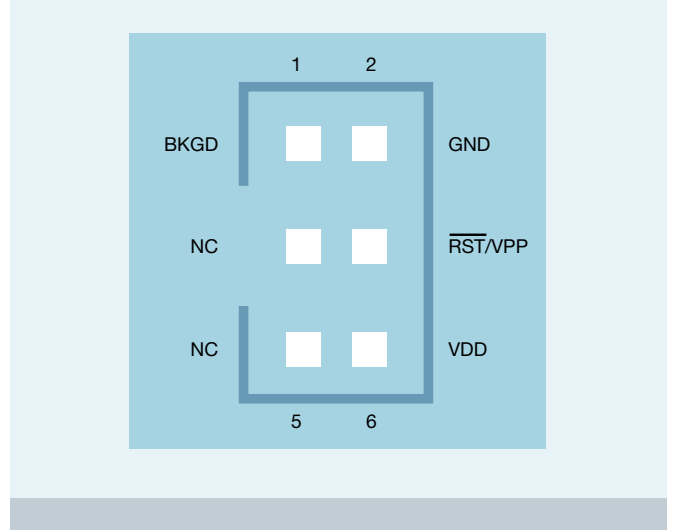


BKGD管脚上。这些命令和数据采样自定义的BDC通信协议发送，最高有效位(MSB)先出。通过单线的后台调试接口，可以采用相对简单的接口盒，将主机的命令转化为BDC的命令。

在USBSPYDER08中，使用的是低速通用串行总线(USB)接口。为了方便，此工具采用USB闪存棒的形式。

USBSPYDER08是由SofTec Microsystems构建的一种经济高效的工具，飞思卡尔半导体提供了全面协助。USBSPYDER08与获奖的CodeWarrior®开发工具一道使用，为您编辑、编译、下载、在线仿真和调试代码提供必需的基本工具。全速率程序执行允许开发人员以实时方式执行硬件和软件测试。这个工具运行的总线速率可高达10MHz，支持3.3V的MCU运行电压。它的电路板上有一个插座安装的MCU，可以替换为其他能支持的PDIP封装器件，这些器件可以从www.freescale.com网站上申请少量样品。为了提高此工具的灵活性，它安装了一个BDM接头，可以对能支持的其他封装的产品进行板外的调试，或者在开发需要时可以与其他电路板元器件协同工作。

图3: BDM连接器

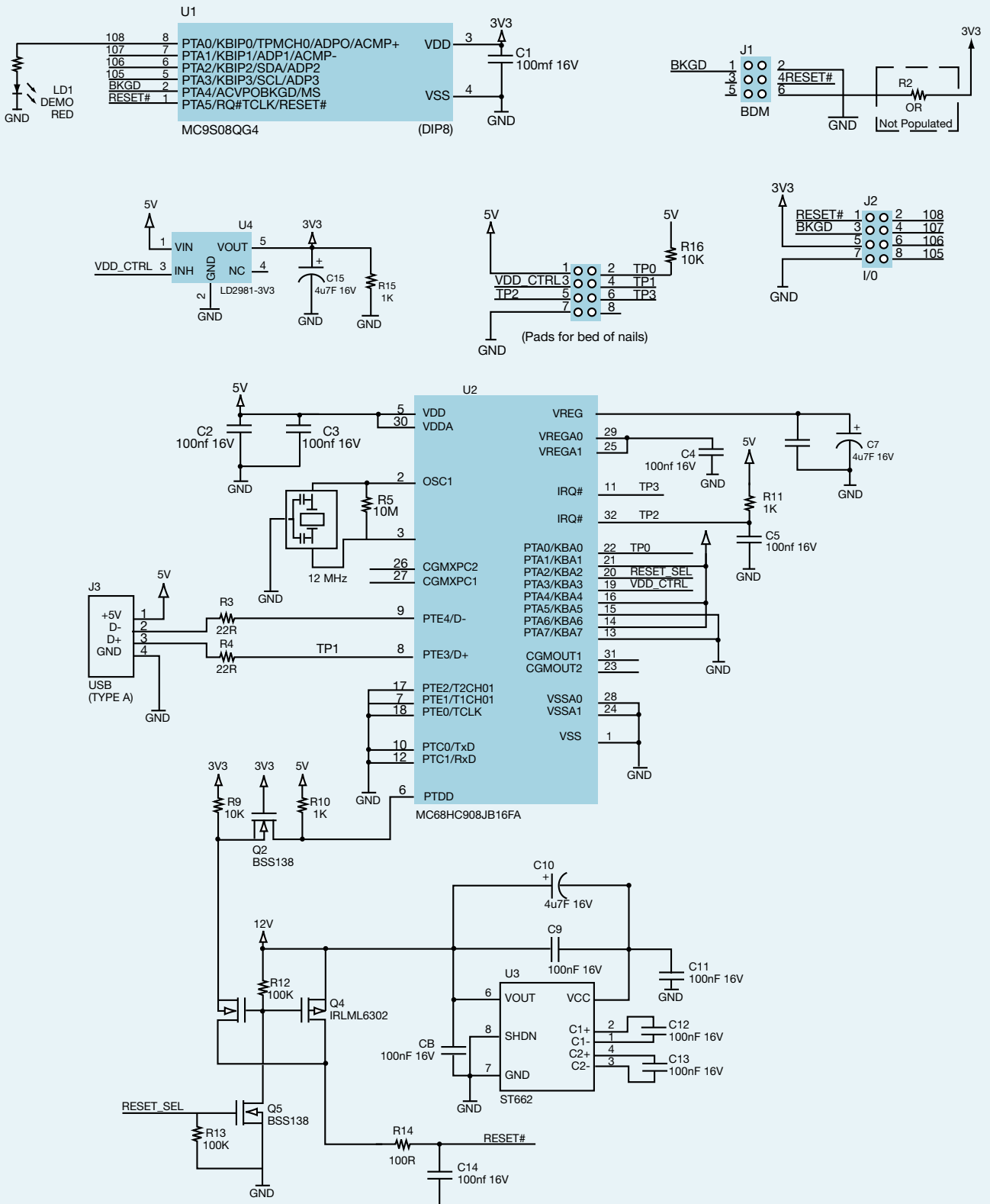


USBSPYDER08是如何工作的

依靠S08和RS08后台调试控制器(BDC)模块，USBSPYDER08能够快捷、便利地对片上闪存及其他内存进行编程。它是进行开发的主调试接口，可以对内存数据进行无干扰的访问，还具有传统的调试功能，如CPU寄存器修改断点以及单指令跟踪命令等。

该工具采用USB接口与PC通信，采用USB总线电压给工具和微控制器供电，不再需要过去的体积庞大的适配器，从而成为真正的便携式设备。如果不用拖动太多电机运行的话，USB电源也可以给目标板供电(高达100mA)。MC9S08JB16 (JB16)微控制器处于USBSPYDER08工具的核心。该MCU带有USB(2.0低速)接口，借助USB提供的5V电压运行。当该工具支持RS08 MCU时，需要12V电压进行编程。JB16识别目标芯片，然后通过PTD0的控制用DC-DC转换器芯片启动12V信号。

图4: SpYder原理图



第1步—USBSPYDER08 电路板组装

USBSPYDER08工具的一些基本部件已经安装在该产品上，您可以通过一个插座安装的微控制器，把它当作一个独立的工具来使用。如果需要采用其他的封装形式或独立的目标板，它同样也支持。在标为R2的空焊盘上加上0 ohm的电阻或短路线，就能把电源连接到BDM插座上，这样您就能使用另一个目标板了。最终的结果是“USBSPYDER08探索工具包”变成了BDM盒。

第2步—调试器安装

USBSPYDER08 CD中提供有免费的CodeWarrior Development Studio for Freescale HC(S)08/ RS08 v5.1特别版。如需了解本工具的更多信息，请访问 www.freescale.com/codewarrior。如果没有证书密钥，本产品将以演示模式运行，代码大小仅为1KB。要突破1KB限制，有两种选择：

- 联系飞思卡尔获得无时间限制，免费的证书密钥，可将代码容量限制增加到16KB。
- 联系飞思卡尔获得30天限制的免费证书密钥，可以在无代码容量限制的情况下运行编译器。

第3步—驱动器

一旦电路板做好了开发准备，CodeWarrior也已安装，接下来就是让USBSPYDER08与调试器进行通信。通过随附的CD安装驱动后，当该工具就与PC首次连接时，Windows会识别新USB设备，“Windows新硬件向导”对话框打开，要求提供需要的驱动。开发人员只需选择“自动安装”选项就可完成安装了。

第4步—准备探索

值得赞叹的是，USBSPYDER0使用目标MCU而非仿真器执行在线调试。这样，微控制器的外围设备(如计时器、A/D转换器、串行通信模块等)就不需要通过软件或外部器件来重新构建了。

为某个USBSPYDER08支持的器件用CodeWarrior工具创建新项目，首先应从CodeWarrior工具入手。从主菜单上选择“文件>新项目”(File > New Project)或在启动选项框中点击“创建新项目”(Create New Project)。然后会出现一个对话框，提示从下拉菜单上选择目标器件。”选择“SofTec HCS08”作为连接类型。然后选择代码类型、项目名称和位置。如果不需要添加新文件，而且

您又不想使用器件初始化工具，请按“完成”(Finish)。

CodeWarrior将创建项目，建立代码框架，引用带有所有寄存器定义的头文件。一旦您写好代码，准备开始调试，请“制作”(make)文件，即开始编译。可以通过多种方法完成该操作，例如可以使用窗口的快捷键，使用下拉菜单“项目”(Project)，或按F5键，然后按F7键。

首次进入调试过程时，系统会出现“MCU配置”(MCU Configuration)对话框，提示选择连接到您的PC的调试硬件。连接类型一定要选择“USBSPYDER08”。这时，CodeWarrior工具已经擦除内存并重新编程了，如果选定的MCU具有相应的功能，它会调整振荡器。

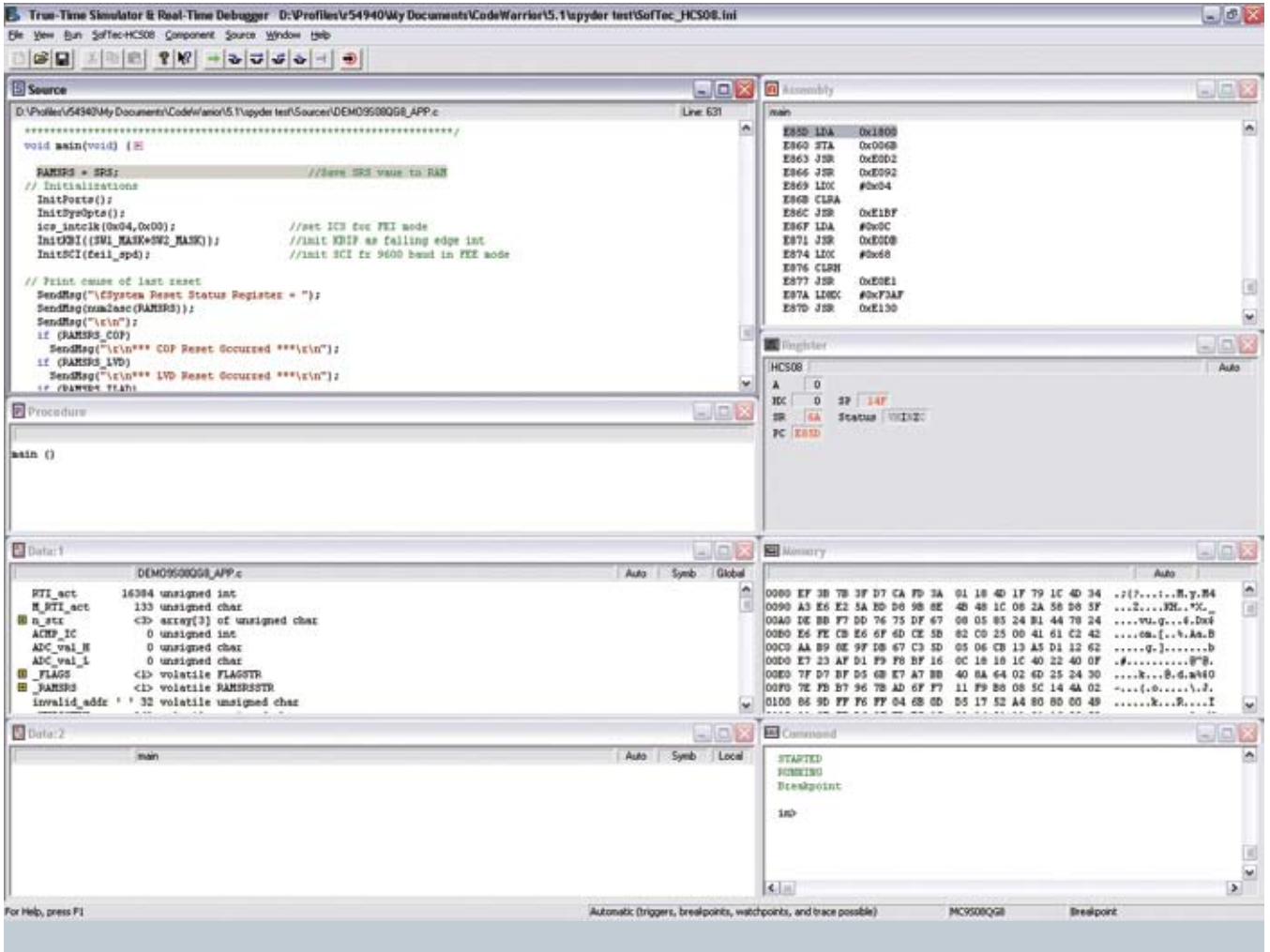
在这个阶段，您已经拥有开始调试代码所需的工具。CodeWarrior工具套件提供了多种不同的方式，如断点、观察点和跟踪缓存等来分析程序流程。所有这些功能都是利用目标微控制器的调试外设来实现的。图5显示了一个CodeWarrior窗口，它包括：

- 显示您的代码的源代码窗口
- 汇编窗口，在此您可以看到编译器由您的源代码生成的汇编代码
- 可以看到CPU寄存器的寄存器窗口
- 内存窗口，在此您可以观察任何位置，或者强制字节为另外的值
- 显示数据、进程和命令的其他窗口

右击窗口并修改模式或格式，可以修改数据格式和数据刷新率。通过“文件>保存配置”(File > Save Configuration)可以保存首选设置。不熟悉CodeWarrior工具套件的新用户需要花一点时间先熟悉Start/Continue, Single Step, Step Over, Step Out, Assembly Step, Halt和Reset Target按钮。在熟悉调试环境时，诸如AN3335—Introduction to HCS08 Background Debug Mode和AN2616—Getting Started with HCS08 and CodeWarrior Using C这样的应用笔记都是不错的资源。

MCU旁边的焊盘可以连接到某个范围，允许实时地监控调试器外部的管脚。

图5: CodeWarrior屏幕截图



小结

USBSPYDER08探索工具包是一种非常灵活的工具，它体积小，使用简单，能够满足低预算开发要求。它支持飞思卡尔的8脚S08器件，也能选择片外目标来支持扩展引脚的器件。当与CodeWarrior工具套件(IDE、编译器、调试器、编辑器、连接器、汇编器、运行控制器件)提供的强大能量结合后，飞思卡尔为大众和专业市场提供了构建平台和应用程序所需的关键工具。

令人愉悦的开发之旅！

更多阅读资料

AN3335

Introduction to HCS08 Background Debug Mode

AN2616

Getting Started with HCS08 and CodeWarrior Using C

BR8BITLOWEND

A Little 8-bit Goes a Long Way

HCS08QRUG

HCS08 Peripheral Module Quick Reference

RS08 Peripheral Module Quick Reference

A Compilation of Demonstration Software for RS08 Modules

Inga Harris 2000年毕业于Strathclyde大学，获得电子和电气工程荣誉学位。

她在飞思卡尔曾先后担任过8位MCU系列的技术市场开发人员和应用工程师，重点研究消费类和工业市场。

适用于传感与控制应用的 ZigBee® 和低成本无线网络

图1: 可以与各种微控制器一起使用的独立收发器示例 (飞思卡尔半导体MC1320X)

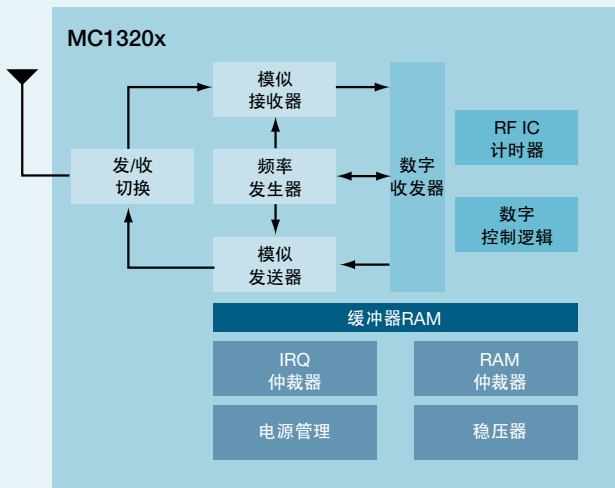
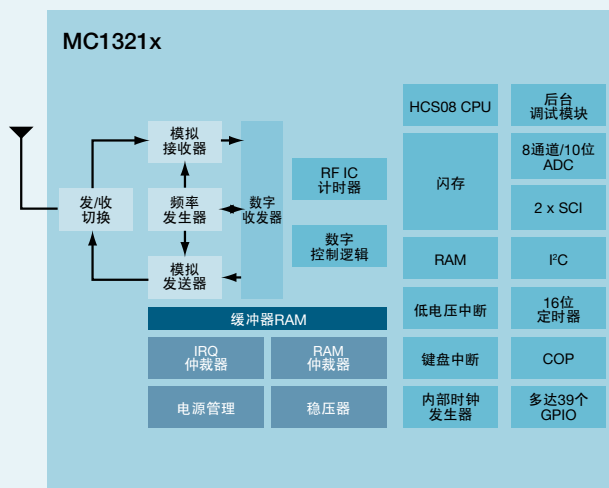


图2: 封装(SIP)在一起的收发器和微控制器集成系统示例 (飞思卡尔半导体MC1321X)



许多感应、监控和控制应用程序都具有融合低成本无线网络功能的独特潜力。低成本无线网络解决方案通常需要30~70米或者更短的距离，250kbps及以下的数据速率，在很多情况中还要求实现最优化电池使用寿命的功能，特别是对于终端节点功能而言。如果在开始设计之前能够提前对几个关键因素进行预先的分析，无线网络的实现会更有效。这些关键因素的阵列会帮助您选择合适的元器件和方案。这里也提供了一些参考设计原理图作为开始设计的基准。在制定您的设计要求时，请考虑检查如下几个方面的问题：

- 集成度
- 无线网络拓扑结构(图3)
 - 射频(RF调制解调器或收发器)
 - 性能
 - 工作电压
 - 数据速率
 - 范围
 - 通道灵活性
 - 输出功率
 - 敏感度
 - 电源管理
 - 外围设备
 - 时钟
 - 多层软件
 - 硬件和软件设计的简单性
 - 天线设计
 - 封装
- 微控制器(MCU)
 - CPU特性
 - 性能
 - 内存选项
 - 电源管理
 - 时钟源选项
 - 数模转换
 - 外围设备
 - 封装
 - 在线调试和编程
 - 软件和硬件设计的简单性

这些分析将为工程决策提供一个结构清晰的前景，为设计成功铺就一条宽阔的道路、为产品面市缩短宝贵的时间，为低成本无线网络的实现提供一个更简单的方法。

大量可选的低成本网络方案能够为您在设计过程中提供高层次上的灵活性。作为一种选择，可以考虑大量供应商提供的各种独立收发器的结构，它们可以选择与大量的各种MCU(图1) 结合使用。另一种同样有效的选择是考虑使用最新解决方案，即提供收发器/MCU集成的产品(图2)。如果您的工作涉及多个相似的终端产品，设计部件和工程投资的再利用应该非常重要。因此，解决方案的结构性评估选项就既具有成本效率又具有资源效率。考虑周全的研究可以为在单一平台上设计多种终端产品提供良好的基础。

无线网络技术

2.4 GHz的工业、科学和医学(ISM)波段支持多种短距离无线网络技术。每种技术的开发都是为了更好地满足特定应用或功能的需求。与2.4 GHz频率范围有关联的最为常见网络拓扑结构是Bluetooth™、WiFi™、ZigBee及其他专用解决方案。非标准的专用解决方案具有一定风险，这些解决方案与供应商具有密切关系，因而随时可能发生变化。

ZigBee联盟定义的ZigBee 是基于IEEE 802.15.4标准的解决方案，是专门为支持感应、监控和控制应用程序而开发的。ZigBee解决方案具有一些明显优势，例如具有功率低、通信稳定和自愈网状网等。ZigBee解决方案的频率通常在868/915 MHz或2.4 GHz频谱范围内。

ZigBee技术解决方案的数据速率是250 Kbps。使用碱性电池或锂电池，功耗必须非常低才能使电池使用寿命以年计(相当于电池保存期限)。理论上ZigBee技术最多可以支持65000个节点。ZigBee技术解决方案支持的最佳感应、监控和控制应用通常包括：

- 个人和医疗监控
- 保安、访问控制和安全监控
- 过程感应和控制
- 供暖、通风和空调(HVAC)的感应和控制
- 家庭、楼宇和工业自动化
- 资产管理、状态和跟踪
- 健身监控
- 能源管理

图3: 无线网络技术

	ZigBee®	Bluetooth®	UWB™	Wi-Fi™	LonWorks®	专有
标准	IEEE® 802.15.4	IEEE 802.15.1	IEEE 802.15.3a (to be ratified)	IEEE 802.11 a, b, g (n, to be ratified)	EIA 709.1, 2, 3	专有
行业组织	ZigBee Alliance	Bluetooth SIG	UWB Forum and WiMedia™ Alliance	WiFe Alliance	LonMark Interoperability Association	无
网状、星状、树形拓扑	星状	星状	星状	媒体相关	P2P、星状、网格	
RF频率	868/915 MHz 2.5 GHz	2.4 GHz	3.1–10.6 GHz (U.S.)	2.4 GHz 5.8 GHz	N/A (wired technology)	433/868/900 MHz 2.4 GHz
数据速率	250 Kbps	723 Kbps	110 Mbps–1.6 Gbps	10–105 Mbps	15 Kbps–10 Mbps	10–250 Kbps
范围	10–300m	10m	4–20m	10–100m	Medium-dependent	10–70m
功率	很低	低	低	高	有线	很低-低
电池使用(寿命)	硷性电池 (几个月/年)	可充电 (几小时/天)	可充电 (几小时)	无	硷性电池 (几个月/年)	
节点	65,000	8	128	32	32,000	100–1,000

RF调制解调器或收发器(射频)

在构建低成本无线网络系统时，射频(RF)调制解调器的几个特性是需要考虑的。大部分低成本的个人局域网(PAN)RF调制解调器解决方案都推荐使用2.0-3.6V的电源。

对轻质无线网络来说，低数据速率就足以支持监控、感应和控制功能，同时也有助于管理系统功耗。在这类应用中，最常选择的是2 MHz通道上的250kbps积分偏置相移键控(O-QPSK)数据，通道间有5MHz的间距，在全频谱范围进行编码和解码。在这种环境中，收发器醒来后，侦听开放通道并以较低数据速率发送小型数据包。然后收发器会关闭，直到遇到下一个事件。顺序、电源快速开启延时、较低数据速率和小型数据包等因素让802.15.4可以选择合适的时间增量，使数据传输达到最高效率。

正如前面提到的，对于感应和控制子系统，ZigBee技术是支持数据传输距离和功率要求的最佳方案。ZigBee联盟规范定义的典型距离是10-70m，但许多解决方案可提供远高于该范围的视线距离。

在相关的设计规划中，仔细检查可用的收发器通道的数量和类型是非常重要的。可选择的收发器通道使设计人员可以利用不同的通道来最大限度地降低噪声，特别是远离更加拥挤的2.4GHz WiFi信道。

您还应检查一下0 dBm以上到+4 dBm范围内的典型发送输出功率。通常在-90 dBm范围的接收灵敏度能够对感应、监控和控制功能已经足够。数据包的发送和接收带有缓冲，这能够简化用于收发器的低成本微控制器的管理。射频和收发器还应提供连接质量和能量的检测功能，以便网络性能的评估。

多种低功耗模式可提供省电功能，用以将系统能耗降至最低。这些模式一般包括关闭电流(off current)、休眠电流(hibernate current)和半休眠电流(doze current)，其电流可达个位数微安(uA)级。当范围或环境需要较低的功率就能完成发送和接收目标时，可编程输出功率的功能还允许设计人员减少功耗。确保选定的解决方案能够提供这些功能将有助于最大限度延长全功能/协调器或终端节点设备中的电池使用寿命，使得电池寿命可以达到整个保存期限。

检查可以减少MCU资源要求的其他基本外围设备(如内部定时器比较器)。在各种不同的配置和引脚数类型中都有通用输入/输出端口(GPIO)。在实际应用中GPIO很大程度上根据与其他设备的接口要求来决定。在那些采用收发器和独立的MCU实现灵活性的解决方案中，通信由串行外设接口(SPI)端口处理。可以预见的是，当射频和MCU集成到单个封装或芯片中时，收发器与MCU通过内置或内部SPI命令通道进行通信。而且，集成解决方案还带有低噪声放大器(LNA)、带内部电压控制振荡器(VCO)的功率放大器(PA)、集成的发送/接收开关、内置稳压电源和全频率范围的编码和解码功能，所以能减少系统外部器件需求，降低整个系统成本。

各种各样的系统时钟配置使您能够灵活地进行最终系统设计。最适用的是能通过外部时钟源或晶振进行CPU定时的选择。调制解调器时钟通常需要16 MHz的外部晶振。调制解调器晶振频率的调整功能有助于确保符合IEEE802.15.4规范要求的严格标准。

根据最终设计的复杂性和高要求，能够提供多种网络软件拓扑结构选择的供应商才能为您提供最好的服务。这些选择可能包括简单的媒介访问控制器(MAC)结构，它采用4 KB或以上的MCU闪存，支持点到点或简单的星形网络。完全符合802.15.4标准的MAC和完全兼容的ZigBee拓扑会需要更多内存，能够为网状和集群树状网络提供更多的支持。

采用供应商提供的参考设计、硬件开发工具和软件开发工具可以简化设计过程。对于硬件开发工具，简单的入门指南、带LED或LCD，可进行视图化监控的基本电路板、还有线缆和电池等，可提供轻松的开箱即用体验。这些工具能帮助您在几分钟内构建一个网络，并能确实地评估网络和方案的性能。过去，一些软件设计工具，特别是那些支持完全符合ZigBee标准的网络的工具，使用起来非常困难。要降低准备RF调制解调器的复杂程度，应该寻找提供基于图形化用户界面(GUI)软件设计工具的供应商，该工具应能引导设计人员逐步完成收发器的搭建。

图4: 13192 SARD参考原理图

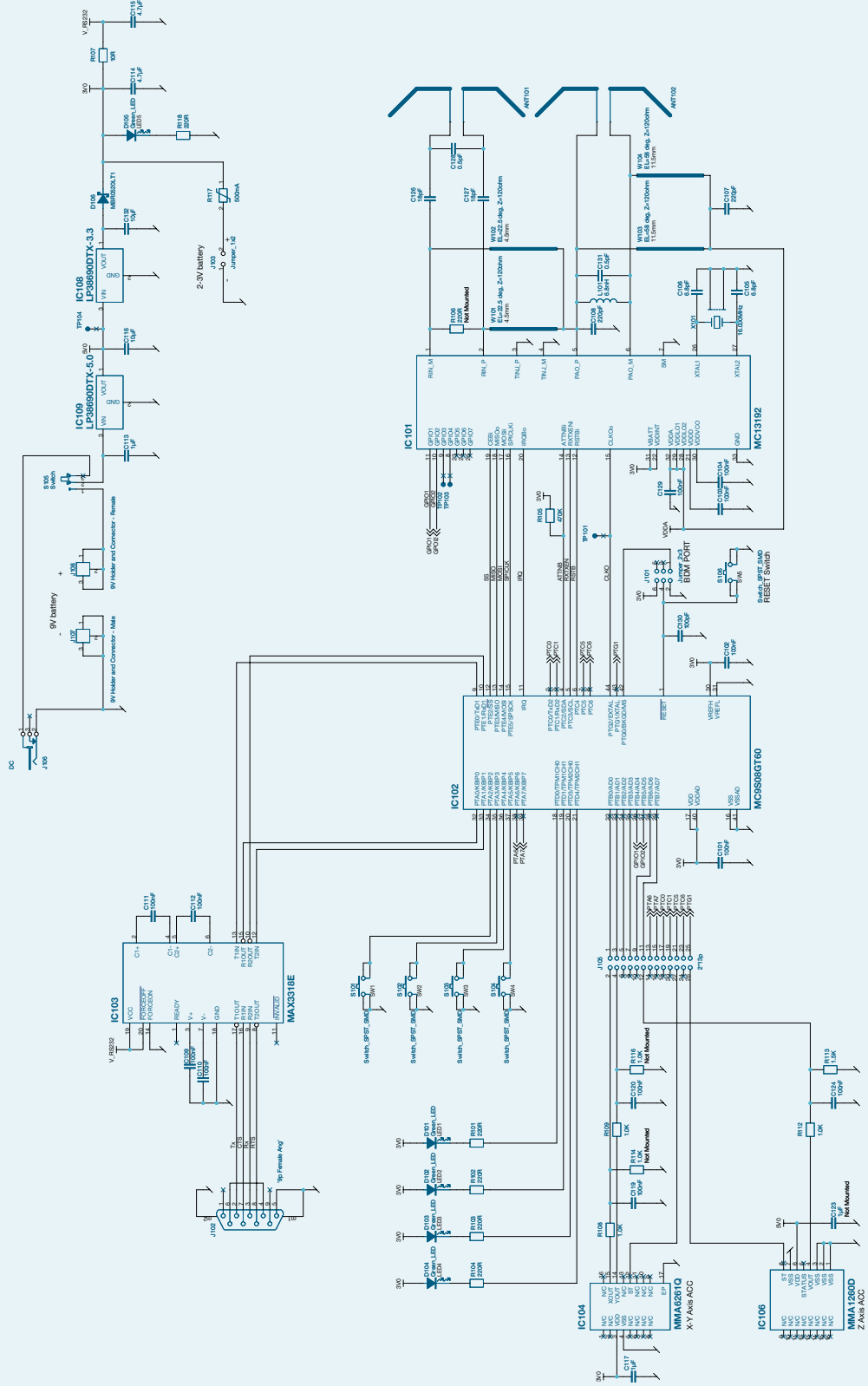
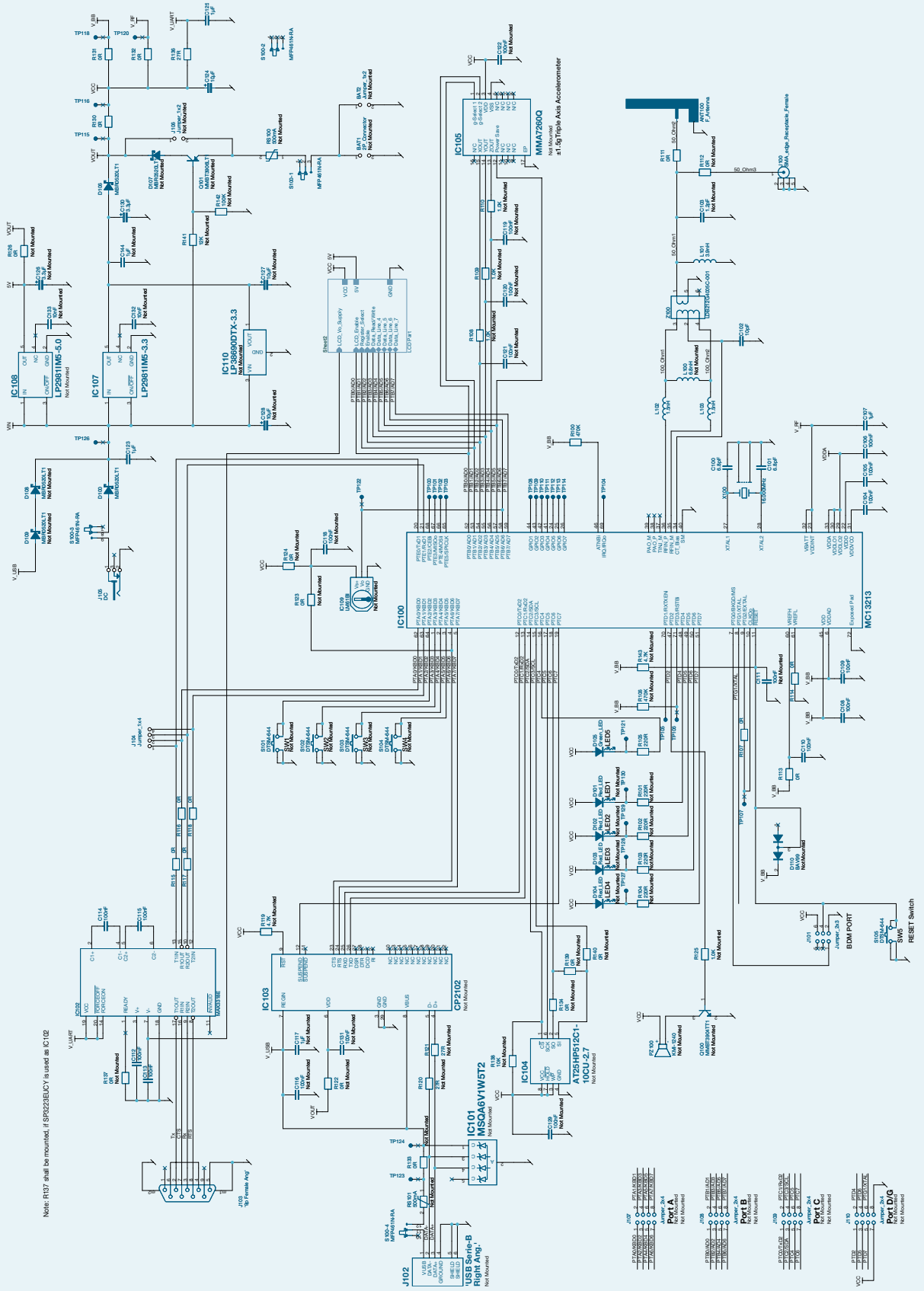


图5: 13213 SED参考设计原理图



天线设计是一个非常复杂的问题，特别是对那些RF设计经验不足的数字设计人员而言。通常情况下，设计人员在设计时会考虑以下诸多因素，如选择正确的天线、天线调试、匹配、增益/损失，以及了解所需的辐射模式等。我们建议通过收发器供应商提供的应用笔记了解天线的基本知识。但是，大部分数字工程师愿意考虑使用提供了天线设计的供应商方案。这样，他们就能集中精力进行应用程序的设计。应该寻找以完整的Gerber文件形式提供天线设计的天线解决方案，这样就可以直接交给印制电路板生产商去制板。提供这样的天线设计方案的供应商就消除了无线应用中与良好的天线设计、出色的范围和稳定的吞吐量等有关的问题。

对于低成本的无线网络子系统的收发器部分来说，方形扁平无引脚封装(QFN)是最佳的小体积封装解决方案。这种封装考虑了感应和控制方案经常面临的电路板空间限制问题。由于实施空间有限，在通常由电池供电的最终节点中，尺寸的大小就显得尤为重要。

微控制器

在选择感应和控制实施方案时有多种替代选项。有些设计人员选择单封装系统(SiP)或单封装平台™(PiP)，它们在单个封装中包含收发器和MCU功能。但是，如果您选择独立的收发器和微控制器结构，就拥有了选择各种不同MCU的灵活性，为不同的终端产品的结构进行组合和匹配。

当选择后一种实施方案时，要通过全面的研究来选择合适的MCU。MCU的选择取决于感应和控制应用的复杂度与合适的性能因素的匹配，包括内存结构和外围模块等。通常对于低成本无线感应系统来说，具有20 MHz CPU运行频率(10MHz总线时钟)范围的8位MCU就能提供符合这些应用的易于实现、成本低廉的替代方案。后台调试功能，以及在线调试过程中支持单个断点(标签和强制选择)设置(外加两个内置调试模块断点)的功能，可以提供理想的调试环境。许多MCU方案都可以支持多达32个中断/复位源。

感应和控制应用对内存的要求一般是8KB闪存加512B RAM，或低至4KB闪存加256B RAM。在整个工作电压和温度上的闪存读取、编程和擦除是必需的。

各种运行模式都能对功耗能够进行精确控制，这是扩展电池供电方案的一个重要功能。应该寻找支持正常操作

(运行)模式、内置后台调试激活模式、多种停止模式(总线和CPU时钟停止)及等待模式等各种选择的MCU。

考虑一下选用具有内部时钟源模块的微控制器，该模块应包含由内部或外部参考时钟控制的锁频环(FLL)，内部参考可精确调整，达到0.2%的解析度和2%的温度、电压偏移。内部时钟源模块应支持1 MHz~10 MHz的总线频率。MCU可为关键模块选择时钟输入，并控制其时钟以驱动模块功能。此外，应寻找具有低功率振荡器模块的MCU，可用软件在31.25 KHz~38.4 KHz或1 MHz~16 MHz频率范围之间选择晶振或陶瓷谐振器，并支持高达20MHz的外部时钟输入。

选择一个能提供系统保护功能的MCU也至关重要，诸如看门狗计算机正确操作(COP)复位选项，而且还可以选择以专用的1KHz内部时钟源或总线频率运行。其他必须具备的系统保护功能还包括可进行复位或中断的低电压检测，带复位的非法指令检测，带复位的非法地址检测和闪存块保护。

各种嵌入式外围设备有助于简化您应用的实现。建议采用8通道、10位模数(ADC)转换器来获得精确的连续逼近值。特别的功能还应包括自动比较、异步时钟源、温度传感器、内部带隙参考通道和可使用实时中断(RTI)计数器硬件触发的ADC。

感应和控制应用的其他基本外围设备包括：可选择与内部参考信号进行比较的模拟比较器模块(ACMP)、串行通信接口模块(SCI)、串行外设接口模块(SPI)、集成电路互连(I²C)总线模块、2通道定时器/脉宽调制器，可用于输入捕获、输出比较、带缓冲的边缘对齐PWM或带缓冲的中心对齐PWM；具有预分频器的8位模数定时器模块，以及可用软件选择边沿或边沿/电平模式极性的8引脚键盘中断模块。

现在有多种小型MCU封装选项能满足感应和控制设计的要求。这有助于优化有限的电路板空间，特别是在终端节点中和电池供电的应用中。部分满足这些要求的MCU封装包括少管脚数的塑料双列直插(PDIP)封装、方形扁平无引脚(QFN)封装、薄型收缩外形(TSSOP)封装、双排扁平无引脚(DFN)封装和窄体小外形(NB SOIC)封装。

在选择MCU时，周全的考虑应兼顾硬件和软件工具的易用性、文档的清晰度、参考设计和应用代码的可用性，

以及其他设计支持。同样，在RF或调制解调器方面的设计中，高效的集成开发环境(IDE)应包括采用图形化用户界面(GUI)的工具，它应内置各种功能和工具，可以简化编码和项目文件的管理，从而加快设计进程。专业工具会对硬件层进行抽象，并根据应用的要求针对特定的MCU提供最佳C代码，这样您就可以将精力集中在应用的概念上。同时也要考虑快速、简单的调试和闪存编程功能。此外，提供允许设计人员创建可重复使用的软件组件，并在不同的项目中重复使用的方法也相当有用。

感应、监控和控制子系统原理图

感应、监控和控制子系统的参考设计原理图经常用作应用的基础，并据此来提出具体的设计要求。作为采用单封装系统(SiP)方案的设计示例，下图给出了飞思卡尔1321x-SRB传感器参考电路板的原理图，它包括1321XDSK(开发人员入门工具套件)、1321xNSK(网络入门工具套件)和1321xEVK(ZigBee评估工具套件)。

1321x-SRB包括飞思卡尔MMA7260Q三轴加速传感器，与入门工具箱一起使用，能帮助您在几分钟内构建工作网络。电路板提供了参考应用设计，它是感应应用开发的起点(见图5)。

您还可以选择使用独立的收发器和MCU，以获得更多的设计灵活性。例如，下面的方框图是飞思卡尔13192传感器应用参考设计，用于感应、监控和控制子系统。我们在13192DSK-A0E(开发人员入门工具套件)和13192EVK-SFTE(评估工具套件)开发套件中提供了基于此设计的完整电路板(图4)。

您可以从网站下载上述参考设计的支持文档。下载文件夹包括原理图、物料单、印制板文件和实现整个参考设计所需的其他文档。

飞思卡尔用于感应、监控和控制应用的无线网络方案

我们为那些希望在最终产品中建立一个新范例的数字工程师提供了一套理想的解决方案。该电路板能够满足上面提到的实现一个低成本无线网络的基本要求。

示例产品包括：

- RF调制解调器或收发器解决方案(简化MAC、802.15.4 MAC和ZigBee)，可与各种MCU一起使用(MC1319x和MC1320x收发器系列与HCS08和ColdFire MCU系列一起使用)。
- 收发器集成方案(简化MAC、802.15.4 MAC和ZigBee)，采用MCU单封装系统(SiP)方案(MC1321x SiP系列)
- 模拟器件
- 传感器器件

为了实现简便易用的ZigBee和其他低成本、低功率无线网络，飞思卡尔为您提供大量参考设计、应用笔记、硬件开发工具和软件设计工具。如需了解更多产品信息，请登录我们的网站：www.freescale.com/zigbee。

Pattye Brown是飞思卡尔半导体的技术营销工程师和渠道营销联络人，在公司已经任职23年。她拥有工业工程学士学位和工商管理及市场营销硕士学位。

适用于成本敏感性应用的微控制器

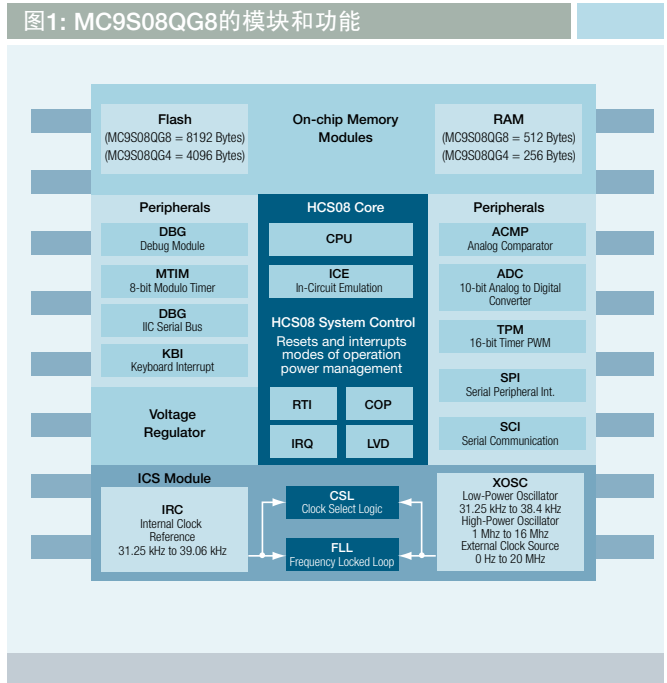
当您仔细观察我们的周围时，你会发现生活中电器的使用无处不在。洗衣机、冰箱、洗碗机、吸尘器、电动工具、室内加热控制系统、汽车座椅定位、动力方向盘和刹车、空调、吊扇、计算机等等，在工作场所包含电子器件的设备就更多了。

每台设备都包含电气驱动部分和辅助电子部分，它们以不同的效率消耗着能量。近来全球范围内对环境友好以及节能的关注让工程师们面临着不小的压力，他们要在设计中保持尽可能高的效率，同时还要得到低成本、简洁而且可靠的方案。目前有一些对能源效率具有正面影响的因素，但其中一些最有效的因素还是使用先进的电机和功率器件，开发出新颖的电机控制技术，并充分利用电子驱动装置特有的待机和唤醒状态。要成功地利用这些因素，还要达到多个国家的标准，必须采用智能微控制器来管理电机和辅助电子装置。

飞思卡尔在广受欢迎的多管脚HCS08微控制器系列基础之上，推出一款先进的HCS08少管脚器件，即MC9S08QG系列。

概览

MC9S08QGx (QG)微控制器具有丰富的功能，同时具有一系列小型封装，因而成为设计人员的理想选择，特别是在开发现代电子设备，同时还必须考虑低功耗、电路板空间和系统成本等问题时。QG系列的高性能体现在它具有低功耗率、可靠的闪存、更高的模拟精度、多种串行通信和增强的运动控制功能。请参见图1。



HCS08处理内核

这些微控制器是以先进的制作工艺生产的。因此，当电压高于2.1V时，它们的运行频率可高达20 MHz(10 MHz总线)；当电压低于2.1V时，其运行频率也可达16 MHz(8 MHz总线)，所以它在低电压时也能提供高性能。HCS08内核与HC08内核的指令集相同，但增加了对进入后台调试模式(BGND)的支持，因而可以充分利用集成的片上后台调试控制器(BDC)与在线仿真(ICE)的优势。

BDC含有一系列无干扰指令，可以在不中断CPU的情况下读/写片上内存。ICE包含强大的分类和跟踪单元，可以设置断点、触发选项并捕捉软件执行流程中的改变。

调试过程仅仅使用芯片的一个BKGD引脚，通过单线的通信接口就能执行。在完成调试过程后，用户应用程序可以将BKGD引脚当作额外的通用I/O脚使用。这样用户通过使用MC9S08QG的片上调试功能，省去了昂贵的仿真工具，同时也无需牺牲有效的管脚数，从而进一步降低开发总成本。

在电池供电的应用领域中，关键的要求是最大限度地减少微控制器的功耗，同时维持应用必需的处理能力。为了满足这些要求，在MC9S08QG器件中仔细地部署了专门的“等待”模式以及另外三个低功率“停止”模式(停止3、停止2和停止1)。这些低功率模式在下文中都称为“省电”模式。

时钟源电路

时钟源分配是所有系统设计中的关键部分。MCS08QG系列的内部时钟源(ICS)模块提供多种时钟源选择，可形成非常灵活又经济高效的方案，特别是对于少管脚数的器件而言。它由锁频环(FLL)、内部参考时钟(IRC)、外部振荡器(XOSC)和时钟选择逻辑(CSL)等模块组成。

FLL将参考频率提高到512倍，它只采用简单的数字逻辑技术，因而与PLL不同，它无需另外的外部元器件。

IRC是完全可调整的，既可用于FLL的参考时钟，也可直接用作CPU和总线时钟的时钟源。其调整精度典型值可达0.1%。调整频率范围为31.25 kHz到39.06 kHz。

XOSC参考时钟可配置成3种模式。第一种模式是低频振荡器模式，目的是与32 kHz~38.4 kHz的晶振体或谐振器一起使用。第二种是高频振荡器模式，可以与1 MHz~16 MHz晶振或谐振器连接。最后一种振荡器模式是特别定制的，可以与高达20 MHz的外部有效时钟一起使用。

CSL选择并划分时钟源，供CPU和其他微控制器模块使用。可以选择FLL、IRC或XOSC时钟源来驱动CPU和总线。输出频率可以在任何时间除以2、4或8，以降低CPU和外围设备的执行速度，从而即使不把微控制器切换到省电模式，也能延长电池的寿命。

片上内存

该系列提供两种闪存/RAM内存配置。设计人员既可以选择具有8 KB闪存、512B RAM的MC9S08QG8，也可以选择4 KB闪存、256B RAM的MC9S08QG4。

这一闪存技术基于飞思卡尔的第3代(0.25 μ)技术，在整个温度范围(-40°C至+85°C)上，电压降到1.8V时仍可编程。闪存可以逐字节编程，并可逐页擦写(512字节)——在室温下可以达到10次万次的闪存写/擦周期，在所有温度和电压工作条件下也能确保1万次周期。闪存数据保持期限设计为至少15年，根据系统情活的不同可以长

很多。闪存的分区分大小适当，又可超快编程，所以让用户能模拟EEPROM使用。

MC9S08QG含有专门设计的电路，可防止对闪存和RAM存储器的非授权访问，从而可防止潜在的应用代码盗用。这种实现机制具有很高的安全等级，一旦器件由这种机制加密，那么根据其设计，只有输入正确的8字节后门对比密钥后才能解锁。重要的非易失性数据和代码(如应用常数和引导程序)也可以选择性地加以保护，防止发生意外的写入和擦除动作。

高级计时器

MC9S08QG系列有两个独立的计时器模块。这些模块可用于定时、机械运动控制和/或软件任务调度。

第一个模块是双通道可编程16位定时器/PWM模块(TPM)。TPM的每个通道都能独立配置为输入捕获、输出比较和带缓冲的边沿对齐或中心对齐脉宽调制(PWM)。支持的两种PWM模式，特别是中心对齐模式，使系统具备了强大的运动控制功能。此外，定时器的寄存器的处理方式与16位的占空比更新操作是协调一致的，这样就避免了产生意外的脉冲宽度。

第二个定时器模块是8位模数定时器中断模块(MTIM)，它具有多个可能的输入时钟源和一个可编程的中断功能。定时器即可以自由运行，也可以作为模数定时器运行，可以在定时器溢出时产生各种周期的中断。

模拟电路

MC9S08QG系列集成了一个8通道、10位模数转换器(ADC)，能够为所有带模拟输出信号的传感器提供简单的接口。ADC可以在选定的通道上执行单次或连续转换。转换可由软件或实时中断(RTI)启动。ADC可以由总线时钟、总线时钟/2或ADC内部时钟源提供时钟。此外，ADC还可配置为将测量电压与给定的上限或下限值连续进行比较，并且分别在超过下限或上限值的瞬间产生中断。要记住的重要一点是，当选择ADC内部时钟作为ADC时钟源时，可以使用该功能将微控制器从省电模式唤醒。

该器件还包含模拟比较器(ACMP)模块，用来比较两个模拟输入电压或把一个输入电压与内部带隙参考进行比较。此模块将其输出引到某些其他片上外围设备上，或引到外部管脚上，这样其他片上外围设备或外部系统就能够对比较器输出信号进行连续监控和后处理。比较器中断可选择由比较器输出的上升沿或下降沿触发，或者二者都可触发。这类中断可以使微控制器退出省电模式。

该器件还带有一个温度传感器，其输出连接到一个保留的输入通道(AD26)上。其设计意图是不用消耗任何模拟输入通道，就能让用户测量硅芯的温度。此外，内部带隙参考电压和电源电压也能用同样的方法进行测量，从而可分别检查片上带隙电压稳压器和外部电源是否工作正常。

通讯模块

MC9S08QG8 (8脚和16脚)和MC9S08QG4(16脚)器件有3个通信模块，给设计人员提供了多种选择以满足其应用的需求。

串行通讯接口(SCI)提供标准的UART通讯，在微控制器和其他远程设备之间提供全双工异步和不归零(NRZ)串行通讯。

串行外设接口(SPI)可与其他微控制器或外围设备，如ZigBee收发器(MC13192)、步进电机驱动器(MC33976)、各种I/O扩展器和智能多媒体卡控制器等，建立高速主机(高达5Mbit)或从机(高达2.5Mbit)的SPI连接。

集成电路互连(I²C)提供了一种多个设备之间的通信方法。该接口设计为以最大总线负载运行时，速度可高达100kbps。它设计人员提供了一种简单的接口，可连接LCD控制器、RTC、DAC和其他外部EEPROM。

MC9S08QG的型号与封装选择

MC9S08QGx器件将飞思卡尔的HCS08产品系列扩展到少管脚数领域，并提升了低电压下的功能。最初有10种不同的器件。它们的不同在于片上闪存、RAM存储器的数量、管脚数、封装选项等。在所有类型的封装中，8 KB闪存和4KB闪存的器件都是管脚兼容的。表1列出了MC9S08Q系列中的各个器件。

开发工具

HC(S)08微控制器的CodeWarrior工具链完全支持MC9S08QG微控制器的高效、快速编程。对于16KB以下的代码容量，可提供免费的特别版FastTrack CodeWarrior工具，包括高效C编译器、调试器、连接器、汇编器、仿真器、集成开发环境(IDE)及其他组件，可帮助开发人员轻松、快速、毫无顾虑地完成应用开发。

每种嵌入式应用都需要进行初始化及使用片上外围设备。正是因为这个要求，飞思卡尔已经授权使用Processor Expert™自动代码生成器，并将其集成到CodeWarrior工具中。Processor Expert自动代码生成器极大地简化了应用开发，加快了开发速度。此外，它使得用户即使对片上外围设备的结构没有深入的了解，也能对它们进行编程。

为了能够在没有拿到最终应用硬件时就开始进行开发，飞思卡尔推出了DEMO9S08QG8评估模块。该模块通过集成的USB-BDM调试接口和各种片上外围设备编程示例软件，实现了简单、快速的应用开发。

典型应用

MC9S08QG系列现在正越来越广泛地应用于小家电、电动工具、智能中继、小型工业设备、汽车电子、灯光调节设备、手持工具、无线通信、计量器、热量计和报警应用中，这里列出的只是冰山一角。

应用实例包括使用三相直流无刷(BLDC)、直流有刷(DC)和通用(AC)等类型的电机。当采用MC9S08QGx微控制控制这些电机时，能够以高效实现高性能。

首先，图2显示的是成熟的BLDC驱动概念。它实际上是一个速度闭环的BLDC驱动，采用低成本的霍尔传感器作为位置和速度反馈。MC9S08QG4芯片在这类驱动中的优势在于能维持最高水平的元器件集成度，并达到运动控制外围设备的最佳使用状态。该驱动要求在外围硬件中采用“与非门逻辑”，以便能自动地将芯片的PWM输出扩展为三相功率平台的驱动能力。

BLDC驱动概念可应用于白色家电、高端泵机、风扇和其他需要以低成本实现高可靠性和效率的应用中。

其次，图3介绍了专用于电池供电驱动的控制机制。这个控制概念描述了如何将微控制器成功地集成到有刷直流

电机的控制应用中。驱动此类应用的微控制器应该能轻松实现多种高级的驱动控制技术，如无传感器闭环速度控制及转矩控制。

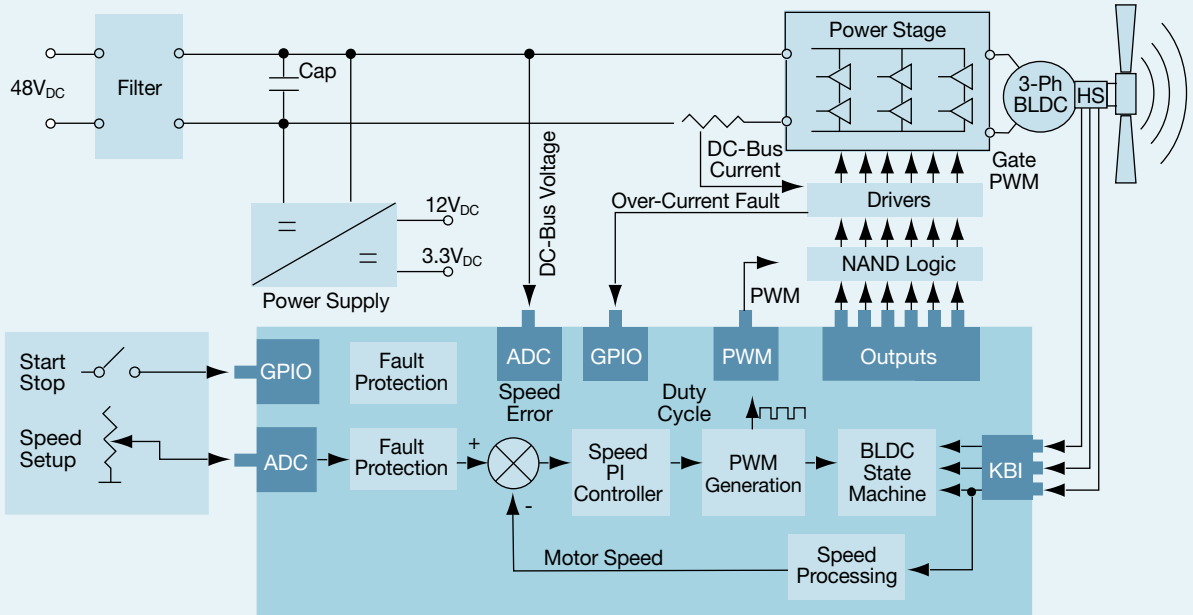
这个驱动概念可用于控制无绳改锥、电池供电钻头、搅拌机、电动牙刷、玩具车或者甚至高尔夫电动车等应用中的有刷直流电机。

最后，通用电机驱动的实现如图4所示。这个应用代表了基于MC9S08QG4微控制器的低成本、闭环通用电机断续控制驱动系统。PWM技术用于调整电机上施加的电压。PWM占空比周期的调制使加在电机上的平均电压值可以改变。与相位角驱动相比，断续驱动对输入电源进行整流。然而，它的优势在于效率更高、噪音更低而且EMC一致性也更好。

表1: MC9S08QG系列中的不同器件

功能	器件			
	MC9S08QG8		MC9S08QG4	
封装	16脚	8脚	16脚	8脚
闪存	8 KB		4 KB	
RAM	512B		256B	
XOSC	是	否	是	否
IRC	是		是	
ICS	是		是	
ADC	8通道	4通道	8通道	4通道
ACMP	是		是	
DBG和ICE	是		yes	否
IIC	是		是	
IRQ	是		是	
KBI	8脚	4脚	8脚	4脚
MTIM	是		是	
SCI	是	否	是	否
SPI	是	否	是	否
TPM	2通道	1通道	2通道	1通道
I/O脚	12 I/O 1个单输出， 1个单输入	4 I/O 1个单输出， 1个单输入	12 I/O 1个单输出， 1个单输入	4 I/O 1个单输出， 1个单输入
封装类型	16 PDIP 16 QFN 16 TSSOP	8DFN 8 SOIC	16 QFN 16 TOSSP	8 DFN 8 PDIP 8 SOIC

图2: MC9S08QG4 16脚QFN/TSSOP框图



通用电机概念可用于洗衣机、手持工具、真空吸尘器、洗碗机、单相和变速驱动中，还可用于低成本、中功率的可变电压输出应用。

现在，新的家用电器(IEC 60730)国际安全标准提倡使用独立时钟的看门狗定时器(COP)。这种定时器不仅在出现软件错误(软件堆栈损坏、程序计数器损坏、从中断或子程序返回出错以及死循环等)时，而且在时钟电路无法

图3: MC9S08QG4 8脚SOIC/DFN框图

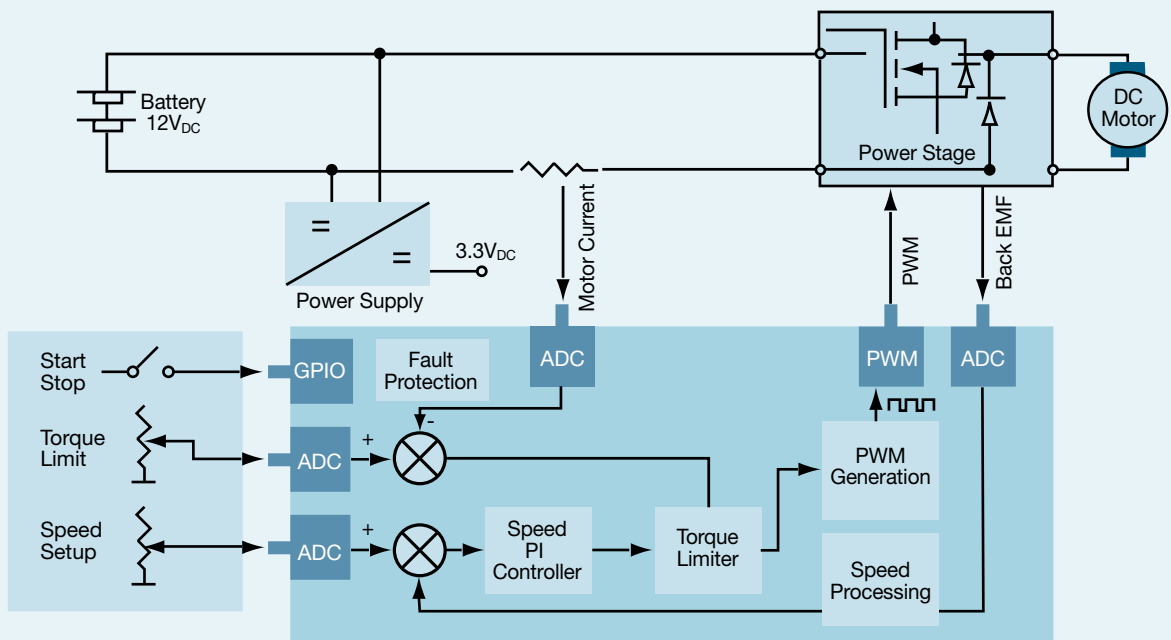
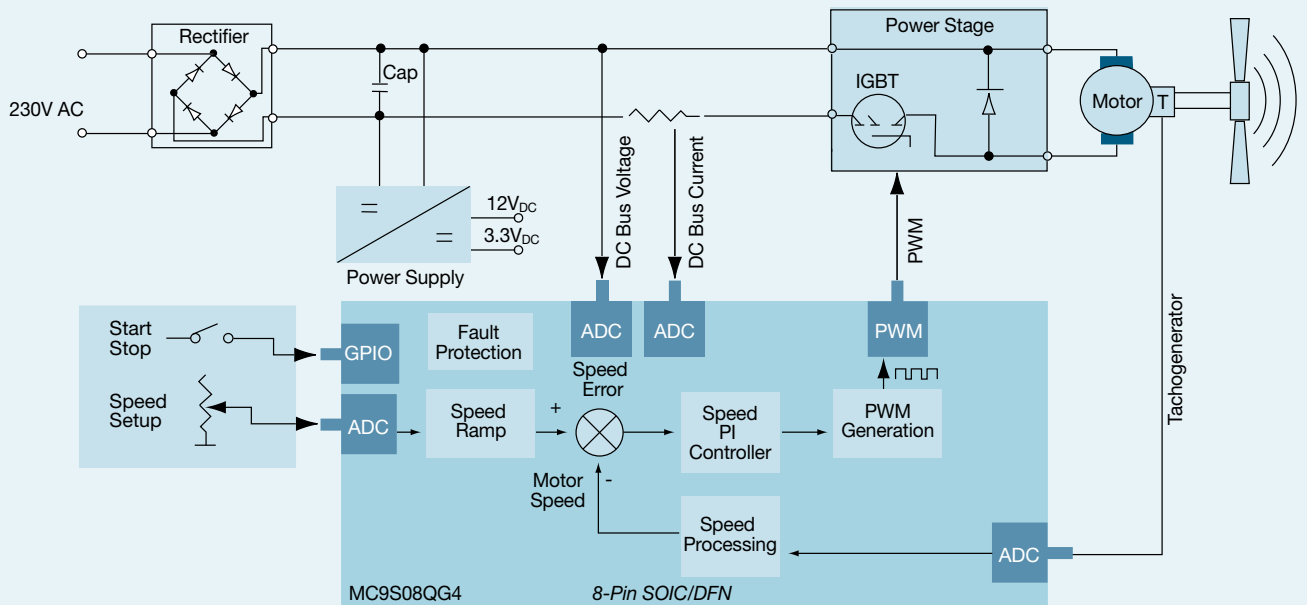


图4: MC9S08QG4 8脚SOIC/DFN框图



正常工作的情况下都能将系统置于确定的复位状态。这个安全功能在MC9S08QGx微控制器上得到完全实现，因而使它们可以根据安全性和新的国际标准应用于家用电器。

结论

最新推出的MC9S08QG微控制器系列，其处理内核、片上外围设备、节电功能和开发工具等，构成了能源效率敏感控制应用的理想解决方案。采用飞思卡尔的MC9S08QGx微控制器系列，不仅开发的费用可明显减少，最终生产阶段的成本也可大幅下降。在开发阶段，一些高级调试功能，包括CodeWarrior全功能工具链和Processor Expert工具包等都是免费赠送的。在生产阶段，内部时钟源模块、模拟电路和EEPROM模拟则减少了对诸如晶振或谐振器、模拟比较器、串行EEPROM这样的外部器件的需求，否则这些外部器件都是印制电路板上必不可少的。

飞思卡尔意识到，只有切实地对新技术进行投资，并且持续不断的以更低的价格开发出更智能、更高速的器件，我们才能维持在8位微控制器全球市场的领先地位(Gartner Dataquest, 2004)。因此，飞思卡尔的工程师一直在不断地扩展已得到广泛应用的HCS08微控制器系列。未来的产品系列有望扩展现有的少管脚和多管脚数的产品，进一步提高性能、增加片上外围设备的种类。

Martin Mienkina分别于1992和1997年获得Zilina大学工学硕士和电子工程博士学位。他曾在Trinec Steel & Iron Works Ltd公司工作过一段时间，负责设计与钢铁和轨道生产有关的现代控制系统。2000年，他加入飞思卡尔(捷克)系统中心，担任系统应用工程师，主要负责电机控制应用、驱动软件及数字信号控制器。Mienkina博士负责飞思卡尔在东欧地区的市场技术活动，工作的重点是新需求拓展和新产品开发。

用低端微控制器 构建简单的无线应用



无线通信是我们日常生活的一部分。在办公室、学校或家庭等场所，我们都在接触无线通信设备，如笔记本电脑、打印机、摄像头、手持设备和照明控制器等。这些设备的复杂程度与它们执行的任务类型有关。

其中许多无线应用采用小型微控制器和少量代码执行简单的任务。本文将讨论用于家庭自动化的无线应用。该应用

是无线控制器，设计为低成本的应用，只采用了三个电子元件：一个低端微控制器(MC9S08QD4)、一个RF收发器(MC13192)和一个2×16的LCD。其他需要的元器件为电阻和按键之类的无源器件。

图1是无线控制器框图。用来在收发器和微控制器之间交换数据的接口为串行外设接口(SPI)。该接口可以读写收发器的配置、状态和控制寄存器。SPI接口还可以读写位于收发器内部的RAM，用于通过RF发送和接收数据。

另一个用于收发器和微控制器间通信的信号是中断请求(IRQ)信号。IRQ脚由收发器进行处理。当收发器的状态寄存器发生变化时，IRQ脚会产生下降沿跳变。IRQ产生后，微控制器要做的第一件事就是读取状态寄存器，确定产生中断的特定事件。表1介绍的是微控制器管脚分配。

图1: 无线控制器框图

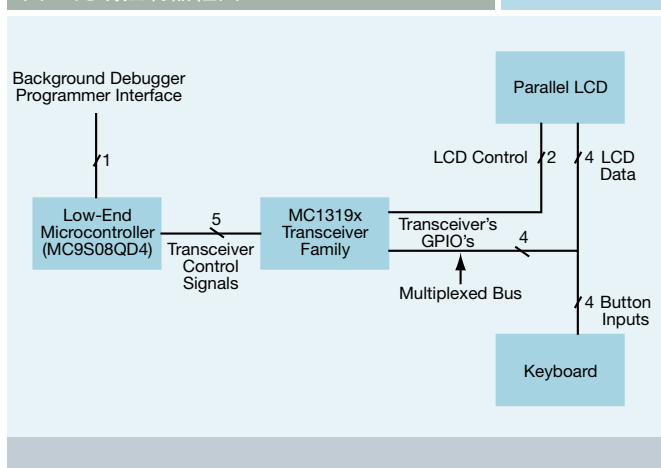


图2：微控制器和收发器的连接

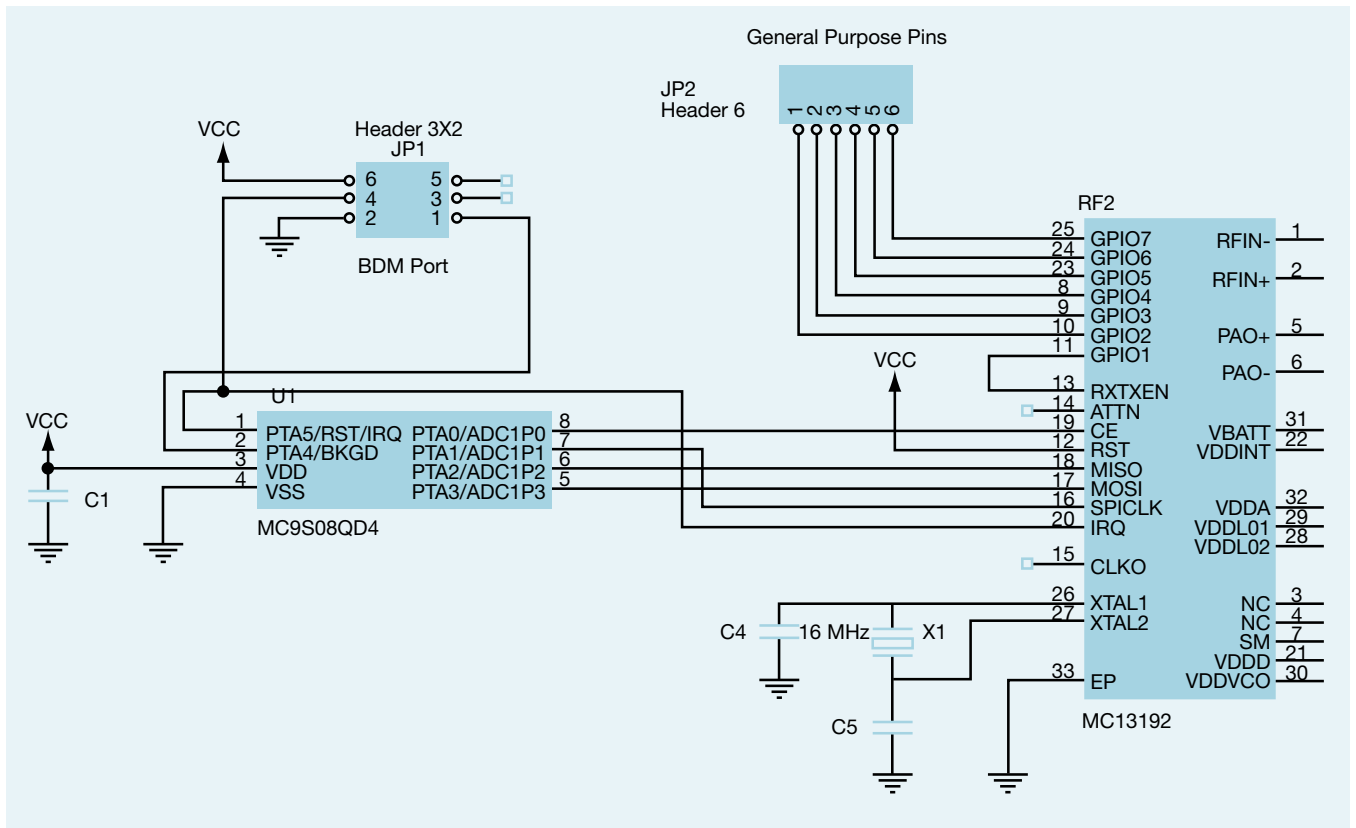


表1：引脚功能介绍

引脚编号	MCU 引脚名	类型	引脚编号	收发器引脚名	类型
1	PTA5/RST/IRQ	数字输入	20	IRQ	数字输出
2	PTA4/BKGD	编程和调试接口			
5	PTA3/KBI1P3	数字输出	17	Mosi	数字输入
6	PTA2/KBI1P2	数字输入	18	Miso	数字输出
7	PTA1/KBI1P1	数字输出	16	SPI时钟	数字时钟输入
8	PTA0/KBI1P0	数字输出	19	芯片使能	数字输入

表2：RXTXEN连接

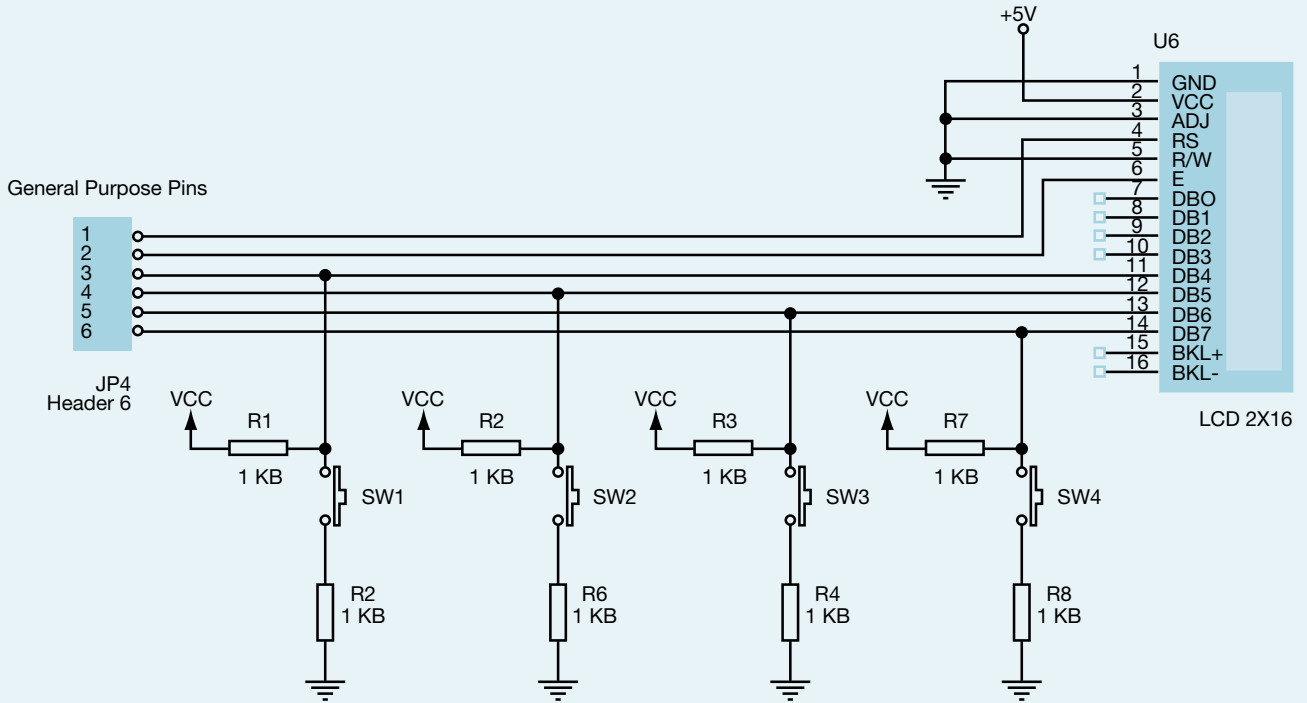
引脚编号	MCU 引脚名	类型	引脚编号	收发器引脚名	类型
11	GPIO1	数字输出	13	RXTXEN	数字输入

表1和表2显示了让收发器开始工作所需的最低硬件要求。参见图2的建议原理图。按照上述的方式连接硬件，可以配置收发器，并发送和接收数据。同时也可以通过SPI配置收发器提供的定时器和GPIO引脚，将其用于其它的目的。

如图2所示，收发器有7个GPIO引脚。其中一个引脚用来驱动RXTXEN信号，其余六个可用于用户的应用。在本推荐应用中，收发器提供的定时器和GPIO引脚用于驱动四个按键和一个2×16的LCD。

收发器要切换运行模式(接收、发送、半休眠、休眠或空闲)还需要另外一个信号。该信号称为RXTXEN。微控制器的所有通用引脚用来管理其他一些信号，如表1所示。RXTXEN信号由微控制器的GPIO1进行处理。表2描述了控制RXTXEN所需的连接。

图3：无线控制器的连接



这里推荐的无线控制器应用原理图分作两部分。第一部分包括开发各种应用所需的基本平台(见表1、表2和图2)。第二部分介绍无线控制器应用的连接(见图3)。

图3表明，该应用需要6条线路来驱动LCD，另外还需要4条线路来驱动4个按键。这一共有10条线路，而这10条线路必须由仅有的6个GPIO引脚来驱动。为了实现这一目的，LCD的4条数据线和4个按键必须多路复用。表3介绍了各个GPIO引脚的性能。

表3：应用引脚描述

收发器GPIO引脚	功能
GPIO2	RS
GPIO3	E
GPIO4	DB4/SW1
GPIO5	DB5/SW2
GPIO6	DB6/SW3
GPIO7	DB7/SW4

如表3所示，GPIO4、GPIO5、GPIO6和GPIO7用来驱动LCD的四条数据线以及四个按键。四个GPIO引脚必须配置成将数据送到LCD的输出，同时这些引脚还必须配置为读取按键电平的输入。

该应用使用LCD来显示不同的菜单，如当前温度、定时设置和设备超出范围等。

四个按键用来设置希望的温度、设置风扇定时时间、以及在菜单之间的导航。表4介绍了各个按键分配的功能。

表4：按键功能

按钮	功能
SW1	显示定时菜单
SW2	减少数值
SW3	增加数值
SW4	接受

当应用开始时，风扇总是处于关闭状态。必须按SW2或SW3按键才能打开风扇。风扇打开后，LCD上会显示当前温度。一旦风扇打开，可以分别使用Sw2和Sw3来降低或增加温度。最高温度可设置为60度。当风扇关闭时，SW1和SW4起作用。

可以设置定时时间来自动关闭风扇。定时时间在1~99分钟之间。如果没有设置定时时间，风扇永远不会自动关闭。

要设置希望的定时时间，步骤是：

- 打开风扇。如果风扇处于关闭状态，就不能设置定时时间。
- 按SW1。这样会在LCD上显示定时菜单。
- 按SW2或SW3来减少或增加希望的定时时间。
- 按SW4来接受新的定时时间。
- 一旦接受新的定时时间后，LCD上会再次显示当前温度。

该应用使用收发器提供的两个定时比较器，一个用来生成定时时间，另一个用来读取按键电平。

定时比较器1用来生成长度为1分钟的时间基准。该时间基准可生成1至99分钟的定时时间。

定时比较器2用于每100毫秒读取一次按键电平。为了读取按键电平，必须将GPIO4、GPIO5、GPIO6和GPIO7配置为输入。该应用将所需的GPIO引脚配置为输入，读取输入值，然后将这些GPIO引脚再配置为输出。

每次用SW2或SW3按键修改了温度后，系统会采用一个简单的协议将当前温度发送到加热-制冷系统。

结论

这里仅使用了一个8脚封装的微控制器，再加上很少的一些硬件资源，说明开发无线应用时并不一定总是需要高性能的微控制器。收发器需要有一些引脚，以便能用于从低端到高端的几乎任意微控制器上。最后，使用收发器提供的定时器和GPIO引脚能减少微控制器资源的使用，如GPIO引脚和定时器之类的模块。同时还必须记住的是，低端微控制器能够完成复杂的无线网络要求的一些简单任务，从而降低了整个解决方案的成本。

参考资料

www.freescale.com/files/rf_if/doc/ref_manual/MC13192RM.pdf

www.freescale.com/webapp/sps/site/overviewjsp?code=ZIGBEE_REFERENCEDESIGN_HOME

Juan Cazares是从事涉及SMAC协议栈的无线解决方案的应用工程师。他拥有电子工程学位，专攻控制和仪器领域。在加盟飞思卡尔前，Cazares曾先后担任过天然气和汽油测量设备的硬件及软件设计师。

共同创建解决方案的 飞思卡尔设计联盟计划

虽然高科技产品、市场和企业一直在不断发展，但是更快速地将最好的方案推向市场的目标始终保持不变。更好、更快、更便宜依然是当前的趋势。另一方面，面对起伏不定的经济形势，确定合理的规模并获得需要的技术及核心竞争力，并帮助您的客户适时推出产品，已成为比以往更重的负担。

飞思卡尔拥有全球最具竞争力的技术和产品，以及行业内最好的生态系统，其中不乏顶级的专家，掌握著最新的应用硬件和软件专业技术。但是，我们自己内部可能并不拥有所有您需要的技术和资源，可以让您的最终产品进入市场。

我们实施设计联盟计划(DAP)的目的是使飞思卡尔以及我们的方案系统可以帮助您实现目标。DAP拥有450多名第三方成员，他们在产品和应用的开发方面拥有不同的经验。从概念到最终产品，他们可以让您在任何阶段开始介入，帮助您的产品进入市场。我们还实施了工具联盟计划(TAP)，拥有约200名成员，他们为您或您的合作伙伴提供用于开发方案所需的最终硬件和软件工具。同时，大学联盟计划也已实施，以确保未来的设计团队可以跟上飞思卡尔工具和产品的发展潮流。

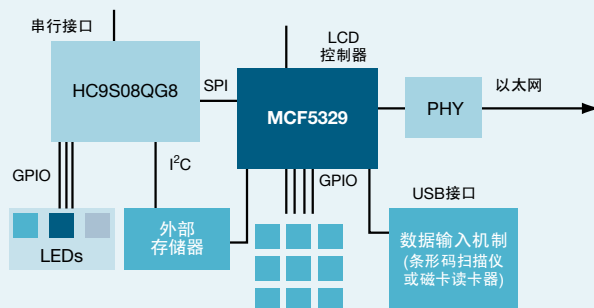
为了证实DAP成员的可信度，我们采取一系列步骤来验证列于目录中的飞思卡尔经验和应用。虽然在很多情况下该目录也包括推出新产品和创造新应用的工作，但是，列表内容只是飞思卡尔所有工作的一小部分。

许多DAP成员都是按纯粹的受雇设计结构，或不重做工程(NRE)费用结构操作，为客户提供完整的设计或生成知识产权(IP)。有的人创建并拥有知识产权，将其许可给他人使用或作为设计的一部分来改善设计流程。同时，借助他们丰富的应用知识和各种不同的市场知识，DAP成员可为每种设计提出不同的观点，从而最大限度地挖掘市场潜力。



Indesign LLC与飞思卡尔地区技术应用中心通力合作，使用32位ColdFire MCF5329嵌入式控制器和MC9S08QG 8位微控制器设计了这个POS参考设计(图1)。如需了解有关Indesign LLC的更多信息，请访问www.indesign-llc.com

图1



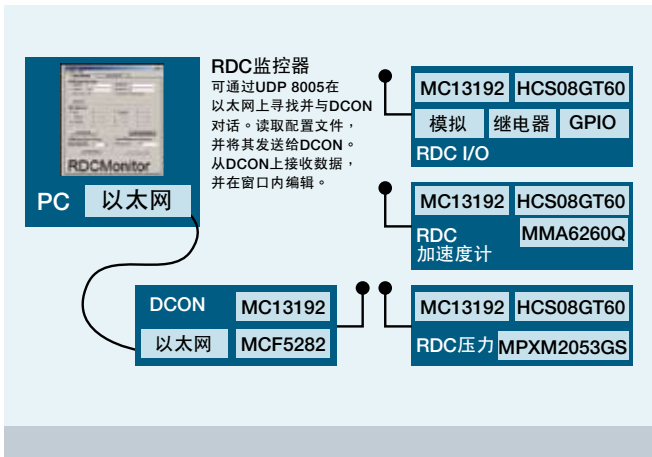
我们将DAP成员推荐给寻求服务或在设计产品过程中需要帮助的客户。我们也与DAP成员携手开发了许多自己的应用参考设计、演示板和评估板。部分实例包括：与Indesign LLC合作设计的POS参考设计、与FSI系统公司联合开发的ZigBee演示系统以及和i2systems联合开发的高亮度发光二极管(HBLED)演示设计。

除了这些例子之外，我们还有其他一些成员可以提供现成的ZigBee的平台电路板；基于Power Architecture技术、可用于网络、通信、工业和商业市场的极复杂电路板；手提电话；PLC控制器平台；电机控制；消费类参考设计等等。我们的DAP成员拥有必要的经验和能力，可以弥补您可能的不足。



FSI系统公司曾与飞思卡尔多次合作。这是自2005年以来我们一直展示给客户和开发人员的支持ZigBee系统的一个实例(图2)。如需了解有关FSI系统公司的更多信息，请访问www.fsisys.com

图2



我们的DAP成员能够了解我们的未来路线图和初步规范，使他们能够确保为您开发的解决方案是适合未来移植的最佳解决方案。

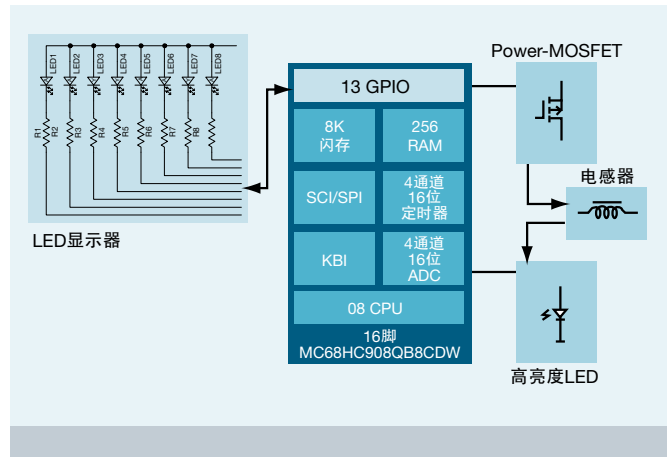
许多DAP成员参与了我们的“Alpha”硅计划，使他们有机会较早接触矽产品。这使得这些成员可以随时将最新的技术应用到您的产品中。飞思卡尔还可以从客户和DAP成员那里获得有关新产品的需求信息，从而使您拥有可能是最好的新技术。

您可以通过访问我们的成员目录，选择所需的服务，使用合适的过滤标准找到具有您需要的经验的DAP成员，网址是www.freescale.com/webapp/sps/site/dap.search.framework?NEXT_SCREEN=SEARCH。

您可以在我们网站www.freescale.com的搜索引擎中键入关键词或产品名称。在左侧菜单栏中，搜索结果可以再

i2 Systems设计了这个方便的HBLed演示(图3)，使飞思卡尔可以展示如何利用MC68HC908QY4微控制器来驱动和管理HBLed的功率控制。该设计是2006年7月在佛罗里达奥兰多召开的飞思卡尔技术论坛上推出的。如需了解有关i2 Systems的更多信息，请访问www.i2systems.com

图3



加上另外的筛选选项。选择“第三方支持”可以找到各个联盟成员的列表。通过左侧的菜单栏还可进一步筛选。

如需了解有关设计联盟计划的更多信息，请访问www.freescale.com/designalliance

飞思卡尔有超过53年的创新和技术，DAP成员也具有丰富的经验。当客户需要帮助去把产品更好、更快、更便宜地推向市场时，我们相信，作为一个团队，我们有能力实现这一切。

Gary Streber在面向客户的半导体技术销售和第三方关系方面拥有25年以上的经验。

使用8位MCU的BLDC电机控制

经济高效的BLDC电机无传感器控制



最近，一系列日益先进的变速驱动器已设计出来，用于提高产品的性能和系统的效率。尤其值得一提的是，过去主要使用直流(DC)电机的应用现在都设计成了使用直流无刷(BLDC)电机。直流电机在变速驱动中的普及得益于其简单的速度控制电路。但另一方面，由于有电刷，直流电机的可靠性较低，因为电刷容易磨损，需要维护或替换。使用BLDC电机就可弥补直流电机的这个缺点。

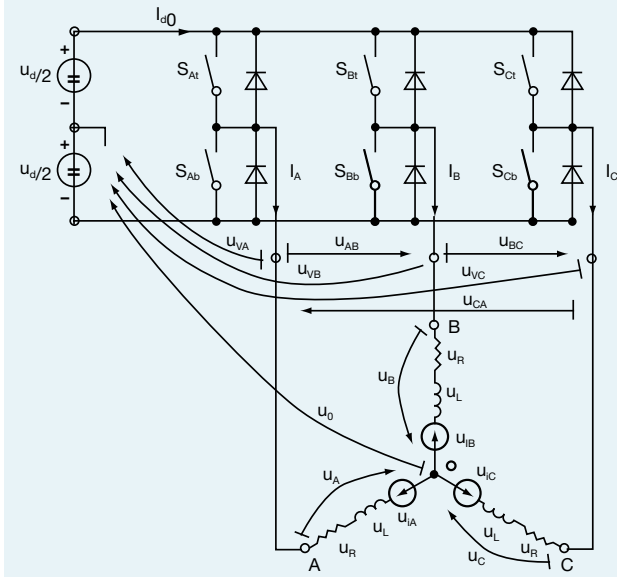
BLDC电机也称为电子换向电机。它的转子上没有电刷，换向以电子方式在某个转子位置上完成。定子的磁场回路通常由磁性薄板制成。定子相绕组插入到磁槽中(分布式绕组)，或在磁极上绕成线圈。转子永磁体的磁场方向及其转换是依据反电动势(由于转子的运动而感应到定子绕组上的电压)呈梯形的方法来确定的。这使得具有矩形形状的三相电压系统，可用于创建低转矩纹波的旋转磁场。在这方

面，BLDC电机与反向直流换向电机一样，它们的磁场旋转，而导体保持静止。在直流换向电机中，电流极性是由换向器和电刷转换的。在直流无刷电机中，极性由半导体开关转换，而切换必须与转子位置保持同步。省去换向器可实现更高的可靠性，也消除了限制直流电机最大转速的因素。因此，BLDC电机适用于需要高转速的应用中。

用BLDC电机来替代直流电机对控制算法和控制电路提出了更高的要求。首先，BLDC电机通常是三相系统。因此它必须由三相电源供电。第二，在某些特定的角度上必须知道转子的位置，这样才能将施加的电压与反电动势对齐。反电动势与换向动作的对齐非常重要。只有这样，电机才能像直流电机一样工作，并在最佳的工作点上运行。这些缺点可以被BLDC电机优异的性能和可靠性、以及功率器件和控制电路不断下降的价格所抵消。

BLDC电机的简单驱动模型(图1)由三相功率平台和一个直流无刷电机构成。系统的电源由一个电压源(U_d)提供。六个半导体开关(SA/B/C t/b)可产生矩形电压。半导体开关和二极管可以看作是理想器件。BLDC电机的梯形控制基于在任一时刻只有两相是接通电压的。这意味着只有一个上臂半导体开关和一个下臂半导体开关打开，且处于不同的相位上。所以最后一个相的上臂和下臂开关都是关断的，且断开了与电源的连接。考虑到所有可能性，我们有六种可能的组合来给三相系统中的两相接通电压。因此这种技术也叫作六步控制。

图1：BLDC电机 — 驱动模式



控制BLDC电机的最常见方法是使用霍尔传感器来确定转子的位置。控制系统检测转子的位置并将合适的电压施加到电机上。但是，至少有两个原因使人们更愿意去掉位置传感器。首先，霍尔传感器需要在BLDC电机与控制系统之间建立额外的连接(通常为五条线)。第二，去掉传感器有助于降低系统总成本。

使用图1可推算出系统的数学模型。对数学模型的分析表明了一种控制无传感器BLDC电机的可能的方法。这种方法称作反电动势过零法，它是将不馈电相感应的反电势电压与直流电总线电压的一半进行比较。过零动作发生在换向阶段的中间点。这种假设只有当两个半导体开关都开通，并且在用于检测反电动势电压的无馈电相上，没有电流流过时才成立。

基于反电动势过零算法的BLDC电机驱动，可以用各种不同的方法实现。最简单的方法是使用专门的模拟电路。这种实现非常容易、快速。但是这种实现受硬件的束缚，不能根据客户的需要进行定制。另一种方法是使用简单的8位微控制器(MCU)，结合使用检测过零信号的外部电路。

市场上也有带过零检测专用外围设备的8位MCU。另一种方法是使用功能足够强大的数字信号控制器(DSC)来估算过零信号。飞思卡尔已经发表了几个应用笔记和设计参考手册，详细解释了反电动势过零算法的实现— AN1913、DRM070和DRM028。AN1913和DRM077描述了使用

56F80x和56F8013 DSC的实现方法。后者介绍了采用8位MC68HC908MR32 MCU和外部比较器来估算过零信号的实现方法。这篇文章还提到了如何使用先进的8位MC9S08AW60 MCU来实施反电动势过零算法。

MC9S08AWx系列器件基于高性能的HCS08内核，可支持5伏的应用。它是一个高集成度高性能产品系列，具有多种宝贵的功能，如16到60KB的闪存、1到2KB的RAM、无需任何外部器件的灵活的内部时钟发生器、低电压检测、模数转换器(ADC)、串行通信模块以及其他等等。9S08AWx系列的设计使得它可以在各种环境下可靠地运行，具有很高的电磁兼容性(EMC)性能，并符合汽车应用的质量要求。该系列广泛适用于家电(白色家电)、工业和汽车等应用。

MC9S08AWx系列的主要关键特性包括：

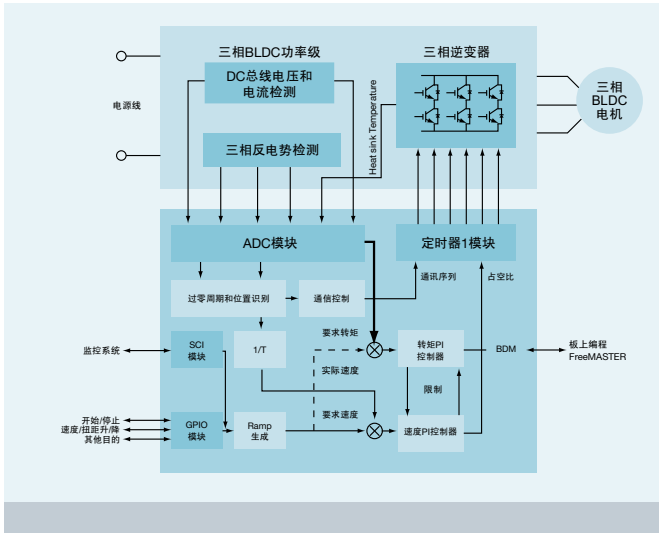
- 8位HCS08中央处理器(CPU)
- 20 MHz的内部总线频率
- 16到60KB的片上可在线编程闪存，具有块保护和加密选择
- 1到2KB的片上RAM

用于电机控制的主要外围设备包括：

- 具有自动比较功能的16通道10位ADC
- 1个2通道和1个6通道的16位定时器/脉冲宽度调制器(TPM)，每个通道都可选择输入捕捉、输出比较或边缘对齐的PWM功能。每个定时器模块的所有通道都可配置为带缓冲的、中心对齐的PWM
- 2个SCI, 1个SPI, 1个I²C，可用于与监控系统的通讯

此外，MC9S08AWx系列是第一个提供6通道定时器/PWM模块的HS08产品系列，这对完全的三相交流感应式和BLDC电机的控制是必需的。这个具有20MHz总线频率的高性能内核可无需外部比较器，实现完全的数字化无传感器BLDC电机的控制。

图2：应用框图



下面所描述的应用中包含以下BLDC电机驱动的功能：

- 无传感反电动势过零算法的实现
- MC9S08AW60 MCU的控制
- 用ADC检测反电势
- 完全的四象限运行
- 双向旋转
- 带有PI控制器的速度环
- 带有PI控制器的转矩环
- 速度范围：100-1200 rpm(取决于电机)
- 手动接口
- FreeMASTER接口

图2是应用框图。BLDC电机驱动包含三相功率平台、三相BLDC电机和MC9S08AW60控制板。所有测量信号都在三相功率平台上调节到ADC范围内。有六个测量数据：DC总线电压、DC总线电流、三相反电势电压以及散热器的温度(如果可以获得的话)。

BLDC电机驱动可通过开始/停止开关和向上/向下按键来进行手动控制。开关在开始位置上，驱动操作就启动了，可以两种工作模式运行：速度模式和转矩模式。在速度模式中，您可以通过向上/向下按钮来设定所需的电机速度。所需的速度将与实际电机速度相比较。速度的差值输入到速度PI控制器中，它将保持希望的电机速度，无论电机负载如何。如果电机达到最大允许转矩，驱动就自动进入转矩模式，维持住电机的速度，保证不超过最大转矩的限制。

在转矩运行模式中，驱动以相似的方式运行。所需的转矩(DC总线电流限制)通过向上/向下按键来设定，转矩PI控制器可维持希望的电机转矩。如果电机达到最大速度，驱动就开始维持最大的电机转速，速度和转矩PI控制器都独立于希望运行模式而运行。两个PI控制器的输出要进行比较，较小者施加于电机。这就自动确保了速度和转矩运行模式之间的平滑过渡。为了避免PI控制器积分部分的饱和，它们可相互限制。如果速度控制器在运行，则转矩PI控制器的积分部分就由速度控制器的积分部分来限制，反之亦然。

对于远程控制，提供了使用PC的FreeMASTER接口，或者驱动器可通过SCI由主系统控制。

PWM调制由一个6通道定时器/脉冲宽度调制器(PWM)模块完成。根据施加到电机上的电压的不同，或相对的半导体开关的通断方式的不同，调制技术也有所不同。如果电机在一个PWM周期中可以看到DC总线和零点电压，我们就将这种调制技术称为单极性调制。如果电机在一个PWM周期中可以看到DC总线电压的两极，我们就将这种调制技术称为双极性调制。

第二个区别是相对的半导体开关的通断方式。如果它们独立进行通断操作，我们称之为独立开关。如果以互补的方式进行通断操作，我们就称之为互补式开关。调制技术的正确选择对驱动特性有很大的影响。由于TPM/PWM模块可以支持任何类型的PWM调制技术，应用设计师可以自由地选择适合应用的调制技术。

另外，不管是边缘和还是中心对齐的PWM模式，都可以支持各种不同的调制技术。在如何选择正确的调制技术上有两种基本的观点。第一是驱动的运行象限的数量。如果需要完整的四象限运行，就必须选择互补式开关。如果是两象限运行，独立开关就够了。必须考虑的第二个方面是正确的反电势检测。正确的反电势检测的要求之一是在两个半导体开关都开通时来检测反电势电压。

由于MC9S08AWx系列不具有PWM到ADC同步机制，所以需要另外的方法来确保正确的反电动势检测。一个简单的方法是采用TPM/PWM溢出中断。在中心对齐的PWM模式中，TPM/PWM溢出发生在脉冲的正中间。当选择双极性互补式开关时，半导体开关的占空比总是等于或大于50%。采用不同的换向表就可以得到顺时针或逆时针旋转方向。想一想中心对齐的双极性互补式PWM，可以看到正确的反电动势检测可以在驱动的任何运行点上得到保证。而且完整象限运行的要求也可以得到满足。

软件结构包括两个周期性中断(ISR)、2个事件中断和1个后台运行环(见图3)。这些周期性中断服务子程序(ISR)是一个TPM/PWM溢出ISR和一个5ms定时比较ISR。第一个事件ISR由SCI通讯驱动，而第二个(IRQ ISR)处理过流事件。控制算法的时间基准从TPM/PWM溢出ISR获取，每62.5μs(16 kHz PWM)调用一次。每次调用此ISR都要进行ADC转换、PWM刷新和过零检测。每个PWM周期进行两次转换。

每个PWM周期的第一次转换都各不相同，它们在DC总线电压、DC总线电流和散热器温度之间共享。第二次转换始终是

不馈电相的反电动势电压。较慢的控制环每5毫秒执行一次。ISR执行速度和转矩测量/滤波、手动接口的处理、转矩和速度PI控制器的计算，以及静止检测。

后台运行环执行应用状态机(ASM)，包括初始化、停止、对齐、启动、运行和错误状态。MCU复位后，ASM从初始化状态进入停止状态。一旦用户将开始/停止开关转到开始位置，ASM就进入对齐状态，将电机与已知位置对齐。经过一段预定的时间，ASM继续运行，进入启动状态。在启动状态下要执行六次换向而不需要任何反馈。换向之间的定时是由启动表设定的。在启动状态之后，ASM进入运行状态，此时引入过零并进行闭环控制。在运行状态下可以接受新的速度和转矩要求，并由PI控制器进行维持。如果检测到转子静止，ASM就再次进入对齐状态，试图执行一次新的启动顺序。如果出现故障，ASM就进入错误状态，然后进入停止状态。

除了一些数学函数采用汇编进行优化外，整个软件都是用C语言编写的。代码的大小为4.5KB，所使用的RAM不到300字节。现在的MCU负载不到40%，显示了MC9S08AWx系列的能力。该应用的详细描述在飞思卡尔网站上以设计参考手册介绍(www.freescale.com/files/microcontrollers/doc/ref_manual/DRM086.pdf)。

图3：主控制中断子程序



Pavel Grasblum作为系统应用工程师在飞思卡尔工作超过七年。他在2001年获得了电力电子和电气驱动专业的博士学位。他采用飞思卡尔的多种产品(8/16位MCU和DSC)支持电机控制和电源转换的应用。

驱动LCD的窍门

使用双MC9S08LC60芯片增加显示段数



您正在寻找高达320段的液晶显示(LCD)驱动方案吗？本文将为您提供使用两个飞思卡尔8位MC9S08LC60 (LC60)微控制器(MCU)的方法，并介绍如何进行硬件连接，以便充分利用其LCD控制功能，以及两个MCU之间的同步。

简介

LC60是高效益、高性能S08系列产品中的一员。这个器件具有一套丰富的外围设备和LCD模块，能够驱动高达160段的显示。12位ADC提供高精度的模数转换。此外，闪存配置在两个独立的阵列中，使得当应用程序正在运行时，部分闪存还可以用作EEPROM。所有这些功能以极低的功耗运行，这是S08架构的特性，使它对于电池供电及手持设备的应用极具吸引力。这种8位MCU的集成度非常高，非常适合温控器、计算器、数字万用表、医疗监控设备、玩具和电器等应用。功能强大的外围设备和LCD模块有助于减轻内核的压力。参见图1提供的结构图可以看到其他硬件模块是如何工作的。

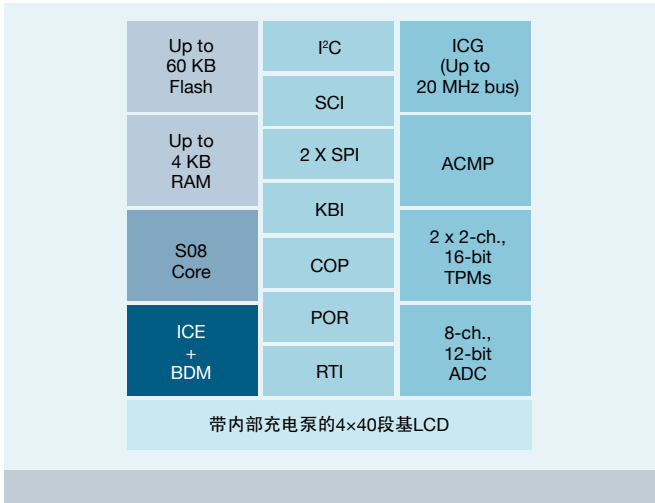
LCD的特点

与MC9S08LC60兼容的TN型LCD玻板应当符合以下规格：

- LCD运行电压—LCD玻板运行电压的典型值是3或5V。此运行电压是加在一个段上的名义门槛电压的电平值。
- 电流消耗—与所有段的面积有关
- 视域—与LCD玻板的大小有关
- 驱动模式—占空比(1/2、1/3和1/4)和偏置(1/3)
- 视角—3点钟、6点钟、9点钟和12点钟
- 显示模式—反射、透射、半透射半反射
- 运行温度—各种运行温度

要让LC60 MCU驱动160个段，可以用40个前面端和4个背板端组成一个矩阵结构，前面端为行，背板端为列。LC60 MCU为前面端和背板端提供四个级别的电压，产生所有必需的波形，驱动各个段。产生这些电压不需要外部稳压器。在图2所示的例子中，采用了 $\frac{1}{2}$ 占空比和 $\frac{1}{2}$ 偏置率(三种级别的电压)。当寄存器LCDRAM [0..20] 载入新的数值时，数据就被载入进行LCD显示。点亮或关闭一个段就是对某个位进行置位或清零。欲知详细信息，请阅读AN3280“LCD到MC9S08LC60的接口”。

图1：MC9S08LC60结构框图

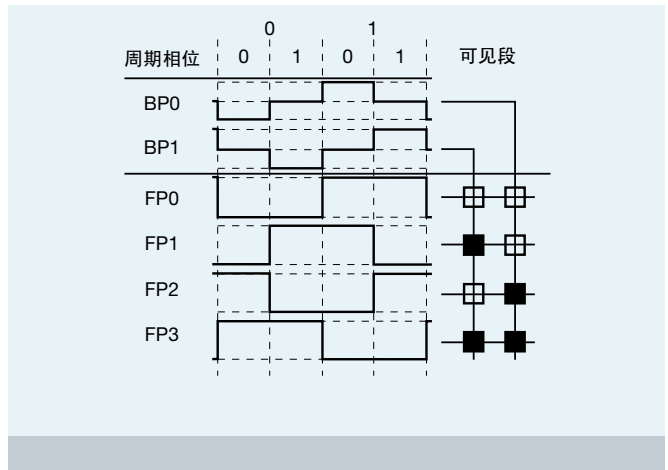


一片LC60 MCU可以驱动高达160段的LCD显示，因此两片LC60可以用于驱动320段LCD。将两片MC9S08LC60连接到一个LCD上时，有一些问题需要考虑，例如：

- 背板端控制
两片LC60 MCU中应只有一片驱动四个背板端，其信号直接连接到显示器的管脚。
- 前面端控制
每片LC60 MCU可控制40个前面端。它们一起能驱动80个前面端。因此，它们必须在每次显示需要更新时交换命令和数据。建议的接口是I²C，一个LC60 MCU用作主机，另一个用作从机。
- 时钟同步
每片MCU会控制一半的LCD前面端。但是，只有主MCU产生连接到LCD背板端的信号。对每个要点亮的段，需要在背板端和前面端之间有特定的组合。这样，由于前面端和背板端配置为一个矩阵结构，所以前面端或背板端信号的任何变化都需要主MCU和从MCU同步进行更新。因此，微控制器必须同时更新LCDRAM、开启段，并采用同一时钟进行同步。

图3是采用两个MCU和320段LCD显示的系统框图。

图2：1/2占空比和1/2偏置率的LCD波形



LCD初始化模块

图4中的流程图介绍了主机方面LCD初始化建议程序，该程序能够确保两片LC60 MCU同时开通它们所控制的前面端。当I²C接口配置好之后，主机控制的LCD必须完成初始化，但不能启动。只有在通过I²C确认，从机已成功接收到所有传送的数据，并准备好启动它控制的LCD之后，它才能启动。一个键盘中断(KBI)信号发送到从机使两个LCD同时启动。这个中断指的就是流程图中的“initOK”命令。

在从机方面(见图5)，应当首先初始化它的LCD模块，通过I²C接收所有将加载到LCD玻板上的数据，并保存在RAM中，然后通知主机传送完成。检测到initOK后(信号由主机发出)，再启动LCD。

图3：系统框图

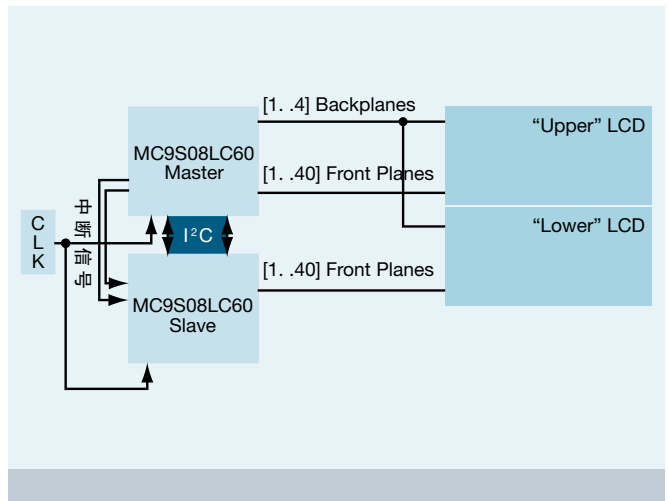


图4和5：LCD初始化模块

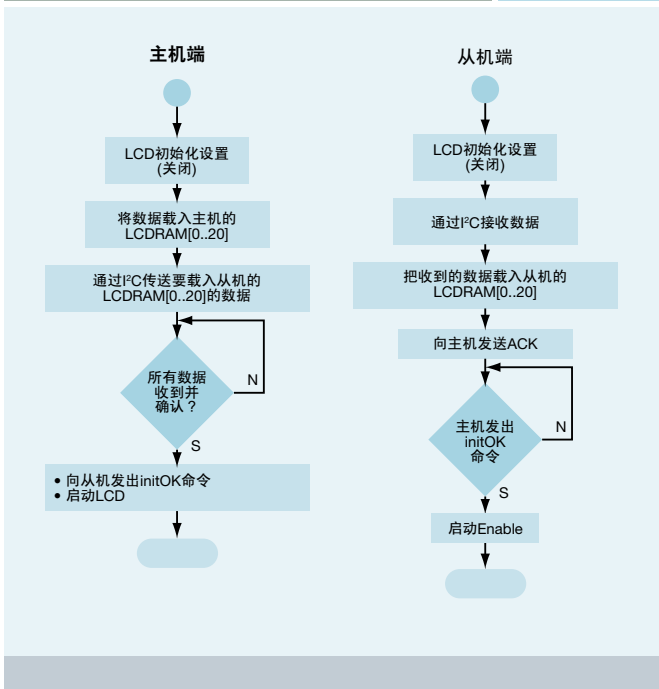
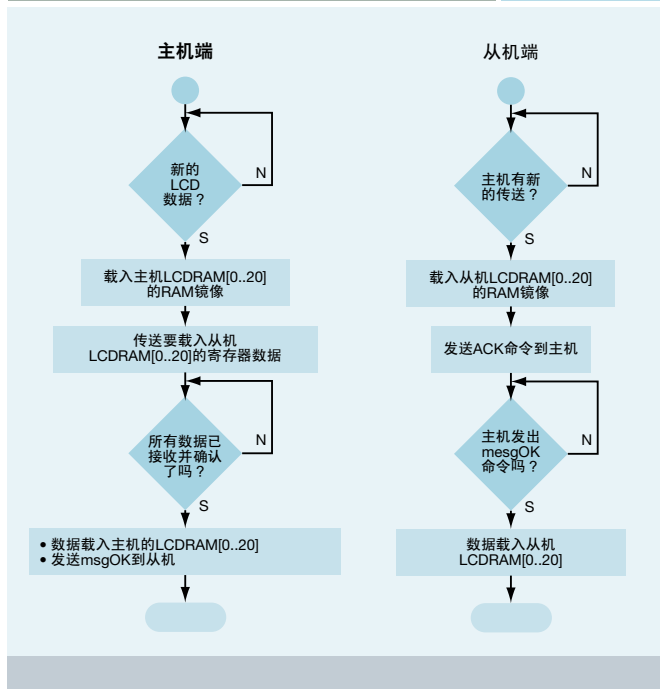


图6和7：LCD数据更新



LCD数据更新

当显示需要更新时，写入RAM寄存器的信号在主机和从机间不会同步，因为从机必须等待主机传送的数据才能进行显示更新。我们建议采用镜像，即主机和从机的RAM中的保留空间，它们可以分别接收和保存主机所控制的前面板的LCD新数据，以及从机所控制的前面板的新数据。这些镜像应该与LCDRAM寄存器的大小相同。当两个LCD中的数据要更新时，所有数据都必须传送到两个LC60 MCU的镜像中，镜像会保存新的内容，一直到下次同步。

发出I²C信号之后，在主机端和从机端，镜像内容会复制到LCDRAM中。图6和7给出了流程图。

主机必须将镜像内容传送到“低端”LCD — 从机控制的40个前面端 — 并等待传送被确认。

另一个键盘中断信号，本文称之为msgOK，也必须发送给从机。从机只有在检测到该信号后，才镜像内容传送到它的LCDRAM寄存器。

总结

S08系列中的高级产品，MC9S08LC60为飞思卡尔的众多8位微控制器系列产品提示了价值。它能够高效益地、轻松地通过I²C实现多个MCU的LCD显示驱动。

参考资料

- 所有资料可以在www.freescale.com.网站上下载
- Daniel Malik, “XGATE Library: TN/STN LCD Driver”, Freescale Application Note, AN3219.
- Steven Torres, “Interfacing a LCD to the MC9S08LC60”, Freescale Application Note, AN3280.
- Steven Torres, “Software to Accompany AN3219 Interfacing a LCD to the MC9S08LC60”. Freescale Application Note, AN3280SW1.

本产品整合了SST许可的SuperFlash技术。

Ronald Gonzalez作为开发人员曾在飞利浦工作了11年，1999年后成为飞思卡尔拉美区FAE，为该地区的汽车和家电客户提供支持。

Tatiana Orofino是计算机工程师，2005年以来作为飞思卡尔拉美区FAE，为该地区的全球客户、以及ZigBee®技术和ColdFire®产品提供支持。

SMAC

轻松构建无线连接

综述

嵌入式工程师明白低端无线传感器网络市场在不同应用中正在快速发展，例如控制、监控、通讯和数据采集。我们可能认为只要有一个能达成网络目标的无线传感器网络协议栈就足够了——遗憾的是，并不完全是那么回事。我们生活在一个竞争激烈的市场，客户在选择时，几分钱之差就有可能完全不同。因此，飞思卡尔开发了并维护着一个免费的协议栈软件，它适用于飞思卡尔的产品，可以和介质访问控制(MAC)和ZigBee协议共存。该协议栈称为简单MAC或SMAC。

SMAC是一种不复杂的软件协议，基于IEEE 802.15.4协议，能够在采用8位微控制器(MCU)控制的飞思卡尔收发器上得到实现。其目的是用于快速产品开发和系统评估。SMAC实现的不是全部ZigBee协议栈，也不是完整的802.15.4层，它是一种简单易用的协议。只需要基本功能的低成本应用，例如传送、接收以及电源和通道选择等就是SMAC的最佳用武之地。

功能和特徵

SMAC能够用于飞思卡尔收发器和单芯片系统，如MC1319x、MC1320x和MC1321x的最低成本的解决方案，它支持星型和对等网络，但也可以开发更加复杂的方法，创建您自己的网络层或增加转发器节点。SMAC集成到BeeKit软件之中，使开发时间快速而准确。它完全采用ANSI C编程，针对的是HCS08内核。它包含源代码，与BeeKit软件一起提供，无需增加费用。

便携性和小体积

SMAC小得可以达到4~8 KB。SMAC的大小取决于所有要增加的额外组件。小体积的另一个巨大优点是您可以通过特别版CodeWarrior工具套件来使用它。特别版免费提供，可以编译多达32个代码文件。它的代码采用C语言编写，因此能够很轻松地移植到其他MCU系列和技术上，例如16和32位处理器以及数字信号控制器(DSC)。

图1：SMAC数据包结构

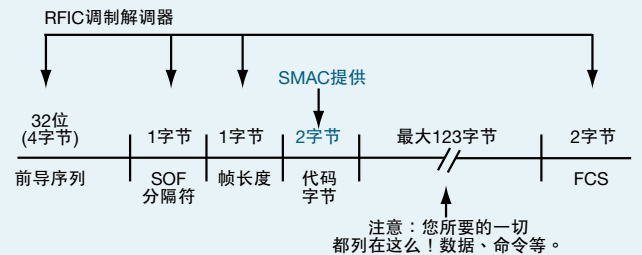


表2：SMAC的数字

存储器(Flash, RAM)		
	MIN	MAX
仅SMAC	2.52 KB, 6B	4.36 KB, 12B
加密模块	0.1 KB, 3B	0.15 KB, 3B
OTAP模块	2.3 KB, 318B	2.57 KB, 461B
硬件(MCU管脚)		
	MIN	MAX
收发器连接	5	7
RF支持	0	4

软件架构

实质上，SMAC可以定义为收发器和MCU之间的驱动软件。它还包括初始化MCU (MCUInit)和外围设备的功能，例如串行通讯接口(SCI)、液晶显示(LCD)、内核调试接口(KDI)和发光二极管(LED)、以及加密和无线编程(OTAP)模块。SMAC层是应用和SMAC内核之间的连接。

SMAC基本功能

SMAC的一些最重要的基本功能包括：

传送

MCPSDataRequest是一个函数模块，用于传送数据包。您必须为创建好的数据包提供一个指针。

接收

MLMERXEnableRequest和MLMERXDisableRequest用于启动/关闭收发器的接收功能。定时参数可以传递给该函数，从而以固定的时间长度来等待接收。

MCPSDataIndication是一个回叫函数，每次成功接收数据包或出现超时后执行该函数。建议该函数应尽量快速地在中断内部执行的。

能源测量和管理

MLMEEnergyDetect和MLMELinkQuality可以用于测量数据包传送前后的信道能量分贝数(dBm)。您可以采用这些功能创建洁净信道评估(CCA)算法和程序，有效地确定传输到最近的设备需要多少分贝。

MLMEMC13192FEGainAdjust用于调节前一个函数的输出，使之更准确。准确的读数帮助您确认设备是“近”还是“远”。

MLMEMC13192PAOutputAdjust设置收发器用于传送数据包的功率。如果您发现设备处于可测量的范围，功率可以随时改变。

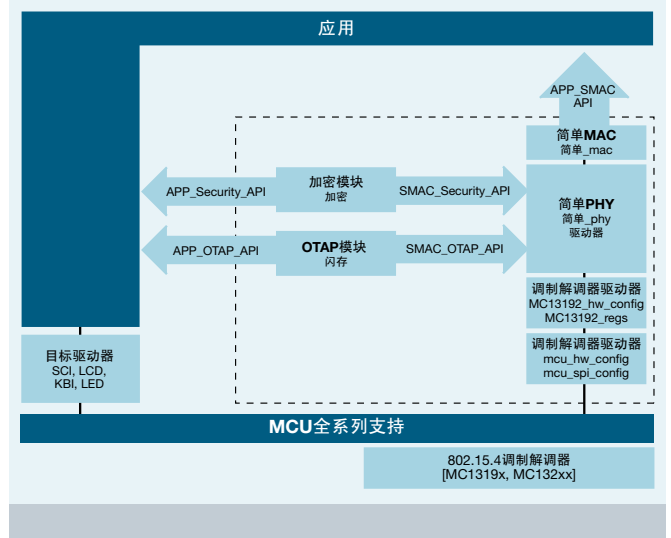
低功耗

MLMEHibernateRequest和MLMEDozeRequest用于将收发器设置为低功耗状态，其休眠状态可实现最低功耗。

MLMEWakeRequest用于将收发器从低功率模式唤醒。有些模式要求MCU采用内部时钟唤醒。

您可以使用这些功能创建小型的自定义“网络层”，这种网络层能够管理多种ID和消息类型。您甚至可以创建自己的CCA或算法，选择最佳信道。有很多方法可以增强其功能，如多个函数的集合(CCA-TX-ACK)。

图3：软件架构



应用和模块

无线编程(OTAP)

OTAP函数用作无线引导程序，让您能够存储固件，再选择一个网络中的设备，然后轻松地点击一下就能更新其固件。这个应用可以通过改进后提供更多的功能，如备份应用或下载信息(传感器测量的数据或其他保存在闪存中的数据)。

转发器

如果您的应用需要较大的范围，而您又不想使用外部设备，您可以使用这种“智能转发器”来增加范围。如果此器件检测到范围之内另一个器件的应答信号，数据包就不会转发。您可以改进这种转发器，从而创建一个简单的路由器来管理您的网络。

表4：无线编程的流程

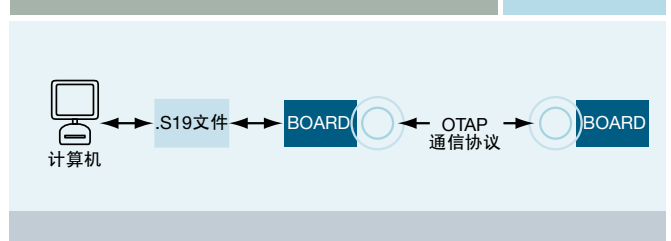
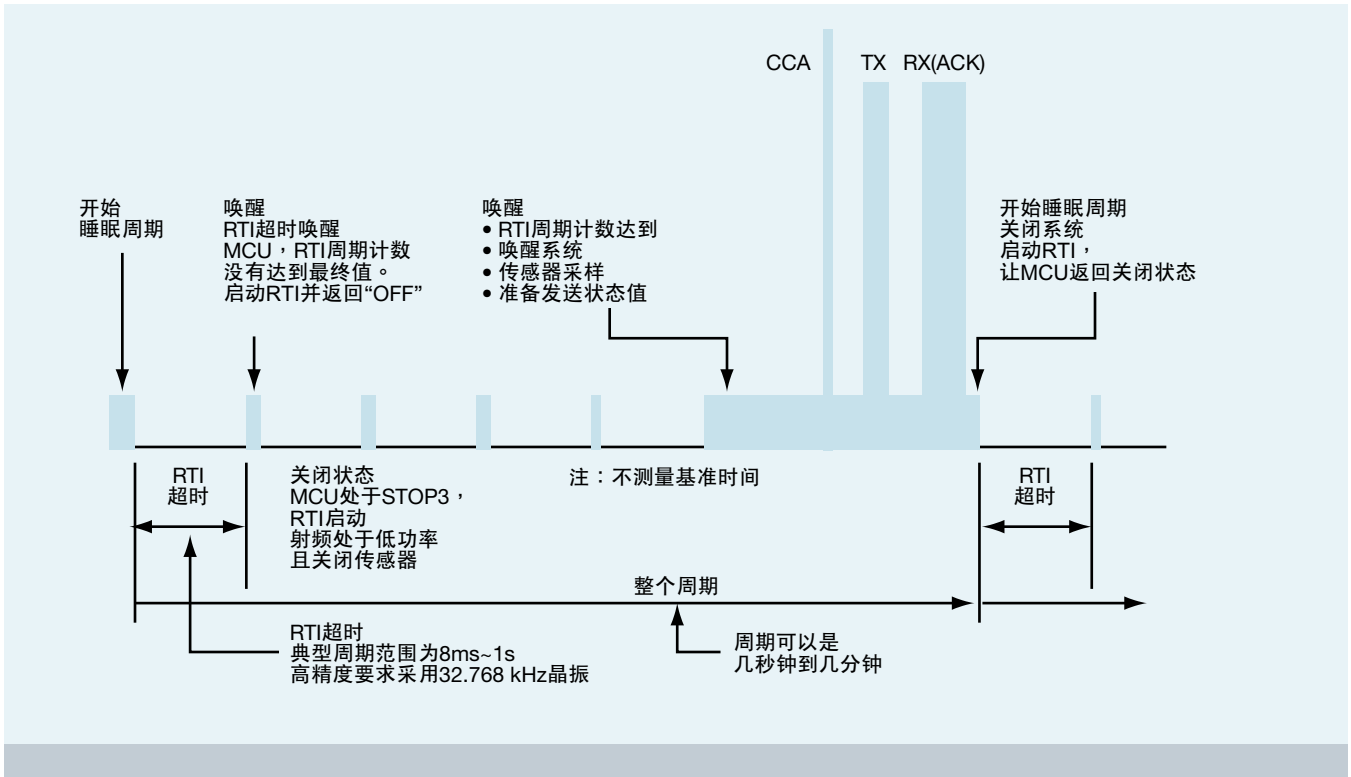


图5：典型的无线传感器节点概览



混合模式

如图1所示，SMAC采用2个字节的代码作为它的发送和接收数据包。每次收到数据包，SMAC会丢弃所有不包含这些字节的数据包。SMAC也可以禁用这些代码字节，从而可接受ZigBee协议栈和802.15.4 MAC/PHY的数据包，使功能更多样化。

加密

SMAC采用一个预定义的密钥和XOR操作实现加密。这可能看起来不像非常严密的算法，但由于提供了源代码，您完全可以把您自己的加密算法加进去。

提示

ZigBee和802.15.4器件在大多数时间里都处于低功率模式，这也是您的SMAC应用的最大优点。在MCU中，您可以采用低功率模式(如Stop 3)，并使用实时中断(RTI)以固定的时间间隔来唤醒设备。收发器可以在睡眠期间进入休眠，然后在工作期间唤醒。CCA可以通过能量检测功能来实现，可检测何时传送或不传送。在工作期间，设备可以发送它的独特ID，以及通过模拟数字转换器(ADC)从传感器读取的特定数据。如果该周期计算精确，电池的寿命能够持续数几年。

通过SMAC，可以进入一个低成本应用的世界。没有做不到，只有想不到。

Gonzalo Delgado Huitrón是墨西哥飞思卡尔RTAC团队的成员。他是一名应用工程师，工作于射频为品部。他的专业技术领域是ZigBee、射频ID (RFID)和8位MCU。

Jaime Herrero Gallardo是飞思卡尔射频产品部的应用工程师。他是SMAC开发者团队的成员。他的专业技术是实时操作系统(RTOS)、ColdFire嵌入式控制器和S08 MCU。

DDR存储器 比较和概述

我们认为计算机最重要的功能是保存大量信息，计算机保存信息的地方我们通常称为“内存”。具体而言，就是随机存储器(RAM)，它保存可以快速、直接访问的易失性信息。考虑到不断增长的系统需要越来越快的速度和效率，因此了解双倍数据速率(DDR)内存对系统开发人员非常重要。

由于处理器的速度提高，RAM内存演进到高性能RAM芯片集，称为DDR同步动态随机存储器(SDRAM)。在时钟周期的上升和下降沿读取数据，能让处理速率翻倍。这和原来的、在时钟周期单沿读取数据的单数据速率(SDR) SDRAM相反。

除了广为人知的计算机应用，DDR存储器还可以广泛应用在其他高速、大内存要求的应用中，如显卡，显卡需要在很短的时间内处理大量信息，达到最高图形处理效率。采用多个刀片或单功能主板，由单个较高效率的电源提供动力的刀片服务器，也需要快速内存存取。这使刀片能够相互快速传输可靠信息，创造更多机会来降低功耗。内存设备在联网和通信应用中也需，任务包括简单地址查找到流量整形/管辖和缓冲器管理。

本文介绍了DDR存储器的主要特征，以及这种内存要求的电源规格。

DDR存储器特点

DDR存储器的主要优势是能够在时钟周期的上升和下降沿读取数据，让特定时钟频率的数据速率翻倍。例如，在DDR200设备中，数据传输频率是200 MHz，但总线速度是100 MHz。

DDR1、DDR2和DDR3内存分别采用2.5、1.8和1.5V供电电压供电，因此在电源管理中要比使用3.3V的常规SDRAM芯片集产生更少的热量和提供更高的效率。

时延是DDR存储器的另一个特征。DDR1存储器时延是通过系列数字(如2-3-2-6-T1，3-4-4-8或2-2-2-5)给出的。这些数字显示时钟脉冲的数字，让存储器执行特定操作，数字越小，存储器越快。

这些数字代表的操作如下：

- CL- tRCD - tRP - tRAS - CMD。要了解这些操作，您必须记住存储器的内部结构为矩阵，数字保存在矩阵中的横列和纵列的交叉点。
- CL：列地址选通脉冲(CAS)延迟是处理器向存储器请求数据和存储器回复请求所需的时间。
- tRCD：行地址选通脉冲(RAS)到CAS延迟是激活保存阵列的行(RAS)和列(CAS)中的数据所需的时间。
- tRP：RAS预充电是禁用访问某一行数据和开始访问另一行数据所需的时间。
- tRAS：激活预充电是下一次访问内存能够开始之前内存必须等待的时间长度。
- CMD：命令速率是存储器芯片激活和第一个命令可以发送到存储器的时间间隔。有时不通知该数值，通常为T1(1种时钟速度)或T2(2种时钟速度)。

表1是RAM内存芯片集的时钟和传输速率比较，这种比较可以在当今的计算机上发现，包括SDR、DDR、DDR2和未来DDR3模块。

表1：SDRAM内存速度比较

存储器	技术	额定时钟	实际最大时钟速率	传输速率
PC66	SDRAM	66 MHz	66 MHz	533 MB/s
PC100	SDRAM	100 MHz	100 MHz	800 MB/s
PC133	SDRAM	133 MHz	133 MHz	1,066 MB/s
DDR200	DDR-SDRAM	200 MHz	100 MHz	1,600 MB/s
DDR266	DDR-SDRAM	266 MHz	133 MHz	2,100 MB/s
DDR333	DDR-SDRAM	333 MHz	166 MHz	2,700 MB/s
DDR400	DDR-SDRAM	400 MHz	200 MHz	3,200 MB/s
DDR2-400	DDR2-SDRAM	400 MHz	200 MHz	3,200 MB/s
DDR2-533	DDR2-SDRAM	533 MHz	266 MHz	4,264 MB/s
DDR2-667	DDR2-SDRAM	667 MHz	333 MHz	5,336 MB/s
DDR2-800	DDR2-SDRAM	800 MHz	400 MHz	6,400 MB/s
DDR3-800	DDR3-SDRAM	800 MHz	400 MHz	6,400 MB/s
DDR3-1066	DDR3-SDRAM	1066 MHz	533 MHz	8,528 MB/s
DDR3-1333	DDR3-SDRAM	1333 MHz	666 MHz	10,664 MB/s
DDR3-1600	DDR3-SRAM	1600 MHz	800 MHz	12,800 MB/s

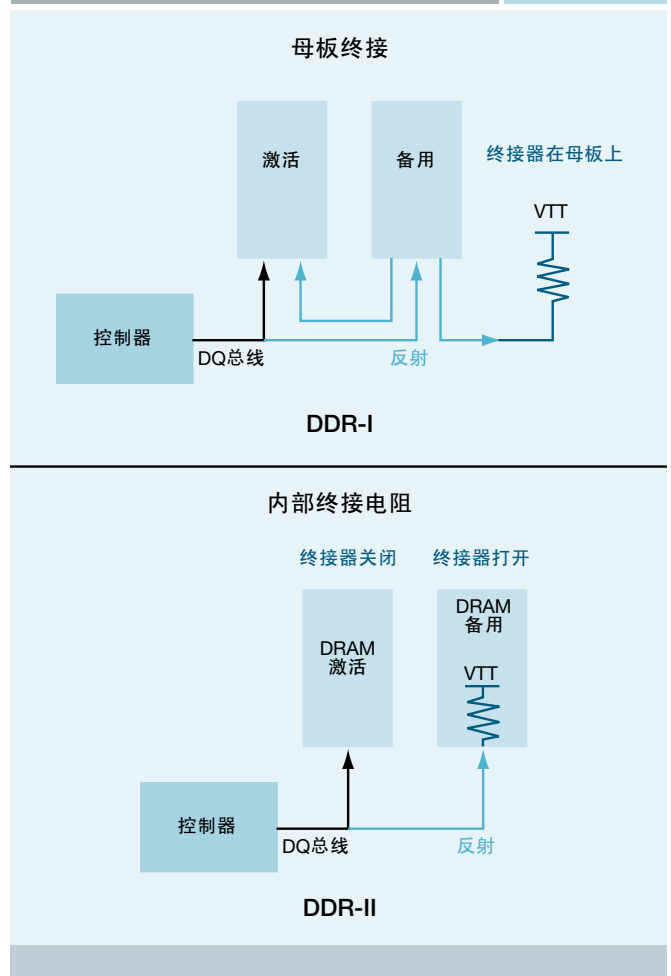
DDR内存的类型

目前有三代DDR存储器：

1. DDR1内存，最大额定时钟是400 MHz，64-位(8字节)数据总线现在变得陈旧，不再大批量生产。技术正在采用新的方法使RAM内存达到更快速度/数据速率。
2. DDR2技术采用400~800MHz的数据速率和64位(8字节)数据总线代替DDR。DDR2存储器被RAM厂家广泛生产，物理上不兼容上一代DDR存储器。
3. DDR3技术捡起了DDR2丢掉的东西(800 Mbps带宽)，将速度提高到1.6 Gbps。ELPIDA已经宣布推出的芯片之一包含最多512兆位的DDR3 SDRAM，卷访问时间为8.75秒(CL7 延迟)，数据传输速率为1.6 Gbps (1.6 GHz)。1.5V DDR3电压水平与DDR2存储器相比还可以节约一些电能。更有趣的是，甚至在较低电压1.36V下，DDR3 RAM也能够以1.333 GHz (DDR3-1333)速率良好运行，带有CL6延迟(8.4秒的CAS总时间)，与目前最快的DDR2存储器的CAS时间匹配。

图2.0.1 显示DDR2/DDR3存储器类型的内部终接电阻(On-die termination ODT)，与DDR1的 motherboard 终接比较，这是最初两种类型物理上不兼容DDR1设备的主要原因。

图2.0.1：DDR-I和DDR-II的终接差异



DDR2与DDR3

DDR2和DDR3模块的主要区别是：

- DDR2存储器包括400 MHz、533 MHz、667 MHz和800 MHz版本，而DDR3存储器包括800 MHz、1066 MHz、1333 MHz和1600 MHz版本。两种类型都能让特定时钟频率的数据速率翻倍，因此列出的时钟是额定时钟，不是真正的时钟。真正的时钟是额定时钟除以2，例如，DDR2-667存储器实际上以333 MHz频率运行。
- 除了带宽增加，DDR3使用的功率也比DDR2小，操作电压为1.5V，比DDR2 (1.8V)减少16.3%。DDR2和DDR3存储器还具有节能特点，例如更小的页面尺寸和激活的电源关闭模式。这种功率消耗优势让DDR3存储特别适合于笔记本电脑、服务器和低功率移动应用。
- 为输出数据缓冲器最新推出的自动校准功能，提高了电压和温度变化过程中系统计时预算的控制能力，这种功能支持强大的、高性能操作，是DDR3架构的主要优势之一。
- DDR3设备引入中断重置，提高了系统灵活性
- 在DDR2存储器中，CL参数是存储器延迟提交被请求数据的时间，可以是3-5个时钟周期，而在DDR3存储器上，CL可以是5-10个时钟周期。
- 在DDR2存储器中，依靠芯片，有0-5个时钟周期的额外延迟(AL)，因此在带有CL4的DDR2存储器中，AL1延迟是5个时钟周期。
- DDR2存储器有写入延迟，相当于读取延迟(CL + AL)减去1。
- 在内部，DDR2存储器的控制器从存储区域预载4个数据位(称为预读取的任务)，而DDR3存储器的内部控制器预先加载8个位。

常见DDR电源要求

DDR x SDRAM设备的VTT和VREF考虑事项

终接电压(VTT)提供必须是采用1.5V输出电压(VDDQ)的灌电流和源电流。这意味着没有分流就不能使用标准切换电源来供应灌电流。由于每一条数据线路连接到电阻相对较低的VTT，因此电源必须相当稳定。电源的任何噪音直接进入数据线路。

必须提供充足的大型支路电容，使电源保持在1/2 VDDQ。同样的噪音问题要求VTT不能产生电压参考(VREF)信号，必须从VDDQ产生，且带有1%或更好的电阻分配器。不要试图用一个分配器生成VREF，并把它从控制器路由到存储器设备。在每台设备上生成本地VREF。使用分散的电阻器生成VREF，不使用电阻器包。

DDR存储器电源管理

DDR1存储器具有推拉输出缓冲器，而输入接收器是一个差动级，需要参考偏移中点，VREF，因此它要求输入电压终接支持源电流和灌电流。这种最新功能可以区分DDR VTT和计算机母板上出现的其他终接。前端系统总线(FSB)终接将CPU连接到存储器信道中心(MCH)，由于终接电源正限，只需要灌电流功能，因此这种DDR VTT终接不能重复使用或采用原来的VTT终接架构，需要新的设计。图3.2.1显示了DDR1存储器的典型电源管理配置。

在驱动芯片集的任意输出缓冲器和存储器模块上的相应输入接收器之间，您必须使用电阻器RT和RS终止路由跟踪或stub(图3.2.1)。加上所有电阻，包括输出缓冲器，每一条终接电线可以支持±16.2mA的灌电流或源电流(高于原来的值±15.2 mA，2000年6月修订JESD79)。对于具有较长跟踪时间的系统，在转发器和接收器之间，可能需要在线路的两端终接，让电流翻倍。

VTT 和VDDQ的峰值和平均电流消耗是正确设置电源系统大小所需的两个参数。要找到终接电压的峰值电源要求，我们必须确定存储器系统中的线路总数。

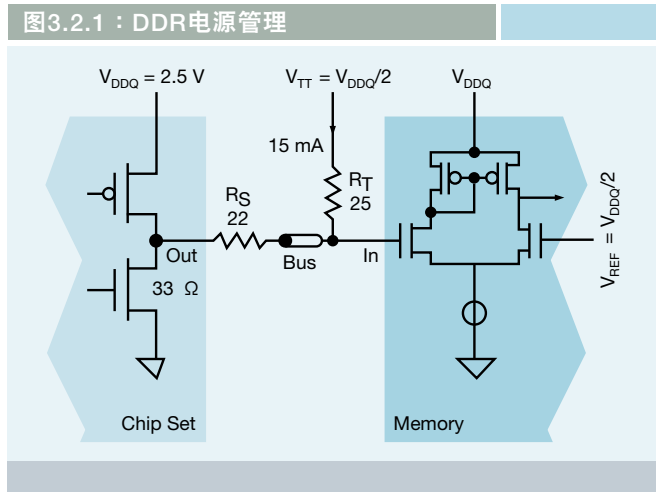


表2：SDRAM设备比较

条目	DDR3 SDRAM	DDR2 SDRAM	DDR SDRAM
时钟频率	400/533/667/ 800 MHz	200/266/333/400 MHz	100/133/166/200 MHz
传输数据速率	800/1066/1333/ 1600 Mbps	400/533/667/800 Mbps	200/266/333/400 Mbps
I/O宽度	x4/x8/x16	x4/x8/x16	x4/x8/x16/x32
预读取位宽度	8位	4位	2位
时钟输入	差分时钟	差分时钟	差分时钟
突发长度	8, 4(突发突变)	4, 8	2, 4, 8
数据选通	差分数据选通	差分数据选通	单一数据选通
电源电压	1.5V	1.8V	2.5V
接口	SSTL_15	SSTL_18	SSTL_2
/CAS时延(CL)	5, 6, 7, 8, 9, 10时钟	3, 4, 5时钟	2, 2.5, 3时钟
内部终接电阻(ODT)	支持	支持	不支持
组件封装	FBGA	FBGA	TSOP(II) / FBGA / LQFP

电源时序违反

DDR存储器必须采用特定方式接通电源。违反通电顺序会导致不明确操作。而且，当DDR设备需要关闭时，断电顺序是一种省电方式，不需要所有其他设备连接到相同电源。违反这种顺序就不会出现较好的省电结果。

DDRx存储器的典型应用

市场分析显示DDR目前应用于50%以上的电子系统，其使用在未来几年有望增加到80%。DDR不是而且永远不是“所有事情都适用于所有设计”的一种解决技术。DDR存储器适合于具有较高读写比率的设计。例如，四倍数据速率存储器设计用于需要50%读写比率的应用。

目前，DDR存储器主要用于双列直插内存模块(DIMM) RAM设备中的计算机应用。与32位微控制器和DSP连接时，可以使用DDR存储器。由于很多存储器的运行采用64位数据总线，微控制器必须获得双倍数据，才能首先获得不太重要的32位，然后获得比较重要的32位。DDR存储器还一直用于连接FPGA，为这类设备提供广泛编程灵活性。FPGA常常用于定制结合了很多数字模块的应用，例如微控制器和特定应用硬件、USB控制器和印刷模块。我们也通常会发现FPGA用作内存控制器，连接其他设备。DDR存储器适合这些类型的设备，因为与其他RAM存储器相比，它们的存储速度快，并且价格便宜。

结论

为了保持竞争力，您必须创建新的设计，利用当今的先进技术。要弄清楚特定设计挑战的最佳解决方案，重要的是进行认真研究，让您有机会衡量技术(如DDR存储器)的优势，把它们整合到新的产品开发中。本文为您提供了DDR存储器的简单、有用的知识，介绍了DDR存储器与其他存储器有何不同，让您在选择适合您的特定需求的最佳解决方案时有一个良好开端。

参考资料

- JEDEC STANDARD JESD79, June 2000 and JESD8-9 of Sep.1998.
- Understanding Ram Timings (Article) www.hardwaresecrets.com/article/26/1
- DDR Memories require efficient power management. (Article) powerelectronics.com/mag/power_ddr_memories_require/
- ELPIDA 1 Gb DDR2 SDRAM memory datasheet

Joaquin Romo是应用工程师，在电源管理和发动机控制应用部门工作。Romo负责创建新的电源管理产品、开发应用软件和评估板，致力于消费电子市场。

在电机控制应用中 使用HCS08 TPM模块

设计人员可以在众多的微控制器中进行选择，为变速驱动提供数字控制。而基于飞思卡尔HCS08内核的微控制器特别适合低端电机控制的应用。

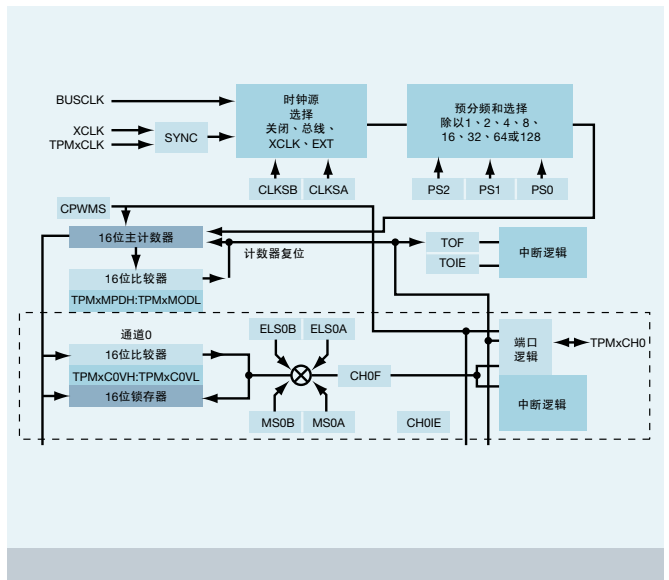
基于这种高性能HCS08内核的器件支持1.8–5V的应用，还封装了许多有用的功能，如高达60 KB的闪存、高达2 KB的RAM、无需要外部器件的灵活的内部时钟发生器、低电压检测、模数转换器(ADC)、定时器(TIM)模块、串行通信模块等。本文介绍了采用TIM模块实现各种脉宽调制的技术，以及所选用的PWM调制方法是如何影响电机控制应用性能的。

PWM调制方法可根据参数的不同进行区分，例如边沿/中心对齐的PWM、单极/双极PWM、互补/独立PWM、PWM通道的数量等。

定时器/PWM模块(TPM模块)源自著名的HC08系列的TIM模块。TPM模块的基本结构请见图1。主要特性是：

- 每个模块多达6个通道
 - 每个通道都可以是输入捕捉、输出比较或带缓冲的边沿对齐的PWM
 - 上升沿、下降沿或任意边沿输入捕捉触发
 - 置位、清零或翻转的输出比较操作
 - 可选的PWM输出极性
- 每个TPM的所有通道都可配置为带缓冲的中心对齐的脉宽调制(CPWM)
- 每个TPM的预分频器时钟源可以独立选择为总线时钟、固定系统时钟或外部管脚
 - 预分频率可分为1、2、4、8、16、32、64或128
 - 外部时钟输入
- 16位自由运行或上/下(CPWM)计数运行
- 16位模数寄存器可控制计数范围
- 定时器系统启动
- 每个信道一个中断，加上每个TPM模块的计数终点中断

图1：TPM模块的基本结构



中心对齐的PWM调制的先进功能将其应用领域扩展到小型家电的电机控制中。

在PWM的生成中需要考虑的另一件事是电流或反电动势电压的检测。低成本电机控制应用通常用分流电阻来检测电流。这时，电流必须在PWM脉冲的中心点来检测。这就要求TPM和ADC模块的同步。由于HCS08系列器件不具备TPM与ADC的同步，在PWM的生成中就必须考虑如何同步这两个模块。这种情况类似于无传感器无刷直流(BLDC)电机控制应用，在这种应用中，反电动势电压也必须在PWM脉冲的中心点采样。其他与PWM生成有关的电机控制应用特性将在后面详细介绍。

边沿/中心对齐的PWM调制

该参数定义了不同TPM通道的脉冲是如何同步的。采用边沿对齐的PWM调制，所有脉冲在PWM周期的起始点开始。采用中心对齐的PWM调制，所有脉冲以脉冲的中心点为中心对齐。中心对齐的PWM的优势可以在三相正弦PWM调制中看出来。如果所有相的占空比都不同，那么晶体管的开关就会分散到整个PWM周期。由于每次晶体管开关都会产生电磁噪声，因此总的噪声水平就会比较低。如果所有脉冲都以脉冲的中心点进行同步，那么它们就可以用于TPM到ADC的同步。输出信号的极性可以设置，这样就可以在脉冲的中心点得到溢出中断。这个中断可用于启动ADC转换。边沿/中心对齐模式可以通过TPMxSC寄存器中的CPWMS位进行设置。

单极/双极和独立/互补PWM调制

单极/双极PWM定义电压如何施加到电机上。如果电机在一个PWM周期内只能看到直流总线和零点电压，我们就称这种调制方法为单极性。如果电机在一个PWM周期内可以看到直流总线电压的两极，该方法就称为双极性。单极性PWM也称为“软开关”，而双极性PWM称为“硬开关”。第二种区别是相对的半导体开关的通断方式。如果它们互相独立地通断动作，我们称之为独立开关。如果它们以互补方式通断动作，我们称之为互补开关。这两种PWM功能可以相互结合，这样我们就可以得到四种可能的组合：单极独立、单极互补、双极独立和双极互补PWM调制。调制方法的选择对驱动性能有很大的影响。各种调制技术的特性可以用一个连接到H桥的直流电机来说明，它包括四个晶体管，Q1-Q4，带有反并联二极管（见图2）。对于BLDC电机的六步控制，可以考虑相同的拓扑结构，因为在任一时刻只有两相通电。

图2：H桥连接直流电机

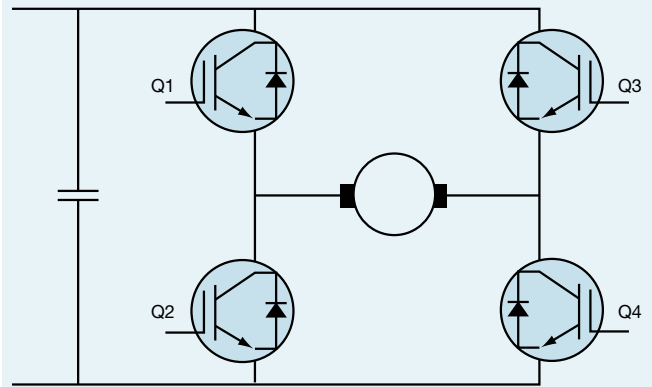


图3：独立双极PWM调制—阶段1

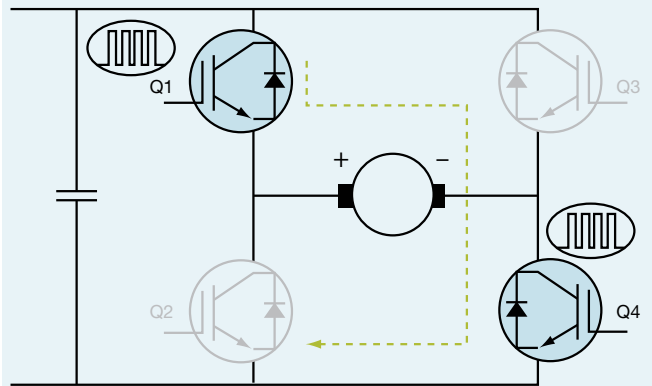


图4：独立双极PWM调制—阶段2

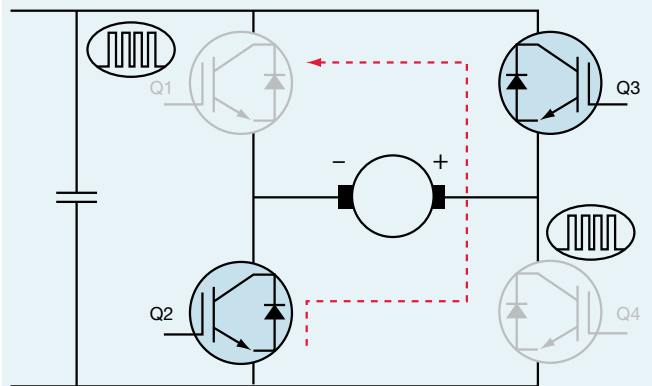
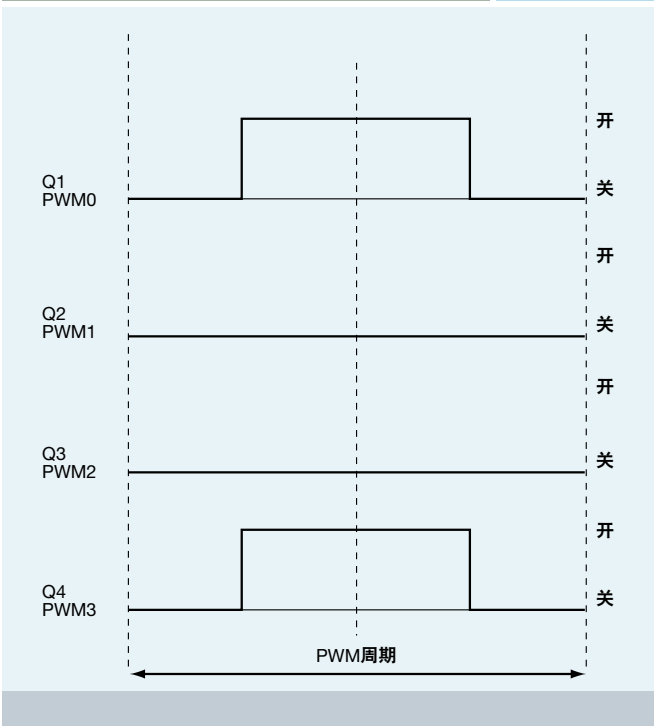


图5：独立双极PWM调制—PWM周期细节
(中心对齐)



独立双极PWM调制

独立双极PWM调制的原理请见图3和4。PWM周期可以分为三个阶段。在第一个阶段，晶体管Q1和Q4由同一个信号驱动。当Q1和Q4开通时，直流电机连接到直流总线电压的正端，直流电机上的电流就会上升。在第二个阶段，晶体管Q1和Q4关断，电流开始下降。电流流经Q2和Q3的续流二极管，而直流电机则连接到直流总线电压的负端。此时独立—各阶段中的晶体管(Q1、Q2和Q3、Q4)独立通断，而且双极—直流电机可以看到直流总线电压的两个极性端。如果直流电机的电流在PWM周期结束之前下降为零，就有了第三个阶段。流经直流电机的电流为零，直流电机断开与电源的连接。一个PWM周期的细节可见于图5。在整个PWM周期晶体管Q2和Q3都是关断的。要以相反的方向运行直流电机，则晶体管Q2和Q3由控制信号驱动，Q1和Q4关断。

这种PWM调制可以是边沿对齐或中心对齐的PWM。由于两个晶体管是同时通断的，因此在降低电磁噪声方面没有什么好处。对齐方式可以根据电流检测方法来选择。

图6：独立单极PWM调制—阶段1

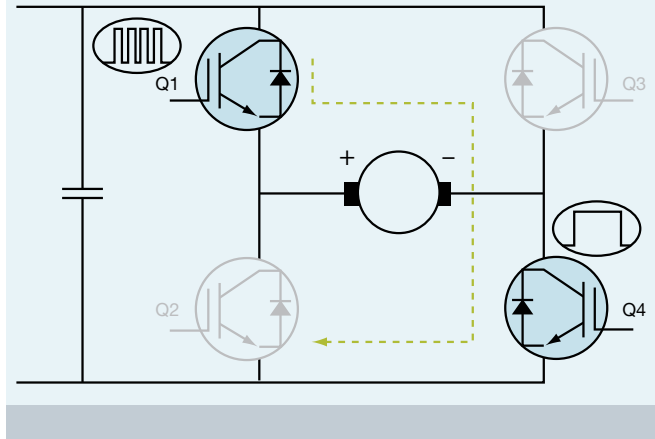
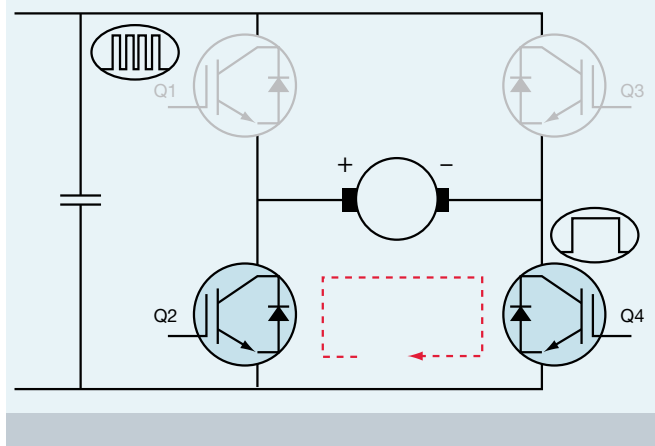
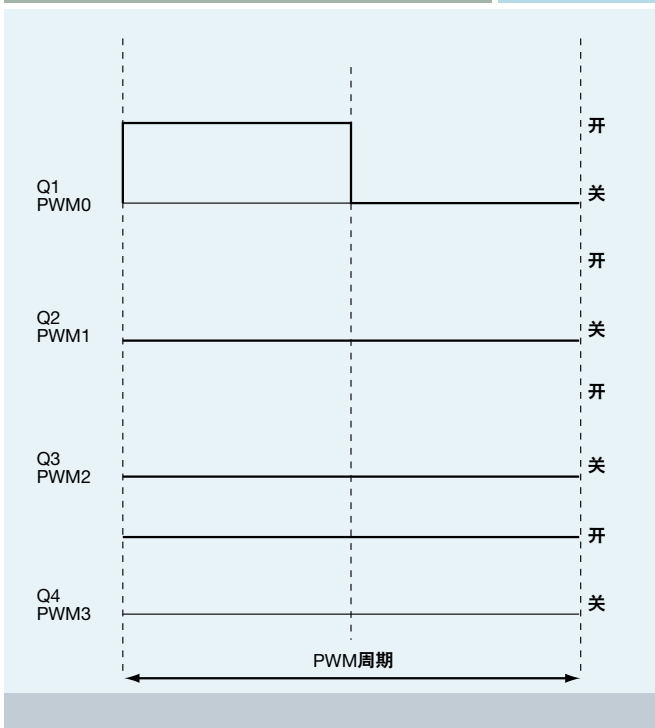


图7：独立单极PWM调制—阶段2



采用TPM 模块很容易实现这种调制。两个通道设置为产生相同的信号。如果符合条件，可以用单个定时器通道生成两个晶体管(Q1+Q4或Q2+Q3)的信号。另一方面也存在一些缺陷，因为从间断电流到连续电流模式的转换会极大地改变驱动的特性。处于间断电流时直流电机呈现低转矩，而在连续电流时转矩会大大增加。这种驱动可以在两个象限内运行，不能主动制动。从电流检测的角度来说，占空比会从0% 变为100%。这意味著需要某种最小脉冲宽度，以便测量电流。

图8：独立单极PWM调制—PWM周期细节 (边缘对齐)



独立单极PWM调制

图6和图7说明了这种调制和前一个之间的区别。在第一个阶段，晶体管Q1由控制信号驱动，晶体管Q4在整个周期都开通。直流电机连接到直流总线电压的正端，则电流上升。PWM周期的第一个阶段与独立双极PWM调制相同。在第二个阶段，晶体管Q1关断，而晶体管Q4仍然开通。电流仍然流经直流电机，但通过Q2的续流二极管闭环。在此第二个阶段中，电机上出现了零电压。直流电机电流开始下降，但是，由于发动机上的零电压，负斜率非常慢。这意味着对于非常窄的脉冲，电流也能进入持续模式，从而产生更好的驱动特性。直流电机用很小的占空比就能呈现的高转矩，因此调节转矩和速度控制就非常容易。这种驱动可用作电动机而不能用作制动器，只能在两个象限运行。而且，它的电流波动较小，这意味着半导体器件的损耗较小。

图9：互补双极PWM调制—阶段1正向电流

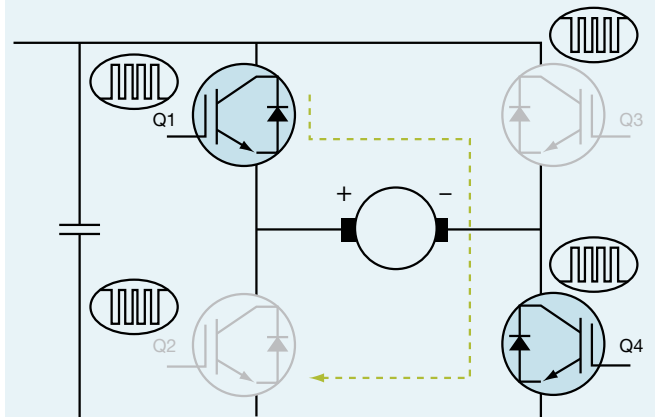
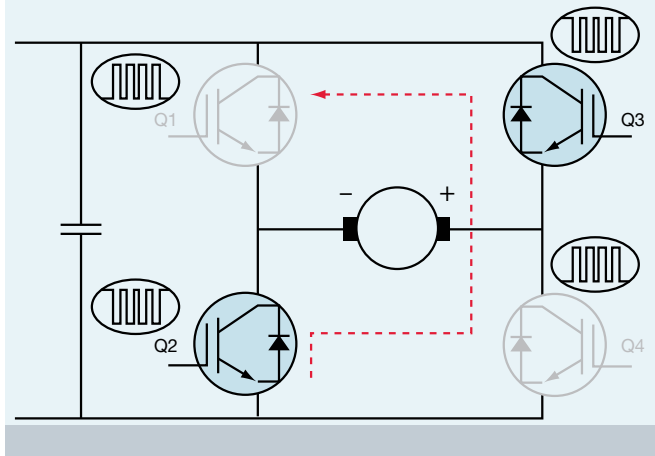


图10：互补双极PWM调制—阶段2正向电流



采用TPM模块的实现方法也很简单。PWM信号由两个TPM通道产生，通常供给上臂晶体管(Q1和Q3)。下臂晶体管可以由其他TPM通道驱动，或者由GPIO管脚驱动，因为它们在整个PWM周期中一直开通。关于电流检测，情况和前一个相同。占空比从0到100%变化，因此需要最小脉冲宽度来进行电流测量。

互补双极PWM调制

与前面的两种调制不同，所有晶体管以互补的方式进行通断。详情请见图9和图10。晶体管Q1和Q4由同一信号驱动，Q2和Q3关断。在第一个阶段，直流电机连接到直流总线电压的正端，直流电机电流上升。在第二个阶段，晶体管Q2和Q3传导电流，Q1和Q4关断。直流电机连接到直流总线电压的负端，流经Q2和Q3续流二极管的直流电机电流下降。如果电流达到零，它可以继续以相反的方向流动。当电流反向时，这种驱动工作于主动制动方式，并在四象限运行工作。在任何情况下电流都以持续模式流动，因此驱动具有稳定的特性。

互补通断要求产生死区时间，以避免交错导通。由于TPM模块没有配备生成死区时间的硬件，因此死区时间必须由软件生成。

由于TPM模块不支持硬件的互补信号，因此TPM模块的设置比较复杂。所有通道的正确设置方法如下所示：

PWM0: (duty_cycle + dead_time),
负继续
(TMPxCnSC:ELSnB=x, TMPxCnSC:ELSnA=1)

PWM1: (duty_cycle - dead_time),
正极性
(TMPxCnSC:ELSnB =1, TMPxCnSC:ELSnA=0)

PWM2: (duty_cycle - dead_time),
正极性
(TMPxCnSC:ELSnB=1, TMPxCnSC:ELSnA=0)

PWM3: (duty_cycle + dead_time),
负极性
(TMPxCnSC:ELSnB=x, TMPxCnSC:ELSnA=1)

duty_cycle在0到100%之间变化。另一方面，电流测量就非常简单。由于在任意时刻电流都流经晶体管，因此电流可以根据实际占空比在PWM周期的前半段或后半段进行测量。PWM周期的细节请见图11。

图11：互补双极PWM调制—PWM周期细节

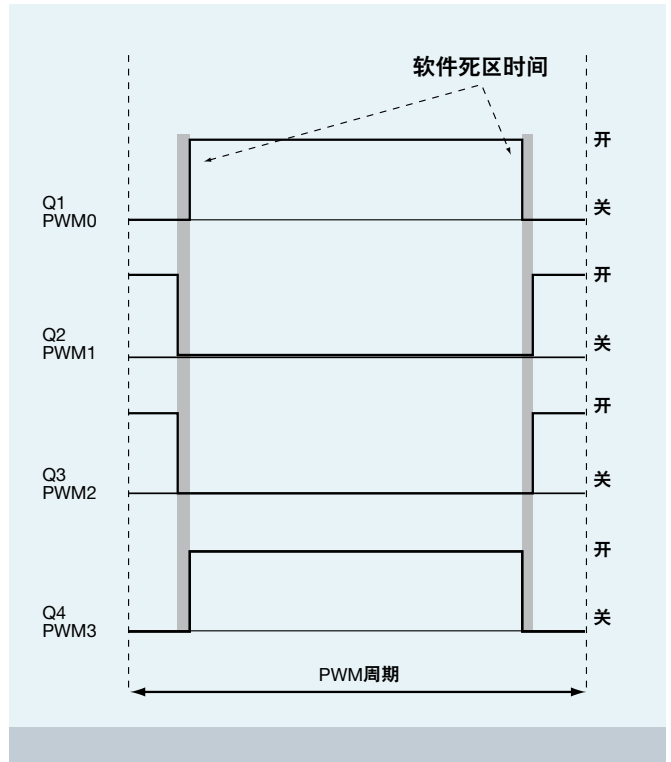


图12：互补单极PWM调制—PWM周期阶段1细节

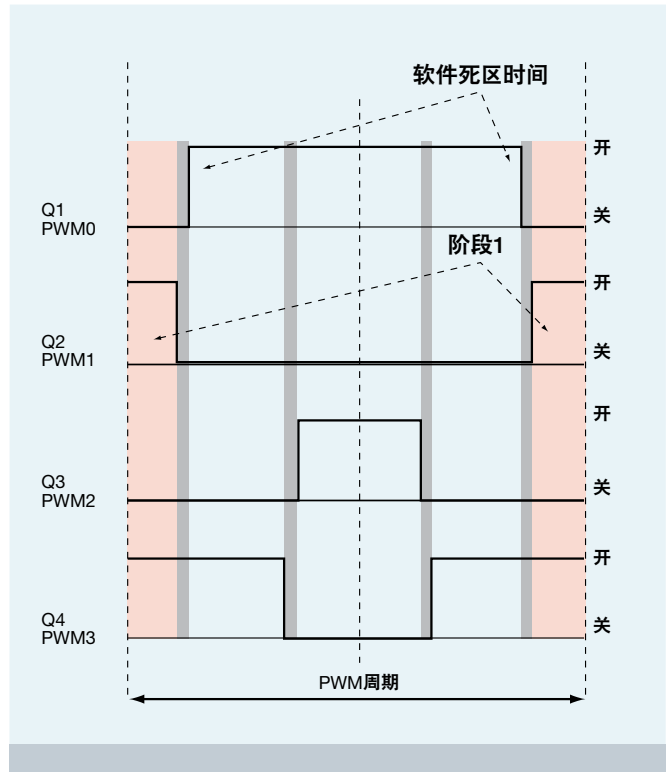
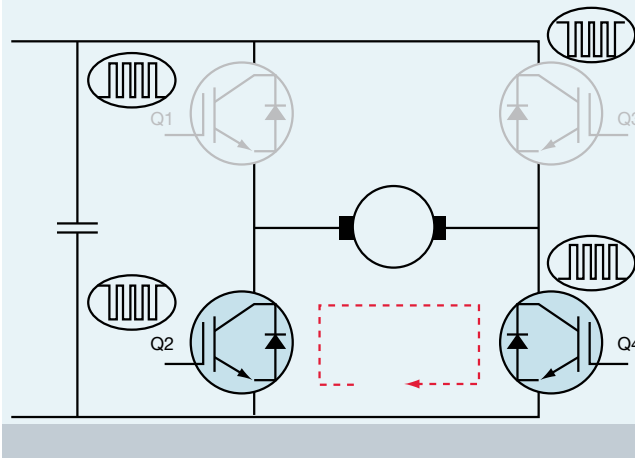


图13：互补单极PWM调制—阶段1



互补单极PWM调制

在这种PWM调制技术中，所有晶体管也都以互补方式进行通断。互补单极PWM调制的PWM周期可以分为三个阶段。第一个阶段请见图12和图13。两个下臂晶体管开通，零电压施加于直流电机，电流流经Q4和Q2的续流二极管。

第二个阶段可见于图14和图15。晶体管Q1和Q4传导电流，直流电机连接到直流总线电压的正端，电流流经直流电机。

第三个阶段与第一个阶段类似，但两个上臂晶体管Q1和Q3开通。零电压施加于电机上，直流电机电流通过Q1和Q3的续流二极管闭环。阶段3的描述请见图16。

这种类型的PWM调制产生最小的电流波动，因为电机两次连接到电源。与前面的PWM调制技术相比，它能够用较低的切换频率获得相同的电流纹波。其驱动特性也等同于前一个PWM调制类型。

这种调制的实现方法也相当复杂，因为互补信号和死区时间都必须由软件产生。TPM通道的配置请见下一段落：

PWM0: $\text{MODULO}/2 \pm \text{duty_cycle} + \text{dead time}$
 PWM1: $\text{MODULO}/2 \pm \text{duty_cycle} - \text{dead time}$
 PWM2: $\text{MODULO}/2 -/+ \text{duty_cycle} + \text{dead time}$
 PWM3: $\text{MODULO}/2 -/+ \text{duty_cycle} - \text{dead time}$

图14：互补单极PWM调制—PWM周期阶段2细节

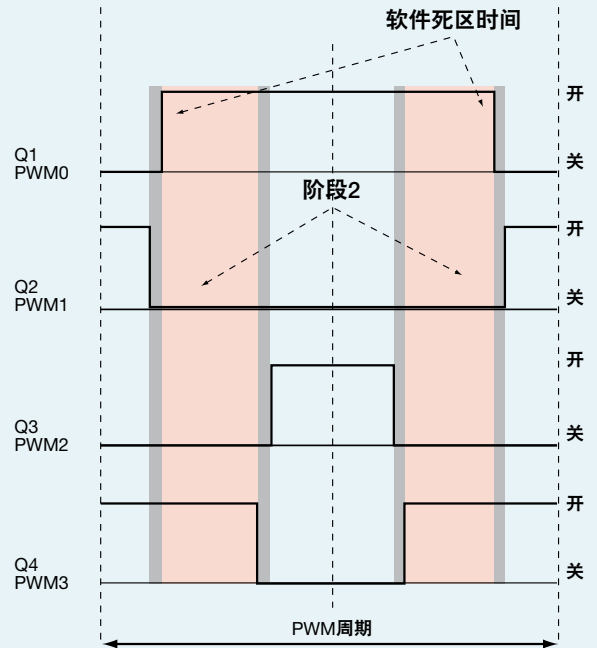
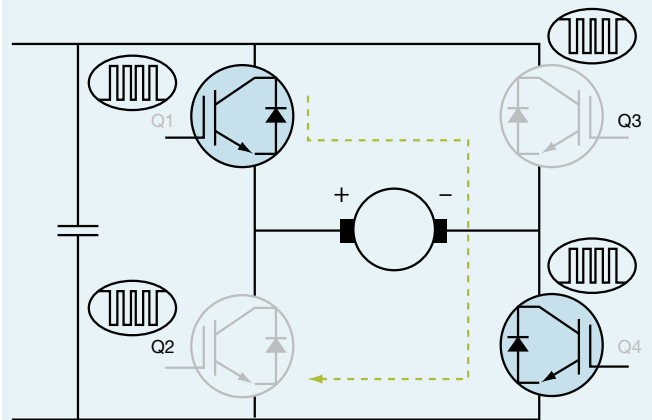


图15：互补单极PWM调制—阶段2



占空比在0到50%之间变化。所有上臂通道设置为负极性 (TMPxCnSC: ELSnB=x, TMPxCnSC: ELSnA =1)，而所有下臂通道设置为正极性 (TMPxCnSC: ELSnB =1, TMPxCnSC: ELSnA =0)。TPM通道配置的符号表示是否将duty_cycle变量加到通道PWM0和1，并从PWM2和PWM3中减去duty_cycle，反之亦然。

正弦PWM调制

考虑三相电机时，正弦PWM调制需要6个TPM通道。这种调制采用中心对齐的PWM，因为可降低电磁噪声。它的三相都采用互补方式工作，而duty_cycle在每个PWM周期由一个正弦表进行调制，并在每一相上移动120个电角度。TMP模块的配置与前述的相同。duty_cycle在0到100%之间变化。所有上臂通道设置为负极性(TMPxCnSC: ELSnB=x, TMPxCnSC: ELSnA =1)，所有下臂通道设置为正极性(TMPxCnSC: ELSnB =1, TMPxCnSC: ELSnA =0)。

根据驱动类型(DC电机 — 单向旋转、DC电机 — 双向旋转、BLDC电机、交流感应电机、永磁同步电机等)，必须选择合适的微控制器。表1显示了根据电机的类型和运行模式所需的TPM通道数量。表2总结HCS08系列的可用产品，包括TPM模块和通道的数量。

使用TPM模块实现PWM调制的实际方法可见于飞思卡尔网站上的应用笔记或设计师参考手册(如DRM086中的BLDC电机互补双极调制)。

图16：互补单极PWM调制—PWM周期阶段3细节

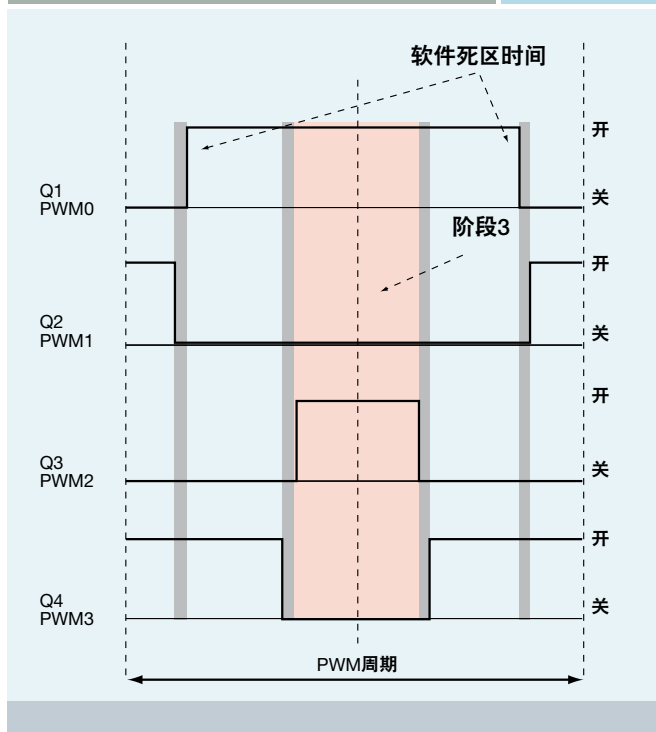


表2：HCS08 MCU概览

MCU	数量	
	TPM	通道
MC9S08QGx	1	2 (1 8-pin)
MC9S08GBx	2	3/5
MC9S08GTx	2	2/2
MC9S08AWx	2	2/6

表1：不同电机类型和运行模式的TPM通道的数量

运行模式	调制技术	TPM通道的数量		
		DC直流电机	BLDC电机	AC/PMSM电机
1Q	单一PWM	1	—	—
2Q	独立双极	4	6	—
	独立单极	2 + 2 GPIO	3 + 3 GPIO	—
4Q	互补双极	4	6	—
	互补单极	4	6	—
—	正弦波调制	—	—	6

Pavel Grasblum作为系统应用工程师在飞思卡尔工作已超过7年。他于2001年获得功率电子和电气传动的博士学位。Grasblum采用众多的飞思卡尔产品(8/16位MCU、DSC)，支持电机控制和电源转换应用。

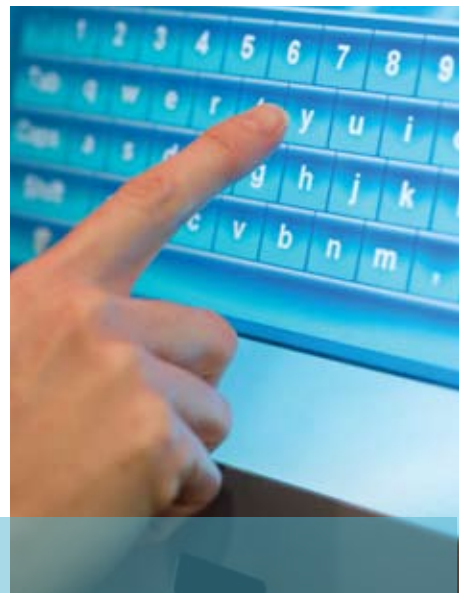


产品概述

8位产品

8位微控制器产品概览

器件	Flash	RAM	USB	ADC通道		SCI (UART)	ESCI	SPI	PC	ACMP	定时器	时钟类型	封装					开发工具		应用/其他特性 * 所有RS08、S08和HC08产品都包括COP、LVI、POR和KBI	
				10位	8位								DFN/QFN	QFP/LQFP	TSSOP	SOIC	DIP	DEMO	EVB		FSICE
MC9S08AW60	60 KB	2 KB		16		2		√	√		6 + 2通道	ICG w/FLL	48	64, 44				√			高集成度，闪存可至5V编程
MC9S08AW32	32 KB	2 KB		16		2		√	√		6 + 2通道	ICG w/FLL	48	64, 44				√			高集成度，闪存可至5V编程
MC9S08AW16	16 KB	1 KB		16		2		√	√		4 + 2通道	ICG w/FLL	48	64, 44				√			高集成度，闪存可至5V编程
MC9S08QG8	8 KB	512B		8		√		√	√	√	2通道，MTIM	ICS	8, 16		16	8	16	√			高性能、低电压，小封装
MC9S08QG4	4 KB	256B		8		√		√	√	√	2通道，MTIM	ICS	8, 16		16	8	8, 16	√			高性能、低电压，小封装
MC9RS08KA2	2 KB	62B								√	MTIM	ICS	6			8	8	√			针对小型MCU的超低端新型RS08内核
MC9RS08KA1	1 KB	62B								√	MTIM	ICS	6			8	8	√			针对小型MCU的超低端新型RS08内核
MC9S08GT16A	16 KB	2 KB		8		√		√	√		3 + 2通道	ICG	48, 32	44			42	√	√		高性能，闪存可低至1.8V编程
MC9S08GT8A	8 KB	1 KB		8		√		√	√		3 + 2通道	ICG	48, 32	44			42	√			闪存可低至1.8V编程，小型封装
MC9S08QD4	4 KB	256B		4							2 + 3通道	ICS				8	8	√			低端、闪存可至5V编程
MC9S08QD2	2 KB	128B		4							2 + 3通道	ICS				8	8	√			低端、闪存可至5V编程
MC9S08LC60	60 KB	4 KB		8 12位		√		2	√	√	2 + 2通道	ICG w/FLL		80, 64				√			集成高段数液晶显示(LCD)驱动器
MC9S08LC36	36 KB	2.5 KB		8 12位		√		2	√	√	2 + 2通道	ICG w/FLL		80, 64				√			集成高段数液晶显示(LCD)驱动器
MC9S08QE128	128 KB	8 KB		24 12位		2		2	2	2	1 + 6通道， 2 + 3通道	ICS		80, 64				√	√		超低功率S08器件，具有1.8V至3.6V的运行电压范围
MC9S08QE64	64 KB	4 KB		24 12位		2		2	2	2	1 + 6通道， 2 + 3通道	ICS		80, 64				√	√		超低功率S08器件，具有1.8V至3.6V的运行电压范围
MC908QC16	16 KB	512B		10			√	√			2通道，16位 4通道，16位	OSC		16, 20, 28		16, 28		√	√		两个独立定时器块，增大闪存和RAM



产品概述

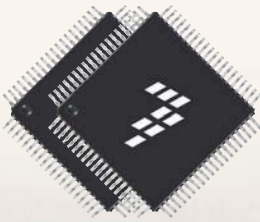
32位产品

32位微控制器产品概览

器件	Freq. (MHz)	MIPS @ Max Freq.	Cache (KB)	SRAM (KB)	Flash (KB)	DMA	GPT*	PIT	10/100 FEC	Crypto	USB	CAN	I2C	UART/ USART/ PSC	SPI	ADC	其他	GPIO Max	调试接口	封装
MCF51QE	50	46		8, 4	128, 64		6通道 · 16位 2x 3通道 · 16位						2		2	12位	RTC w/32kHz Osc	70 16位 RGPIO	ICE和 BDM	LQFP 80, LQFP 64
MCF520x	166	159	8 K I/D	16		16通道	4通道 · 32位	2	1 (可选)				1	3 UART	QSPI		DR/SDR SDRAM 控制器	30, 50	JTAG和 BDM	LQFP 144, QFP 160 MAPBGA 144 MAPBGA 196
MCF521x	66, 80	76		16, 32	128, 256	4通道	4通道 · 32位 4通道 · 16位	2				1 (可选)	1	3 UART	QSPI	12位		33, 44, 55	√	LQFP 64, LQFP 100 MAPBGA 81
MCF521xx	66, 80	76		16	64, 128	4通道	4通道 · 32位 4通道 · 16位	2					2	3 UART	QSPI	12位	RTC w/32kHz Osc	56	JTAG和 BDM	LQFP 64, LQFP 81, MAPBGA 81
MCF5221x	66, 80	76		16	64, 128	4通道	4通道 · 32位 4通道 · 16位	2			全速设备/ 主机/OTG		2	3 UART	QSPI	12位	RTC w/32kHz Osc	56	JTAG和 BDM	LQFP 64, MAPBGA 81
MCF5222x	66, 80	76		16, 32	128, 256	4通道	4通道 · 32位 4通道 · 16位	2			全速设备/ 主机/OTG		1	3 UART	QSPI	12位	RTC	56	JTAG和 BDM	LQFP 64, MAPBGA 81
MCF5223x	60	57		32	128, 256	4通道	4通道 · 32位 4通道 · 16位	2	1	可选		1 (可选)	1	3 UART	QSPI	12位	EPHY, RTC	73	JTAG和 BDM	LQFP 80, LQFP 112, MAPBGA 121
MCF523x	80, 100, 150	144	8 K I/D	64		4通道	4通道 · 32位	4		可选		标准一个 第二可选	1	3 UART	QSPI		16通道eTPU 可选 32通道eTPU SDR SDRAM	142	√	QFP 160, MAPBGA 196 MAPBGA 256
MCF5253	140	125	8 K I	128		4通道	2通道 · 16位				高速 On-the-Go	2	2	3 UART	QSPI	12位	IDE, I2S SDR SDRAM	60	JTAG和 BDM	MAPBGA 225
MCF527x	100, 166	96, 159	8 K I/D 16 K I/D	64		4通道	4通道 · 32位	4	标准一个 第二可选	可选	可选 全速器件		1	3 UART	QSPI		SDR SDRAM DDR SDRAM	61, 69, 97	√	QFP 160, MAPBGA 196 MAPBGA 256
MCF528x	66, 80	76	2 K I/D	64	256, 512	4通道	4通道 · 32位 8通道 · 16位	4	1			1	1	3 UART	QSPI	10位	SDR SDRAM DDR SDRAM	150	√	MAPBGA 256
MCF532x	240	211	16 K I/D	32		16通道	4通道 · 32位	4		可选	全速主机、 全速/高速** OTG	1 (可选)	1	3 UART	QSPI		SVGA LCD DDR/SDR SDRAM	63, 97	JTAG和 BDM	MAPBGA 196 MAPBGA 256
MCF537x	180, 240	158, 211	16 K I/D	32		16通道	4通道 · 32位	4	1	可选	可选全速 主机、全速 OTG		1	3 UART	QSPI		DDR/SDR SDRAM	46, 62	JTAG和 BDM	QFP 160, MAPBGA 196
MCF547x	200, 266	308, 410	32 K I 32 K D	32		16通道	4通道 · 16位	2	2	可选	可选 高速设备		1	4 PSC	DSPI		PCI DDR/ SDR SDRAM	83, 99	√	PBGA 388
MCF548x	166, 200	256, 308	32 K I 32 K D	32		16通道	4通道 · 16位	2	标准一个 第二可选	可选	可选 高速设备	2	1	4 PSC	DSPI		PCI DDR/ SDR SDRAM	83, 99	√	PBGA 388

*GPT可以支持PWM和/或DMA功能
**USB移动高速功能通过ULPI接口实现

这是我们产品的精选列表。要获得我们的8位和32位微控制器的全部产品概述文档，请在www.freescale.com上搜索“8BITCIPRODMPPFS”和“BRCOLDFIRESUM”。



MC9S08QE128
MCF51QE128

在此正式宣布： 8位和32位从此可以 兼容。

控制器联合体：Flexis™ MCU系列



隆重介绍Flexis系列：全球首个8位和32位之间管脚、外围设备和工具兼容的微控制器。QE128产品族，该系列中的首款芯片对，可为您提供一条跨越价格与性能鸿沟的移植路径。现在，当设计需要时，您可以在8位和32位之间轻松移植并共享工具，如我们的新版本CodeWarrior®微控制器开发工具软件，还带有Processor Expert™。另外，超低功率运行的外围设备能够帮助延长电池寿命，这也是另一种竞争优势。利用前所未有的“控制器联合体”的设计灵活性，加快您的产品面市时间。只有飞思卡尔能够做到。

View our free webcast at freescale.com/flexis

