分解任务、逐一解决，以使嵌入式控制器获得成功

By (Jon Pearson, Cypress Semiconductor Corp.)

如今，尺寸和功能各异的嵌入式控制器比比皆是，尽管消费者很少能够直接目睹。例如，蜂窝电话就包含了多达数十个嵌入式器件，所控制的事项从无线电传输、用户界面、显示器到小键盘，甚至还有背面照明 LED 的光照强度等等，可谓无所不包。发展趋势很明显；接受一个复杂的问题，将之分为若干个较小、较简单且更加明确的问题，并针对具体的任务运用合适的工具。即使是最为普通的嵌入式系统也适用于这一原则，而且，由此产生的结果将是设计周期的缩短、灵活性的增加以及可维护性的提高。赢得成功的关键在于采用一种普遍适用的通信策略，以使对于外界而言，远程和本地功能似乎天衣无缝。

主控器-从动器、主机-客户机、集线器-节点：说的全都是一回事

无论您如何称呼它，其原理归根结底是相同的：集中决策（这是主控器）、分头行动（这些是从动器）。在最为复杂的系统中，这种方法是必不可少和自动执行的。比如：蜂窝电话具有一个主处理器，用于决定屏幕上的显示内容以及将需要由哪些外设（如 LCD 显示控制器或无线电收发器）来完成。而由主处理器来直接控制显示器的各个像素或无线电编码和解码的可能性几乎为零。主控器将告知显示器的从动器件显示什么信息，并由显示器来决定如何显示；主控器向收发器的从动器件提供需要编码的音频信号，而收发器则提供经过其解码的音频信号。

在其他嵌入式系统中，任务的分割或许不那么明显，但是，动机以及由之带来的好处却是相同的。如果一个中央主控器能够与远程从设备以及本地设备进行通信，则系统控制在整个系统中都可以是一致的。为了使一个分割或分布式系统获得成功，需要明确规定（并且为人熟知的）的接口。以一个简单的事实为例，所有汽车的油门踏板都在右侧，而刹车都在左侧，这样，我们在世界任何地方都可以租一辆车，并成功地驶离停车场（之后，您或许将开始遇到诸如当地的交通图以及交通信号的辨识等问题）。

虽然微控制器中提供了许多标准的通信方法，不过，在主/从嵌入式系统中，其中有三种特别常用，就

是 RS232 串行、SPI 和 I²C。采用这些通信接口的从设备可直接在市面上采购。作为较低级的器件实例，其中就包括了模拟-数字转换器（ADC）、数字-模拟转换器（DAC）、串行 EEPROM 以及各类数字 I/O。而电压排序和监视器件以及闭环风扇控制器等则是高级器件的实例。就其自身而言，这些器件的作用微乎其微（甚至根本不起作用）。而当组合成为一个系统并通过一根标准总线与主控器进行通信后，这些从动器就变成了一个大型系统的主要组成部分，而且，借助主控器通过通信总线进行的主动控制，还有望使自身的功能得到增强。

I²C —— 非正式的事实嵌入式标准

当您调查嵌入式产品市场，并研究现售的从动器件以及可被用作主控器件或定制从动器的微控制器时，一种通信方法便会跃然眼前，那就是 I²C（即：集成电路间总线）。其流行有两个原因：廉价（双线/引脚和两个上拉电阻器即可构成该总线）和易用。有许多独特的原因会导致人们选择一种特定的总线（而弃用另一种），不过，就分布式嵌入系统而言，I²C（400kHz）往往是最佳选择。

采取分割任务并逐一解决（我们姑且称之为“分而治之”，译者注）的关键好处在于：简单的问题要比复杂的问题更加容易解决。此外，将器件彼此分开可以减少它们之间的耦合，并提高系统的可靠性。凭借正确的设计（功能的分配和接口的规格），能够避免普遍存在的故障模式。最后，如果能够很好地利用自己的经验，则任务分割将产生多得多的可重用设计要素，从而使得工程师们在实施下一个项目不再需要从头开始（甚至可使下一个项目只需着眼于改进就行了）。

“分而治之”DIY（自动动手）指南

“分而治之”依靠的是一根通信总线，而且，您或许由于某种缘故而偏爱选择一种非 I²C 总线。不过，这并没有什么问题，因为该策略在总线的使用上具有不确定性。需要对主控器/从动器加以明确的描绘，而且，在有些场合，总线将会影响到某些主控器/从动器特性（比如：通信的启动位置）。在本例中使用 I²C 作为总线，这是因为支持该总线的现售器件和微控制器已很普遍，而且测试和调试所需的外部工具价格低廉或易于设计。

这种主控器/从动器原理在任何嵌入式设计中的运用都是以相同的方法来处理的。首先，确定哪些决

策是必需集中做出的，并将它们分配给主机，然后把具体操作分发给从动器去执行。奥妙之处在于如何进行任务的分割。一种高效策略是：“不要让主控器因为任何事情而去等待某个从动器，而且，如果从动器需要从主控器获得某些信息，则它必须呼叫（而且所需的最好是重要信息）”。这种方法在有些场合需要第三根线，这是一根中断线，以使从动器能够“呼叫”主控器。

策略是第一步

您在开始设计之前需要拥有一项策略，而且这种策略是越简单越好。无论是什么策略，它必须帮助您回答这样的问题：我应该做 *X* 还是 *Y*？和所有诸如此类的问题一样，答案有 4 种：X、Y、“无所谓”和“不知道”。如果您的策略经常得出“不知道”的回答，则要么是您的问题不够明确（问题提得很蹩脚，应将其分解成一个或多个的更加具体的问题），要么是您采用的策略过于简单（无效策略，应继续提炼）。应保存一份专门用于记录您的策略、提问、回答以及最终的策略更新（当策略被改变时）的文件，并使之能够随时供项目的所有相关人员使用。在讨论时间表改期的会议上、或者在项目的后期增添人员的时候，这将变成一份颇有价值的辅助材料。

您的策略必须形成文字资料。这样，每个人都能够按照相同的策略来工作，您将可以根据需要进行修正，并且很容易地与其他团队实现共享。遵循一个明确策略的几个小组，不管他们是共同工作还是独立工作，将实现设计的共享并最终导致设计周期的缩短和置信度的提高。

由于设计是一个由若干部件组成的系统（其中的一些您或许不能全面掌控，就像采用现售器件时那样），因此，在计算公式中必须加入各个器件的约束条件。

利用经验，一个明确表述的策略将有助于您的系统的定制部分的设计以及标准器件的选择，从而减少项目后期的折衷并实现更大的灵活性。

让从动器执行尽可能多的任务并向主控器提供高级数据

虽然目标是让一个嵌入式系统中的主控器向从设备提供指令（即：在相同的位置上对它们进行配置），但是，测量、数据分析和操作执行通常将是一个自主过程。将此与上述的策略结合起来（主控器从不

等待，从动器只在需要时发出呼叫），您就能够开始发现其在分布式设计中的价值。每件事都将做到最好，并以仅仅满足需要作为目标。

人类始终就是这样做事的。例如，当我把衬衫丢给洗衣店时，我就不希望他们在衣服洗涤过程的每一步都给我回电话，询问我每件衬衫应该怎么处理（水温多少？与其他衬衫一道洗涤还是单独洗涤？转笼干燥还是滴干？高温熨烫还是低温熨烫？需不需要上浆？）或者更加糟糕，如果洗衣工每做完一步就等我打电话告诉他下一步做什么，那又该怎么办？如果我的选择和指令是必要的，那么我希望在丢下衬衫时只被询问一次，而且是回答有关某些具体问题的决定（比如：我对浆硬领口的偏爱）。换句话说，为什么要把每一项操作的执行和决定的做出都让中央（使用过度的）设备来承担？然而，在一个分布式嵌入设计中，必须采取清醒的计划，并慎重实施。

简单的实例：采用 I^2C IO 扩展器件的面板控制器

该原理最为简单的表现形式就是用于控制一部设备的面板的系统。面板由按钮、开关和 LED 组成。为了实现这种面板设计（它是一个更大、更复杂的设备的子系统），主控器将是已经在负责管理此逻辑框的主处理器（很可能是某种形式的嵌入式 Windows 或 Linux 计算机），而对于从动器，我们将从几十种现售的 I^2C 总线连接输入/输出扩展器件中进行挑选（I^2C 的发明者 Philips 公司可供应许多此类产品，不过，其他十几家公司也具备这样的能力，其中许多器件都具有引脚对引脚和功能对功能的互换性）。

通过选择与 I^2C 总线相连的现售 I/O 扩展器，并让主处理器在上电时对其进行配置，按钮/开关检测输入和 LED 状态输出将被连接至主处理器。任何具有一根 I^2C 总线的处理器都可以是主控器；按钮和 LED 的配置可很容易地改变（特定的从动器件就能做到这一点）。主控器对设备进行软件配置。下图示出了这种简单的系统。

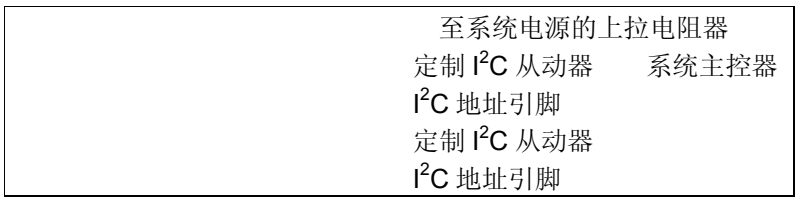
蓝色的圆圈代表按钮输入，红色和绿色形状代表 LED。如果您已经对这些 I^2C I/O 扩展器有所了解，则显然会萌生这样的问题：“为什么使用了两个小器件，而不是一个较大的器件？”部分原因是我偏爱把输入和输出功能隔离开来，还有就是智能设计模块化：允许把输入子系统的变更与输出子系统加以分离。当在主控器中进行决策时，在每项功能的周围放置一个“逻辑框”（box），可使某一特定功能的变更不太可能影响到另一项功能。

而且，由于所有的控制功能都在主控器中，因此简单的硬件变更起来是很容易的。这种简单方法的缺点是它受限于其工作范围。比如，假设您希望增设一个用于调节 LED 亮度的环境光传感器的时候，会发生什么？工作负荷将由主控器来承担，以把从动器加以“缝合”，并适应所有的设计变更，当发生根本性的改变时，主控器对面板的期望并未改变：它仍然只想知道哪些按钮被揿按，并控制 LED 的接通和关断。

为什么不设计定制 I²C 从动器件并卸除主控器的工作负荷？

这正是精明的设计师现在应该就想提出的问题，大多数有经验的计划管理员都能够提供基本的答案：因为它是一个“定制 I²C 从动器”，我们将无法按时/接进度地获得（因为没有过先例）。但是，情况未必如此，而且不应被视作一个常识。只需稍做安排并采取正确的方法，定制从动器件便可以实现正常交货。

如果您记得我们的策略，那么原始设计在哪方面未能满足要求？首先，所有的智能都包含在主控器中，而且从动器需要近乎恒定的关注。为了与现售解决方案相当，有一些特殊的器件解决了我们采用面板按钮和 LED 较好地加以定义的原始问题，但是，所有的逻辑信息都被存储于主控器中，而全部的变更都将通过主控器而在系统中产生波动。作为替代方案，我们可以定义一个面板界面，并将所有的细节交给一个分布式从动器系统来处理，因而可减少主处理器的干预。



问题仍然存在，就是如何在免受常见困扰的情况下设计定制器件？应采用定义了一个稳定协议和接口且经过验证的 I²C 从动器实现方案，最常用的协议是大多数现售 I²C 从动器件所采用的、基于寄存器的协议。如欲了解该协议的细节，您不妨研究一下 Philips 公司的 I²C 从动器件，不过这里我们提供了

相关的概要信息：

- 对于 I²C，所有的事务处理均由主控器来启动
- 每个从动器件都具有一个 I²C 7 位地址（最低有效位指示事务处理是“读”还是“写”）
- 每个从动器件都具有一个内部地址寄存器，用于维持至一个包含数据、命令或状态的内部“表格”的指针
- 每个从动器件均定义了其寄存器的地址及其功能，包括它们是“只读”型的还是“读/写”型的
- 一个写处理事务由一个具有 I²C 器件 7 位地址和写操作位的字节以及一个位于其后的、用于设定内部地址寄存器的字节组成，如果在处理事务中有任何更多的字节，则它们将被写入以新设定的内部地址为起点（从第二个字节）的从动器
- 一个读处理事务由一个具有 I²C 器件 7 位地址和读操作位的字节以及一个位于其后的主控器（用于记录来自其所期望的从动器的字节数目，并在记录完毕后提供一个“停止”信号）组成

正如您能看到的那样，I²C 从动器的魅力在于其与双端口 RAM 很相似，而且两者的易用性几无差别。

现在要选择的是可编程器件，虽然这将是一个具有非常强的个人色彩的选择，但是，关键因素是器件应具备在不进行再学习（即：耗时的学习曲线问题以及硬件和工具会导致一个项目陷入中断状态）的情况下支持众多需要的能力。赛普拉斯公司推出的微控制器型器件（名为“PSoC”）拥有了处理大多数 I²C 从动器所必需的全部功能，并提供了一种易用型工具，从而使得增加 I²C 从动器能力的工作简单到了“拖、放、选择地址”的地步。

首先，我们定义按钮输入从动器件，设计一个具有三个地址引脚（简易可升级性）和 7 个按钮输入的器件，并通过配置使之接受一个靠近系统电源（假设为 5V DC）的常开开关。如上所述，按钮状态将在一个可由一个 I²C 主控器进行存取的字节中提供。\

其次，我们定义用于控制 LED 的从动器件，设计一个具有两个地址引脚并能够以 10mA 的电流来驱动 8 个 LED（分为 4 红 4 绿）的器件。一个被称为“Command”的字节被定义为起 I²C 命令输入的作用，以控制 LED，该字节的 4 个低位（低效半位字节）用于控制红光 LED，而 4 个高位（高效半位字节）则负责控制绿光 LED。

文章到此可以结束了（我们已经从自己的原始实例中提供了现售器件的一种定制版本），或者，我们也可以继续下去，更加细致地定义定制 I^2C 主控器/从动器接口，并再次运用我们项目（或者我们公司的一批项目）的特性，以进一步实现从动级的客户化设计。如果我们想把系统主控器从一个轮询器件变为一个接收报警信号（例如，当任何按钮的状态发生改变时）的中断器件，则我们将希望给按钮输入设备增加一根输出线。这些改进能够进一步地把主系统处理器与低级细节隔离开来，并提供了在不影响主系统的情况下继续改善子系统的更大灵活性。

将“分而治之”的概念从 I^2C 推广到所有的通信总线

本文所讨论和说明的概念并非为任何特定的总线类型所特有。需要做的是定义满足您的策略（可能包括最大限度地缩短无线、电池式系统中的传输时间，抑或是在苛刻的工业环境中实现完善的检错/纠错）的协议。主控器可以被称为集线器（而不是启动器），从动器可被称为节点并具有一个预定的响应时间，但是，“分而治之”的思想仍然适用 —— 把普通、重复的测量和低级控制分配给最低的点，而将重要的工作留给系统控制器来完成。而且，在每种场合中都必需建立功能强大、精确定义的接口，以便为下一级的设计师留出一定的自由度，以便在不影响较高级元件的情况下变更实现方案。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com