

# 蓝牙低功耗数据泵的实现

作者 George Charkoftakis, Panagiotis Sifniadis

版本 1.0

日期 **30/10/2019**

Achilleos 8 and Lamprou Katsoni,  
17674, Kallithea, Athens  
希腊

电话: +30 210 93 10 580  
传真: +30 210 93 10 581

Neue Straße 95  
73230 Kirchheim/Teck  
德国

电话: +49 7021 805-0  
传真: +49 7021 805-100

## 蓝牙低功耗数据泵的实现

### 序言

本文介绍了蓝牙低功耗（BLE）的一些特定用例，称为数据泵的应用，在该类应用中，外部的 MCU 可向有 BLE 功能的设备推送和拉取数据。该类应用的关键要求是可通过低功耗蓝牙无线链路进行数据透传，而不关心数据的类型、配置文件（profile）或数据的最终应用。

在本文中，我们呈现了对接到蓝牙软件协议栈中不同层的各种可能性。数据泵架构的每种设计选择均有明确的优缺点，从纯技术的（如速度或功耗）到与业务相关的（包括成本、可维护性和上市时间）。围绕这些方面，我们将建议您如何根据不同的应用做出正确的选择。

在考虑特定用例的情况下，本文的其余部分介绍了 Dialog 的 DA14531 器件在实现数据泵应用时的适用性。DA14531 是一款成本优化的解决方案，可作为完全可配置的 BLE 主机控制器接口（HCI），或运行实现与数据无关的“无线串行”端口的应用等。

### 数据泵应用

图 1 显示了通用数据泵架构的简化图。它由两部分组成，支持 BLE 的子系统和上层应用的子系统。我们称前者为控制器，后者为主机。根据在每个子系统中实现的 BLE 协议栈的不同部分，可以进行几种配置，后文将对这些配置进行介绍。

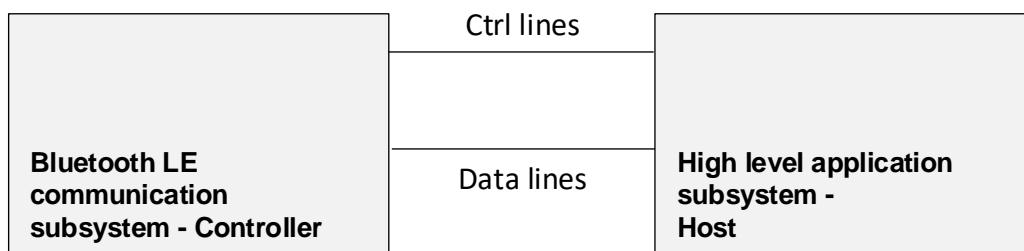


图 1 通用 BLE 数据泵架构

两个子系统共享数据线来交换信息，控制线来实现功能，如流控制、睡眠控制、事件指示、复位、甚至是对控制器的电源控制等。数据接口的实现通常是一套标准的串行协议，例如 UART、SPI 或 I<sup>2</sup>C。它可以遵循标准流程，比如 UART HCI 协议，也可以遵循定制流程，以节省引脚（I<sup>2</sup>C）或提高链接速度（SPI）。其余与控制相关的功能是使用其他通用输入/输出（GPIO）实现的。也可以将部分控制要求与数据线复用，或使用带内控制信令。正确的选择通常取决于控制器和主机中可用的 GPIO，以及可用的硬件接口及其功能。

图 1 所示的系统架构非常通用化。它存在于大多数嵌入式系统中，有经验的工程师都会比较了解，因此有助于将现有的非 BLE 系统迁移到蓝牙的世界。BLE 子系统成为应用子系统的外设，可以使用标准串行接口将其连接到应用上。BLE 的标准对其中一种设计选项进行了定义：HCI。这是数据泵架构中最强大的一种，为应用子系统提供了对 BLE 控制器的最大控制。但不应认为这一设计选项一定是最佳选择。使用 HCI 的学习曲线是不可忽略的，而对 BLE 的专业知识也至关重要。HCI 解决方案还会占用应用子系统大量内存和 CPU 资源。那么可以对其进行简化吗？可以，而且已经开发了多种不同的解决方案来解决这个问题。

在下一节中，我们将讨论使用数据泵构建系统时的需求和挑战。

### 主机和控制器之间的接口

UART 是连接 BLE 数据泵控制器及其主机的最常用接口。它具有全双工传输的优势，且仅需要最简单的两条线路。一般而言，两个设备均可以随时发起通信。可实现 921 kbps 数据速率。UART 接口被认为是实现 BLE HCI 和不同数据泵架构的标

## 蓝牙低功耗数据泵的实现

准，我们将在后文进行讨论。其中一个主要缺点是 UART 只在两个设备之间使用，并且无法支持总线架构。同时，也是一种昂贵的资源，因为系统中总是没有足够的 UART 接口。

另一个选择是 SPI。它也是全双工传输，但可以实现超过 20 Mbps 的更高的数据吞吐量，并且支持总线架构。它至少需要四个信号。但支持总线架构的功能也带来了一定的缺陷，即所有传输都由总线上的 SPI 主设备控制和启动。SPI 从设备无法启动事务，必须请求主设备进行。SPI 主设备准许 SPI 从设备的请求并读取数据。因此需要具有信号交换和优先级的协议。通过这样的协议处理全双工通信是很复杂的，通常实际的数据传输是半双工的，这意味着两个设备中只有一个会在给定的时间点通过 SPI 链路传输有用的信息。

第三个常用的同步通信接口是 I<sup>2</sup>C。其主要优点是它仅需要两个信号即可在通信设备之间共享，并且可以形成带内寻址总线。最大数据速率从 100 kbps 到 1 Mbps 不等。I<sup>2</sup>C 绝对是半双工的，并且具有与 SPI 相同的缺点，即从设备无法启动传输，因此需要信号交换协议。

### 低功耗管理

靠电池供电的应用需要优化功耗。对功耗敏感的应用要求系统的两个部分（主机和控制器）都可以尽可能地进入低功耗状态，通常称为睡眠状态。BLE 数据泵的应用场景也不例外，且必须不间断地支持睡眠管理。在主机/控制器交换数据的情况下，控制器必须能够唤醒并通知主机有需要处理的数据，反之亦然。控制器和主机都应能够缓存少量数据，以补偿唤醒对方所花费的时间。长时间处于休止状态的系统可能还需要在不使用通信接口时，将控制器置于非常低的功耗模式，甚至使控制器断电。睡眠管理通过专用的 GPIO 引脚实现，或与流控制或数据线引脚复用。

### 带宽

数据泵架构可用于满足不同的数据吞吐量要求：从偶尔传输小数据块到连续传输大量数据。数据泵必须能够处理所有这些情况。在连续传输大量数据的情况下，应使用专用的二进制模式，从而允许通过共用接口以尽可能高的速率透传原始数据，而没有因为 header 和其它带内信号字段带来的延迟和开销。

另一方面，对于零星的数据传输，所使用的协议可以支持任何带有 header 的通知机制，以将数据作为消息的一部分进行传输。在这种情况下，信号交换机制和 header 只会对系统带来最小的负担。

### 电源选择

在上文中，我们简要讨论了功耗和睡眠管理。设计人员在选择控制器时还必须考虑电源本身，尤其是当打算将其集成到现有设计中时。现有的主机子系统可通过多种能源供电，包括具有不同额定电压和容量的电池、太阳能电池板或电网。控制器应能够使用现代微控制器子系统中常用的各种电压工作。

### 其它要求

另外，还需考虑其他几个参数。下文列出的是我们已知的可能影响数据泵控制器选择的一些要求。

- **尺寸**：嵌入式系统通常具有很大的尺寸限制。现代 BLE 的数据泵控制器的体积小且组件少，即使在空间极度受限的设计中也可以轻松实现接口。
- **Boot 延迟**：引导时间是指控制器在复位后可被外部主机访问所需的时间。它可能直接影响整体功耗和响应能力。
- **传输输出功率控制**：这是微调功耗或实现近场模式功能所必需的。
- **最大输出功率**：提高射频链路吞吐量，尤其是在进行流传输时。
- **成本**：成本始终是重要的考量因素。数据泵控制器的总成本越低，现有未与外部连接的应用接入连接的可能性就越大

## 蓝牙低功耗数据泵的实现

### 通过 BLE 实现数据泵

在介绍 BLE 数据泵架构的不同形式之前，让我们回顾一下蓝牙协议栈架构（图 2）。BLE 协议栈由几层组成。底层包括射频、基带和链路层；处理射频操作，射频事件调度，连接的建立、管理和数据传输。

用于对接协议栈底层和上层的是标准的 HCI 协议。该标准允许用户以最小的改动置换主机协议栈或控制器 IC

在 HCI 上方，信息分为两条不同的路径。

- a) 橙色路径主要涉及数据驱动的操作，包括 L2CAP、ATT 和 GATT 层。它还可能包括位于 GATT 层之上的标准配置文件和服务的实现。
- b) 蓝色路径主要与连接控制和射频操作有关。GAPM 层处理大多数此类操作，而 SM 层主要负责与安全性相关的操作。

对应用层的接口方面，有可用的 GAP 层 API 可供控制，有可用的标准配置文件的服务或可直接调用的 GATT 层 API 可供数据传输用。

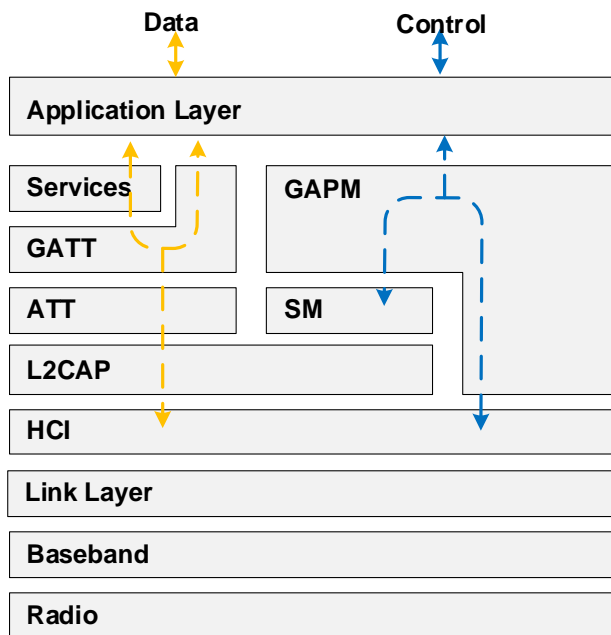


图2 BLE 协议栈

### 数据泵的变化形式

图 3 显示了 4 种 BLE 协议栈的划分方法，不同的划分对应不同的主机-控制器组合配置。在每种情况下，切线（Cx）将 BLE 协议栈分为下部（控制器）和上部（主机）。数据泵对经过切线的消息进行序列化，并将它们传递给通常位于外部子系统的主机。BLE 协议栈中缺少的层的功能必须在主机设备中实现。

表 1 总结了数据泵架构的不同形式，以及一些基本属性和所使用的协议。

## 蓝牙低功耗数据泵的实现

表 1 数据泵架构的不同形式

切线	通信协议	对控制器的控制	备注
C1	HCI	完全	标准 HCI 主机-控制器的实现
C2	DGTL	完全	Dialog GTL 主机-控制器的实现。将所有协议栈的消息序列化之后给到应用层，反之亦然。
C3	DGTL – 混合	选定的	Dialog GTL 主机-控制器的实现，可以选择哪个应用层的功能在控制器中实现，以及哪个应用层的功能向主机公开。
C4	应用定义的协议	有限的	控制器运行完整的 BLE 应用，并给主机提供有限的（如 CodeLess AT 工程）控制或无控制（如 DSPS 工程）；并提供一条数据管道用于数据透传。

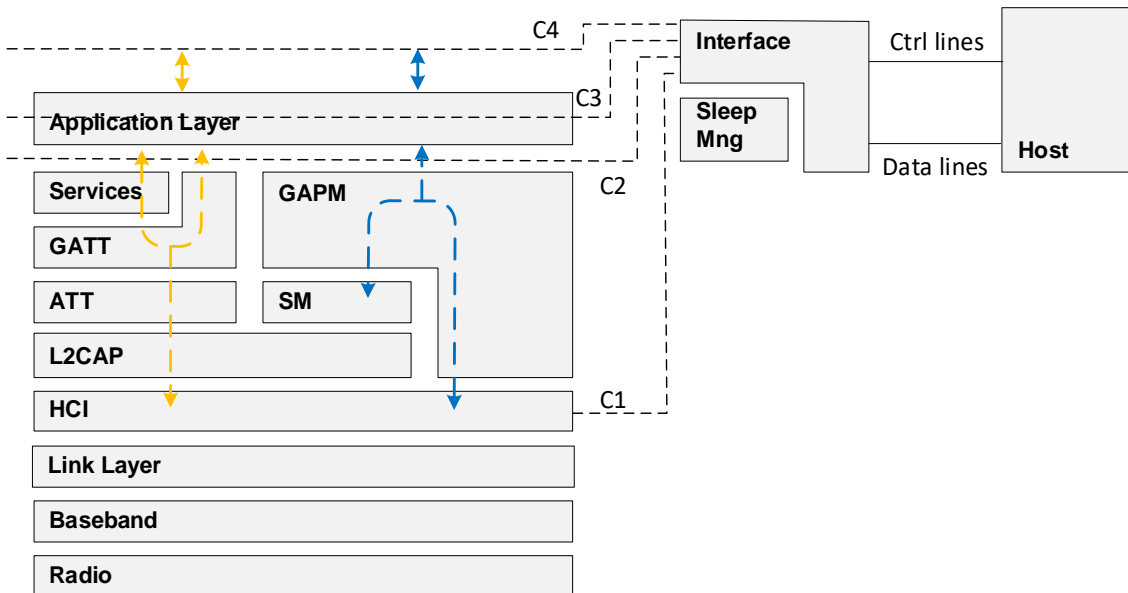


图 3 可能的控制器-主机配置

### HCI 主机-控制器架构

上文中 C1 产生了 HCI 配置。这是蓝牙技术联盟的一个标准协议，受到诸多供应商的支持。Dialog 除了提供标准的 UART 接口的 HCI 实现之外，还提供其它经过验证的解决方案——可通过专用的协议将 HCI 消息序列化之后用 I<sup>2</sup>C 或 SPI 接口传输。

HCI 控制器的实现系基于蓝牙技术联盟认证的一个控制器。这使开发人员可以从众多供应商中选择一个 BLE 主机协议栈，并且仍然可以确保高度的互操作性。

主机可以不受限制地完全控制控制器的功能。控制器支持的任何功能都可以被调用和控制。

这种方法存在一些缺点。例如，BLE 的上层或类似功能必须在主机中实现。这通常意味着实现经认证的主机协议栈会产生额外成本，并且需要具备与 BLE 有关的专业知识。但是蓝牙认证的控制器仍然可以确保最小的互操作性。

该架构也不是功率最优化的。控制器需要进行唤醒以服务射频事件，并与 BLE 协议栈的上层交换信息，以操作应用事件和与应用无关的事件。这意味着主机必须更频繁地被唤醒，从而消耗更多功率。

## 蓝牙低功耗数据泵的实现

蓝牙上层的存储资源也不容忽视。代码可能需要 50 Kbytes，数据可能需要 10 Kbytes。

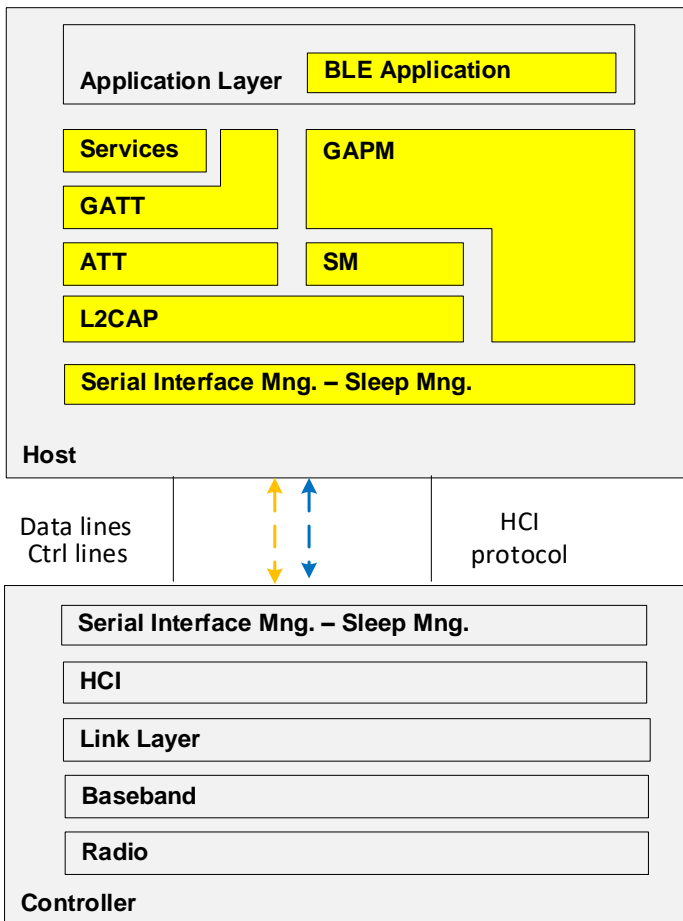


图 4 HCI 主机-控制器架构

### GTL 主机-控制器架构

GTL 架构在控制器中包含了完整的蓝牙协议栈。GTL 作为一项专用协议，可通过串行链路交换所有与应用程序相关的消息。使用的接口可以是 UART、I<sup>2</sup>C 或 SPI。GTL 按照与 HCI 协议类似的原理进行操作。它提供了一个针对 BLE 应用的所有事件和命令的接口。GTL 比 HCI 更省电，因为外部主机只应处理与应用有关的信息。

此外，由于在控制器中实现了蓝牙协议栈，因此该架构用户只需要为外部主机开发更少的代码。但它依然很复杂，需要对控制器供应商协议栈的所有操作有充分的了解。

Dialog 自己的 GTL 方法（即 DGTL 数据泵架构）非常通用，它有两种形式。纯版本（图 3 中的 C2）遵循标准 GTL 方法。同时，在 DGTL-混合（图 3 中的 C3）中，设计人员可以将部分应用代码移至控制器中，并将部分功能开放给外部主机。这提供了进一步优化功耗并降低主机软件复杂性的潜力。一个典型的应用是，可选择将服务的实现放在系统的哪一部分中（主机或控制器）。包含静态信息（如设备 ID）的服务，最好在控制器上实现，这样就无需每次在远程用户希望从该服务上读取信息时都唤醒外部主机。相反，可以在外部主机中实现包含动态信息的服务，以简化信息的验证，并减少通过串行接口交换的数据量。

## 蓝牙低功耗数据泵的实现

特别是对于新应用，DGTL 比 HCI 更好。它不仅更加节能，而且由于主机和控制器之间需要交换的信息较少，因此可以实现更高的吞吐量。此外，它提供了对协议栈的完全控制，消除了互操作性的问题，因为该规格来自供应商的控制器实现，并且不需要额外的成本来为主机获取蓝牙软件组件。

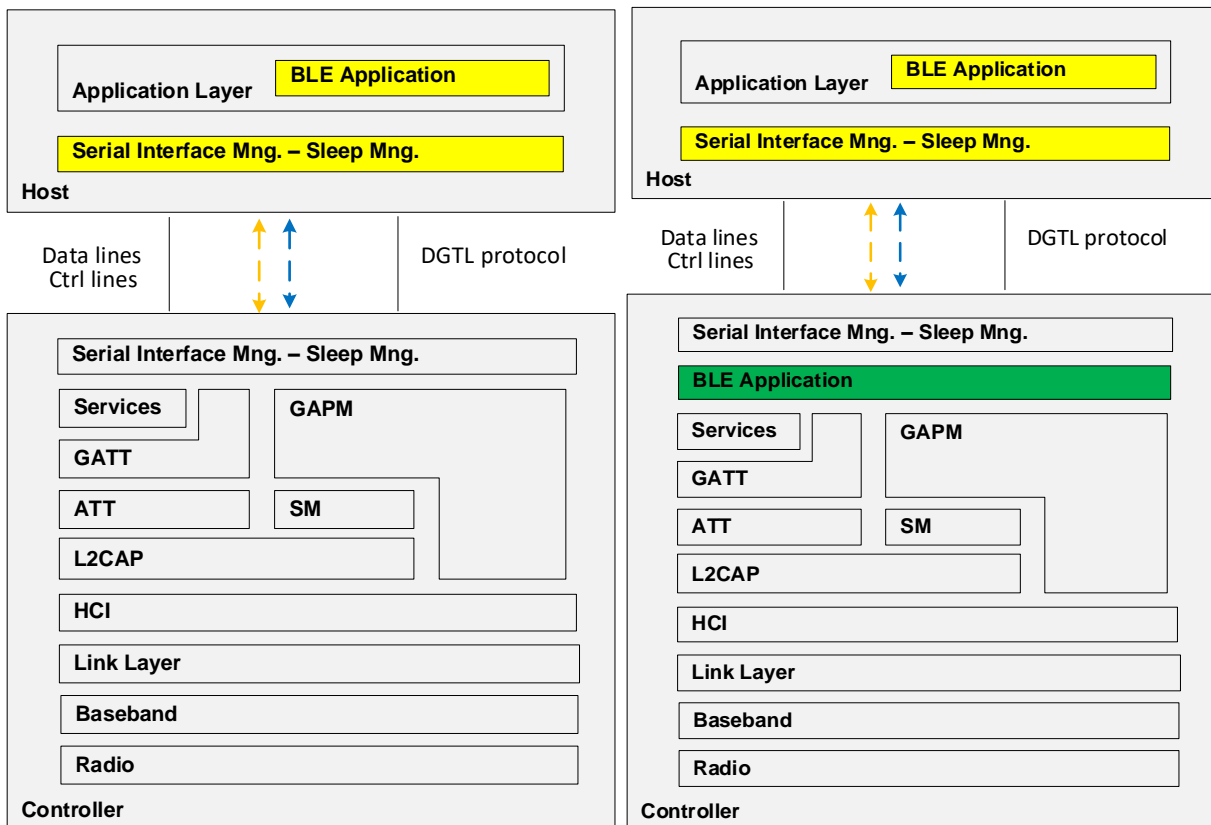


图5 DGTL (左) 和 DGTL-混合 (右) 架构。注：在 DGTL-混合情况下，部分 BLE 应用位于控制器内 (以绿色显示)

但也存在一些缺点。DGTL 将外部主机与控制器的供应商紧密耦合。使用此架构需要很好的蓝牙技术，并且对控制器中的实现方式有很好的了解。

### 应用定义的协议 – DSPS / CodeLess AT

图 3 中最上层的切线 (C4) 涉及在控制器中实现一个完整的 BLE 应用。该应用处理所有蓝牙相关的功能，并通过自定义协议将经过协议栈的特定的数据、命令和事件公开给外部主机。

### DSPS 主机-控制器架构

该架构的一种应用，是 Dialog 的串口服务 (DSPS，充当 UART 串口线) 参考设计。参考设计中提供了两种实现方式：一种用于中央设备，一种用于外围设备。在现实生活中，中央设备通常是安卓/iOS 手机。

蓝牙低功耗数据泵的实现

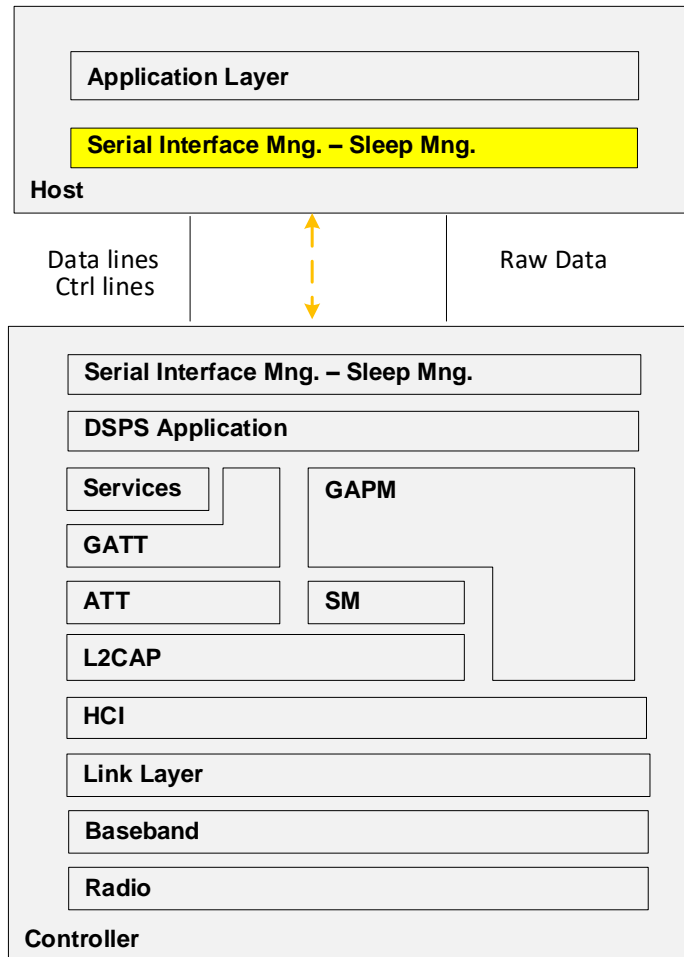


图6 DSPS 主机-控制器架构

DSPS 在设计时考虑了简易性和性能。对蓝牙链路无控制。连接参数是硬编码的，且连接由控制器处理。在使用安卓/iOS DSPS 软件时，终端用户可以选择要连接的 DSPS 设备。可以将 DSPS 视为串行端口的替代，或将串行化的数据传输到主设备或从主设备传输串行化数据的通用方法。DSPS 应用的核心是一套定制的 BLE 服务，仅使用三项特征。其中两个用于 RX/TX，另一个用于流控制。对于控制器和主机之间的串行接口，当两端都支持该特性时，可以通过 RTS 和 CTS 信号的标准流控制实现。

利用内部循环缓冲区、DMA 和流控制机制对 DSPS 进行了优化，最大程度地提高数据吞吐量。对于特定设置，吞吐量可以超过 600 Kbit/秒。DSPS 是功率最优化的解决方案之一。控制器和主机仅需在与数据相关的应用活动中相互唤醒。所有其他操作均由控制器处理。



## 蓝牙低功耗数据泵的实现

Host		Android	iOS	DA14585	
Connection parameters	MTU	octets	>130	>130	
	Connection Interval	ms	12.5	30	
	Host max. write command size	B	128	128	
Throughput measurements	<b>Half Duplex</b>				
	Central Tx	packets/conn event	2	4	2
		kB/s	11.0	11.6	11.2
		kbit/s	88.0	92.8	89.6
	Peripheral Tx	packets/conn event	2	5	2
		kB/s	11.5	10.0	11.2
		kbit/s	92.0	80.3	89.6
	<b>Full Duplex</b>				
	Central Tx	packets/conn event	3	2	3
		kB/s	10.9	7.1	11.2
		kbit/s	87.4	56.8	89.6
	Peripheral Tx	packets/conn event	3	2	3
		kB/s	11.5	9.5	11.2
		kbit/s	92.0	76.0	89.6

图 7: 典型 DSPTS 性能表

作为一个数据泵，DSPTS 架构可以在主机干预最少的情况下传输大量数据。

那么对蓝牙链路的控制情况如何呢？当应用需要控制连接及其参数时，DSPTS 并不是一个好选择。当然，DSPTS 实现是在源代码中提供的，终端用户可以通过修改代码以编译自定义 DSPTS 来优化连接参数并增加对连结的控制。这需要具备 DSPTS 软件知识，了解其内部架构以及在控制器环境中进行开发的能力。

当现有的 DSPTS 功能完全满足应用需求时，与 DSPTS 类似的数据泵是想要连接智能手机的设备之完美选择。几乎不需要具备蓝牙知识即可使用控制器。主机的代码开发简化为处理用于数据传输的串行接口，并且可以毫不费力地建立自管理的虚拟 UART 无线连结。但是，如果某些应用要求得不到满足，设计人员将面临要么投资定制 DSPTS 解决方案，要么选择另一种能提供更大灵活性的架构的困境。

### CodeLess AT 命令数据泵

与 DSPTS 数据泵类似的另一种应用是 CodeLess AT 命令数据泵。CodeLess AT 命令应用不仅提供数据管道，而且通过蓝牙连接提供了有限的控制路径，因此与 DSPTS 相比，它适合于更多的用例。

## 蓝牙低功耗数据泵的实现

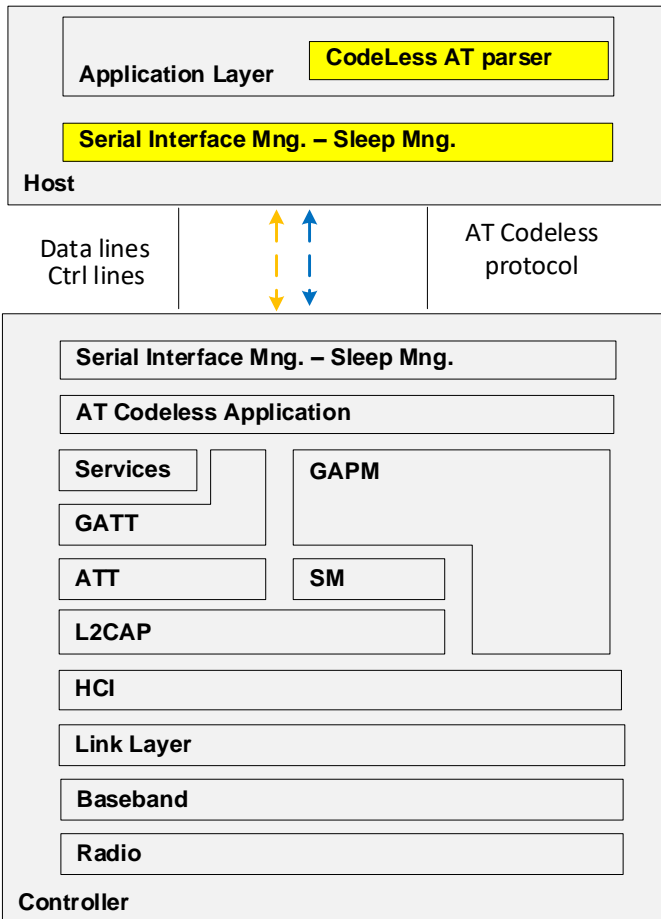


图 8 AT Codeless 主机-控制器应用

与前面描述的 HCI 和 DGTL 数据泵解决方案相比，CodeLess AT 命令的主机实现明显更简单，并且在 BLE 协议中对使用者的知识能力要求很低。因此，基于这种方法的解决方案可以大大缩短具备 BLE 连接功能的产品上市时间。

Dialog 半导体公司推出了 CodeLess AT 命令解决方案。这类方案在市场上经常可以找到。顾名思义，这些解决方案是基于 AT 命令的，并试图向用户隐藏 BLE 实现细节。AT 命令是简短、易于理解的文本命令，可以通过任何可用的通信接口（UART, SPI, I<sup>2</sup>C）发出。每个命令都指示 CodeLess AT 控制器执行一个或多个预定义功能。根据功能产生的结果，CodeLess AT 控制器会给出适当的响应。

典型的 CodeLess AT 命令支持扫描其他 BLE 设备，建立新连接和基本连接管理。主机开发的基础是在适当的时候发出 AT 命令，并响应错误代码以及作为数据源或数据接收器。

与更传统的与控制器交互的方式相比，此方法具有多个优点。在“主机-控制器”链路上传输的主要是应用层相关的消息，功耗由控制器管理，因此它比较节能。此外，CodeLess AT 还提供了简化的 BLE 连接管理和数据传输控制。因此，用户无需熟悉 BLE 底层的细节。

在主机中开发最终应用所需的工作最少。AT 命令很容易记住，所有消息均采用用户可读的形式，没有任何含义模糊的代码和晦涩的信息，设计人员因此可以直接与控制器进行交互或实验。

CodeLess AT 命令应用可被用于 BLE 的入门，降低 BLE 的学习门槛。用户可在深入钻研更复杂的解决方案之前，自己通过该应用熟悉一些 BLE 的概念。因此，该应用能显著加快开发进度。

## 蓝牙低功耗数据泵的实现

这种方法也存在一些缺点。CodeLess AT 在数据吞吐率上没有作优化。尽管功能强大，但 CodeLess AT 命令应用仅支持有限的一些 BLE 协议栈功能。因此，在开发过程中，设计人员可能再次面临困境：要么扩展 CodeLess AT 控制器功能，要么采用提供完全灵活性的其他解决方案。

### 数据泵应用中的 DA14531

#### DA14531 关键特性

DA14531 是 DA145xx 系列 BLE 产品中的最新成员。该设备在成本和功耗上都经过了优化，主要应用于简单的、一次性及数据泵类的产品。它可用作本文中介绍的所有数据泵架构，作为 HCI、GTL、DSPS 或 CodeLess AT 控制器。

#### 满足数据泵的要求

如上所述，数据泵应用的关键要求包括数据吞吐率、主机与控制器之间的接口选项、低功耗、供电的灵活性、小尺寸和低启动延迟。DA14531 从一开始就被设计用来满足数据泵控制器的基本要求。

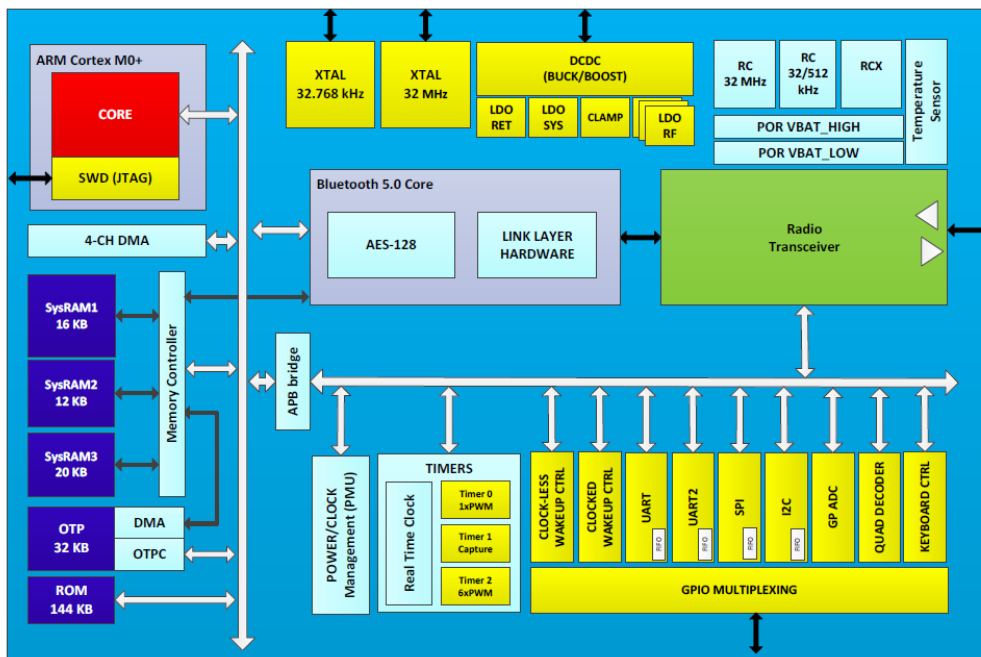


图9 DA14531 框图

除了符合蓝牙 5.1 规范外，它还提供全部三种类型的串行接口（UART，SPI 和 I<sup>2</sup>C），从而允许主机子系统中使用任何类型的串行通信。DA14531 的功耗得到了优化，这得益于其可配置的发射功率，和其在 Buck 模式(3.3V 供电)下低于 2.2mA 的射频接收电流。

电源管理电路围绕两条电轨（VBAT\_HIGH 和 VBAT\_LOW）构建。在两个电轨之间，有一个 DC / DC 转换器，一个 LDO 和一个内部开关。该设置允许 DA14531 在主机子系统中可用的任何电压下工作。例如，如果电压低于 1.8 V，则可以在 VBAT\_LOW 处提供该电压，并使用内部 DC / DC 转换器将其升压至 VBAT\_HIGH。可以向 VBAT\_HIGH 提供 1.8 - 3.3 V 的电压。在这种情况下存在多种选择。可以使用内部 DC / DC 转换器将电压降压至 VBAT\_LOW，以优化功耗。对于降低成本的解

## 蓝牙低功耗数据泵的实现

决方案，两个电轨都可以在外部进行缩短以节省 DC-DC 电感器的成本。或者，可以使用内部 LDO 或开关代替降压 DC / DC 配置，以将 VBAT\_HIGH 的操作扩展到 1.8 V 以下，以消耗电池电量。

DA14531 提供多种封装选择。FCGQFN24 封装尺寸为 2.2 mm x 3.0 mm，并提供 12 个可用的 GPIO。WLCSP17 封装尺寸仅为 1.694 mm x 2.032 mm，但仍具有 6 个可用的 GPIO，从而最大程度地减少了实现 BLE 连接所需的占板尺寸。

如果从外部闪存读取代码，则设备冷启动将在不到 30 毫秒的时间内完全运行。如果代码存储在内部 OTP 存储器中，则启动时间将降至 1.5 毫秒。

## 应用案例

### 智能白色家电

大多数中高端白色家电已经包含了一个微控制器单元，以实现一些高端功能，例如精细控件和高级用户界面。但由于成本和复杂性，仍缺少无线连接功能。将无线连接功能引入此类设备将带来一系列新的可能性，使消费者可以通过智能手机与家电进行交互。简单的应用场景包括交换宝贵的使用情况统计数据（如工作时间和功耗），以及服务或错误代码日志以远程诊断问题。此外，可以通过智能手机提供的功能强大的接口来配置家电，或进行个性化设置。在这种情况下，BLE 的安全性将成为重要考量因素。Dialog 提供了有助于此类应用开发的工具（SDK、参考设计）。

由于目前还没有为大多数白色家电定义 BLE 的标准服务，因此类似 DSPTS 的数据泵是为这些设备提供连接的绝佳起点。不但没有互操作性要求，而且在 DSPTS 提供的无线串行接口上唯一需要开发的是一个专有的命令响应协议，能够维持无线连接中可能发生的突然断连。仅通过该协议的实现，该家电就可以交换信息和命令，而无需花费时间做出与 BLE 相关的决定。

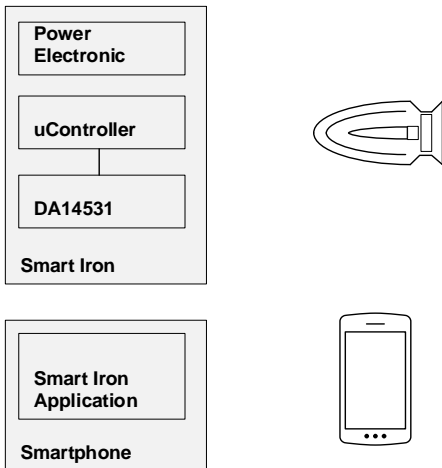


图 10 “智能”熨斗用例

家电在通电时会向外界告知其自身的存在，并能够连接到附近的智能手机。人们很容易联想到智能洗碗机或智能电炉，但 DA14531 的成本和尺寸使其甚至可以为小体积和成本敏感的应用提供连接功能，如智能熨斗或智能电吹风。许多已经具备微控制器并可以省去几个引脚的小型设备，可以通过连接到智能手机来实现各种功能。想象一下，Alexa (亚马逊的语音智能助手) 可以从互联网的数据库中，选择最适合你的连衣裙的熨烫机的配置。

## 蓝牙低功耗数据泵的实现

### 电池供电的工具

无线连接功能可以为所有小型设备带来一系列类似的好处。但对于电池供电的设备（例如无绳电钻）而言，则需要对蓝牙链路有更多控制。为了节省电量，设备不应始终向外界告知自己的存在。同时，设备应该在用户不使用的情况下，自动断开连接。使用 CodeLess AT 命令可以轻松实现此类功能。CodeLess AT 命令的应用可以提供足够的控制和信息，以了解 BLE 的连接状态。

### 选择正确的架构

如果已经定义了与特定用例相关的 BLE 服务，则 GTL 架构或许是最好的方法。这将使服务能够开放给其他可互操作的应用，甚至允许目标设备访问智能手机提供的服务。

温度计、血糖仪、血压计、体重秤都是可以从 BLE 的标准服务中受益的例子。GTL 架构通过串口提供了蓝牙协议栈的所有灵活性，但会增加主机上蓝牙应用的开发成本。一个简单应用的代码占用量为 5-10 Kbits。

综上所述，下表列示了所有通信协议的优缺点。

表 2 – 各通信协议的优缺点

通信协议	优点	缺点
HCI	<ul style="list-style-type: none"> <li>众多供应商支持的标准接口</li> <li>高性能（吞吐量）</li> <li>对蓝牙有完全控制</li> </ul>	<ul style="list-style-type: none"> <li>需要具备高深的 BLE 的知识</li> <li>需要 BLE 协议栈，这意味着额外的费用</li> <li>极大的复杂性</li> </ul>
GTL	<ul style="list-style-type: none"> <li>对蓝牙有完全控制</li> <li>节能</li> </ul>	<ul style="list-style-type: none"> <li>非标准</li> <li>比 HCI 简单，但仍很复杂</li> <li>需要有蓝牙知识</li> </ul>
DGTL/DGTL-混合	<ul style="list-style-type: none"> <li>可以选择在控制器中实现哪些应用层功能以及向主机公开的功能</li> <li>对蓝牙有完全控制</li> <li>节能</li> </ul>	<ul style="list-style-type: none"> <li>非标准</li> <li>比 HCI 简单，但仍很复杂</li> <li>需要有蓝牙知识</li> </ul>
DSPS / AT CodeLess	<ul style="list-style-type: none"> <li>无需事先具备 BLE 知识</li> <li>使用简单-所需开发最少</li> <li>快速上市</li> <li>Dialog 提供源代码</li> </ul>	<ul style="list-style-type: none"> <li>对蓝牙链路有限控制（AT CodeLess）或无控制（DSPS）</li> <li>难以添加新功能，因为可能需要重新编译</li> </ul>

### 结语

本文介绍了不同的 BLE 数据泵架构。特别研究了不同的数据泵架构，从需要大量专业知识和蓝牙知识的解决方案，到简单得多的解决方案。精心设计的解决方案无疑对数据透传过程提供了更好的控制，而简单的解决方案则允许以最小的工作量和在较短的时间内创建应用。除了 BLE 数据泵外，文章还介绍了 Dialog 半导体的 DA14531，非常适合数据泵应用。文章在最后还提供了几个应用案例，以及为每个应用案例选择最合适的数据泵架构的指南。

## 蓝牙低功耗数据泵的实现

### 术语与定义

ATT	属性配置文件
BLE	蓝牙低功耗
DSPS	Dialog 串口服务
GATT	通用属性配置文件
GPIO	通用输入/输出
GTL	DA14531 原生支持的类 GAP 协议
HCI	主机控制器接口
I <sup>2</sup> C	集成电路总线
L2CAP	逻辑链路控制与适配协议
LDO	低压差
MTU	最大传输单元
OTP	一次性可编程的
SPI	串行外设接口
UART	通用异步收发器

### 参考资料

- [1] DA14531, Dialog 半导体数据手册
- [2] 蓝牙技术联盟, 蓝牙 v5.0 核心规范