



Shiomi Itzhak

## 摘要

CC33xx 系列器件是新一代 Simplelink™ 嵌入式解决方案。这些器件的主要作用是满足新兴物联网 (IoT) 用例的要求，同时与最新的尖端技术 (如 Wi-Fi® 6 和低功耗 Bluetooth® 5.3) 兼容。

在主机处理器运行 Linux® 或 MCU 主机运行 RTOS 的嵌入式应用中，这些新一代器件可实现经济、可靠且安全的连接。为实现与各种主机处理器进行连接，主机接口设计必须稳健可靠，并提供可供设计人员选择的灵活配置。

## 内容

1 引言.....	3
2 主要特性.....	3
2.1 双主机接口.....	4
2.2 共享主机接口.....	4
2.3 自主模式.....	5
2.4 主机中断.....	5
3 接口.....	5
3.1 引言.....	5
3.2 SDIO 接口.....	5
3.3 SPI 接口.....	6
3.4 UART 接口.....	9
3.5 引脚数选项.....	10
4 主机通信.....	10
4.1 协议概述.....	10
4.2 SDIO 包装器.....	12
4.3 SPI 包装器.....	13
5 启动流程.....	14
5.1 SDIO.....	15
5.2 SPI.....	15

## 插图清单

图 2-1. CC33xx 双接口.....	4
图 2-2. CC33xx 共享接口.....	4
图 3-1. 16 位，大端字节序，写入方向.....	8
图 3-2. 32 位，大端字节序，写入方向.....	8
图 3-3. 16 位，大端字节序，读取方向.....	8
图 3-4. 32 位，大端字节序，读取方向.....	8
图 3-5. UART 接口流控制.....	10
图 4-1. 通用命令.....	10
图 4-2. Send 或 Sendto 命令.....	11
图 4-3. Recv 或 Recvfrom 命令.....	11
图 4-4. Read_status 命令.....	11
图 4-5. Cnys 命令.....	11
图 4-6. SDIO 包装器.....	12
图 4-7. SPI 包装器.....	13
图 5-1. 主接口初始化.....	14
图 5-2. 辅助接口初始化.....	15

图 5-3. SDIO CMD0 命令.....	16
--------------------------	----

## 表格清单

表 2-1. CC33xx 主要功能.....	3
表 3-1. SDIO 接口特征.....	5
表 3-2. SDIO 接口信号.....	5
表 3-3. SPI 接口特征.....	6
表 3-4. SPI 接口信号.....	6
表 3-5. SPI 接口格式配置.....	7
表 3-6. SPI 接口配置汇总.....	7
表 3-7. UART 接口特征.....	9
表 3-8. UART 接口信号.....	9
表 3-9. UART 接口配置.....	9
表 3-10. 引脚数选项.....	10
表 4-1. 协议结构.....	12

## 商标

Simplelink™ is a trademark of Texas Instruments.

Wi-Fi® is a registered trademark of Wi-Fi Alliance.

Bluetooth® is a registered trademark of Bluetooth Sig, Inc.

Linux® is a registered trademark of Linux Foundation.

is a registered trademark of Linus Torvalds in the U.S. and other countries.

所有商标均为其各自所有者的财产。

## 1 引言

本文档旨在描述主机处理器和 CC33xx 配套 IC 之间的主机接口，并为系统设计人员提供轻松集成所需的全部技术信息。

连接到 CC33xx 配套 IC 的常见主机处理器是通常运行 Linux 等高级操作系统的高端处理器。此外，还可以使用中端或高端微控制器，通常通过 SPI 接口进行通信，提供完全的可扩展性和灵活性。Linux 操作系统（以及选定的 MCU）上的主机驱动程序实现也作为解决方案的一部分提供，使系统设计人员不必深入研究所有技术细节，从而更容易实现。

以下几部分介绍了主要功能、各种接口选项，以及主机器件通信流程和协议。

## 2 主要特性

表 2-1 列出了 CC33xx 配套 IC 支持的主要功能。

**表 2-1. CC33xx 主要功能**

功能	说明
Wi-Fi 主机接口	SDIO 或 SPI。 有关详细信息，请参阅节 2.1。
BLE 主机接口	<ul style="list-style-type: none"> <li>• UART</li> <li>• SDIO 或 SPI (共享接口)</li> </ul> 有关详细信息，请参阅节 2.1。
用于 Wi-Fi 和低功耗蓝牙的共享主机接口	通过 SDIO 或 SPI 节省硬件引脚。 有关详细信息，请参阅节 2.2。
自动主机接口检测	是
自主模式	在不通知主机的情况下进入睡眠状态和唤醒。通过 UART、SPI 和 SDIO 实现。 有关详细信息，请参阅节 2.3。
安全主机接口	预计在后期阶段提供
带外中断	<ul style="list-style-type: none"> <li>• 单独 - 一条线路用于 Wi-Fi，一条线路用于低功耗蓝牙</li> <li>• 共享 - 一条线路同时用于 Wi-Fi 和低功耗蓝牙</li> </ul> 有关详细信息，请参阅节 2.4。
带内中断	在 SDIO 模式中。 有关详细信息，请参阅节 2.4。

## 2.1 双主机接口

双主机接口是常见的配置，其中每个 IP 都使用自己的接口。Wi-Fi 可以使用 SDIO 或 SPI ( 使用相同的硬件线路 )，低功耗蓝牙使用 UART 线路。就硬件引脚而言，双接口模式会造成极大浪费。如果 SDIO 接口用于 Wi-Fi，则最多使用 10 条线路，其中 6 条用于 SDIO，4 条用于 UART。如果 SPI 接口用于 Wi-Fi，则最多使用 8 条线路，其中 4 条用于 SPI，4 条用于 UART。此计算不考虑额外的中断线路。表 3-10 中列出了包含所有引脚数选项的完整表格。

图 2-1 展示了双主机接口设置。

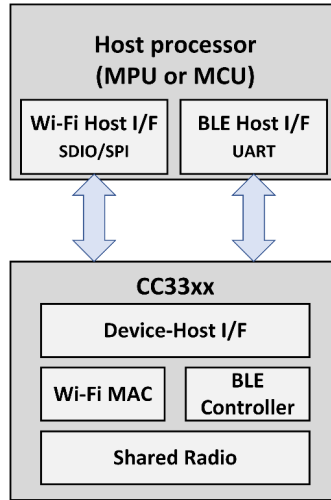


图 2-1. CC33xx 双接口

## 2.2 共享主机接口

在共享主机接口中，Wi-Fi 和低功耗蓝牙共享同一接口，该接口可以是 SDIO，也可以是 SPI ( 使用相同的硬件线路 )。使用共享接口的主要优势是减少了硬件引脚数量，节省的引脚可用于其他用途。如果使用 SDIO 接口，则最多使用 6 条线路；如果使用 SPI 接口，则最多使用 4 条线路。此计算不考虑额外的中断线路。表 3-10 中列出了包含所有引脚数选项的完整表格。

节 2.2 展示了双主机接口设置。

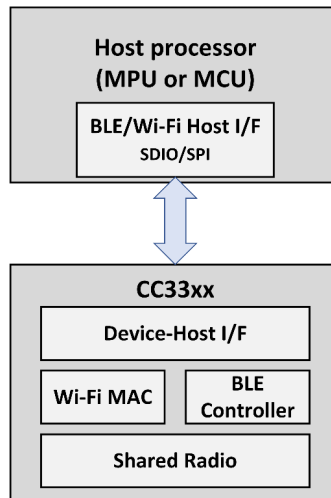


图 2-2. CC33xx 共享接口

## 2.3 自主模式

在自主模式下，器件可在不通知主机处理器的情况下进入睡眠状态和唤醒。所有接口均支持该模式。自主模式使用始终打开并准备好接收数据的内部缓冲区，即使器件处于睡眠状态也是如此。这是强制性的，因此器件不会丢失数据。在睡眠期间接收数据时，器件利用流控制机制停止来自主机处理器的数据，直到器件完全唤醒。器件在此模式下与主机处理器不间断连接，因此，看起来始终处于唤醒状态。自主模式是一种实时机制，甚至可以在非连续帧之间触发，从而在功耗方面具有出色优势。

## 2.4 主机中断

### 2.4.1 带外中断

器件使用中断线路来通知主机处理器存在挂起的事件或数据。还支持使用单独线路的带外中断，并且这是推荐的选项。推荐使用带外中断的原因主要是为了保持与某些主机处理器的兼容性，因为有些主机处理器不会在带内中断时唤醒。

在共享传输模式下，一条通用硬件线路可同时用于 Wi-Fi 和 BLE，从而节省一个引脚。然后，主机处理器必须读取寄存器来确定设置的 IP 源。

### 2.4.2 带内中断

如果主机处理器缺少 GPIO 或没有可用的 GPIO，也支持 SDIO 带内中断。带内中断是指无需使用专用线路即可通过其中一条数据线向主机发出信号的能力。例如，在 4 位 SDIO 模式下，第二条数据线 DAT[1] 用作带内中断。

## 3 接口

### 3.1 引言

主机接口包括三种类型的协议：SDIO、SPI 和 UART，并支持与两个 IP (Wi-Fi 控制器和低功耗蓝牙控制器) 进行通信。只要满足本段中描述的所有要求，器件就可以连接到运行 Linux® 的处理器主机或运行 RTOS 的 MCU 主机。

### 3.2 SDIO 接口

#### 3.2.1 SDIO 概述

表 3-1 列出了 SDIO 主机接口的特征。

表 3-1. SDIO 接口特征

特性	说明
SDIO 版本	高速 SDIOv2 和 SDIOv3 的一些实现
最大频率	50MHz
数据总线	4 位 SDIO ( 也支持 1 位 SDIO )
最大波特率	200Mbps ( 50MHz 时为 4 位 )
I/O 电压电平	1.8V

表 3-2 列出了 SDIO 主机接口的信号。

表 3-2. SDIO 接口信号

信号名称	吸电流	说明
CLK	控制器	串行位时钟
CMD	控制器	命令行
DAT0-DAT3	控制器/外设	数据线路

#### 3.2.2 SDIO 流控制

SDIO 流控制属于规范的一部分。SDIO DAT0 数据线路用于实现对主机控制器的 BUSY 信号。主机控制器会将该线路的有效 ( 将引脚拉至低电平 ) 状态解读为器件繁忙，进而等待执行其写入/读取操作。

与自主模式一样，流控制是一种实时机制，甚至可以在 SDIO 块之间触发，从而在功耗方面具有出色优势。

此方法还用于实现自主模式，以指示外设何时准备就绪以及何时未准备就绪。

### 3.3 SPI 接口

#### 3.3.1 SPI 概述

表 3-3 列出了 SPI 主机接口的特征。

表 3-3. SPI 接口特征

特性	说明
最大频率	非标准 SPI 模式下为 25MHz，可选 50MHz <sup>(*)</sup>
数据总线	1 位 SPI
最大波特率	26Mbps (25MHz 时 1 位)。非标准 SPI 模式下为 52Mbps <sup>1</sup>
I/O 电压电平	1.8V

1. 对于非标准 SPI，主机处理器通过在时钟上升沿进行采样和移出，将支持的时钟频率加倍。

表 3-4 列出了 SPI 主机接口的信号。

表 3-4. SPI 接口信号

信号名称	吸电流	说明
CLK	控制器	串行位时钟
CS	控制器	芯片选择
PICO (外设输入控制器输出)	控制器	串行命令和数据输入
POCI (外设输出控制器输入)	外设	串行总线和数据输出，CS 为高电平和复位期间为高阻抗

#### 3.3.2 SPI Configuration

SPI 标准中定义了四种操作模式。为了成功进行通信，必须以相同的方式配置控制器和外设。这四种模式都是 SPI 时钟极性和相位的组合。仅支持模式 0，这意味着 SPI 时钟处于高电平有效状态，在这种状态下，数据在时钟的上升沿被采样，并在时钟的下降沿被移出。

为了获得更好的兼容性和灵活性，在主机接口初始化期间可以配置更多 SPI 参数。

- **小端字节序和大端字节序**：两者都受支持。在大端字节序中，最低有效字节在线路上优先传输。
- **16 位和 32 位字长**：均受支持。
- **位混合**：启用后，位混合会反转线路上每个字的位顺序。唯一例外是在 BUSY 信号期间，如“流控制”部分中所述。

表 3-5 说明了线路上数据的不同选项。

表 3-5. SPI 接口格式配置

格式	线路上的数据			
无混合的 16 位小端字节序	B1 ( b15 至 b8 )	B0 ( b7 至 b0 )	B3 ( b31 至 b24 )	B2 ( b23 至 b16 )
带混合的 16 位小端字节序	B0 ( b0 至 b7 )	B1 ( b8 至 b15 )	B2 ( b16 至 b23 )	B3 ( b24 至 b31 )
无混合的 16 位大端字节序	B0 ( b7 至 b0 )	B1 ( b15 至 b8 )	B2 ( b23 至 b16 )	B3 ( b31 至 b24 )
带混合的 16 位大端字节序	B1 ( b8 至 b15 )	B0 ( b0 至 b7 )	B3 ( b24 至 b31 )	B2 ( b16 至 b23 )
无混合的 32 位小端字节序	B3 ( b31 至 b24 )	B2 ( b23 至 b16 )	B1 ( b15 至 b8 )	B0 ( b7 至 b0 )
带混合的 32 位小端字节序	B0 ( b0 至 b7 )	B1 ( b8 至 b15 )	B2 ( b16 至 b23 )	B3 ( b24 至 b31 )
无混合的 32 位大端字节序	B0 ( b7 至 b0 )	B1 ( b15 至 b8 )	B2 ( b23 至 b16 )	B3 ( b31 至 b24 )
带混合的 32 位大端字节序	B3 ( b24 至 b3 )	B2 ( b16 至 b23 )	B1 ( b8 至 b15 )	B0 ( b0 至 b7 )

配置这些特性是在主机接口初始化期间完成的。由于主机接口的硬件线路默认为 SDIO 线路，并且由于 SPI 也使用相同的线路，因此使用 SDIO 命令 CMD0 应用 SPI 特性的初始配置（请参阅 SPI 章节）。

表 3-6 总结了所有可能的配置。

表 3-6. SPI 接口配置汇总

特性	支持的配置
时钟极性	在上升沿上对数据进行采样。对于标准 SPI，数据在下降沿上被移出，对于非标准 SPI，数据在上升沿上被移出
时钟相位	时钟在逻辑 0 处空闲
字大小	16 位或 32 位
主机字节序	除电流限制以外的
位顺序	可通过位混合进行配置
芯片选择极性	低电平有效
主机中断极性	除电流限制以外的
时钟频率	标准模式下最高为 26MHz，非标准模式下最高为 52MHz
字之间的芯片选择置位	两个字之间可能会出现高电平
自主模式	支持

### 3.3.3 SPI 流控制

与 SDIO 不同，SPI 接口不包含内置流控制机制。一种方案是添加硬件引脚来指示从器件是否准备好从主机控制器接收数据。另一种方案是实施专有的带内协议，该协议以专用命令或其他信令方法的形式使用 SPI 数据线。

CC33xx 配套 IC 实现外设到主机控制器的流控制，这意味着外设可以指示主机控制器停止传输。这是使用 POCI 数据线路通过指示 *busy* 信号来完成的。此方法还用于实现自主模式，以指示外设何时准备就绪以及何时未准备就绪。

值为 0 的位流在 *busy* 信号为高电平时，主机控制器预计会收到 16 位数据。图 3-1 说明了在写入方向中采用大端字节序格式的 16 位数据（最低有效字节在线路上优先传输）。

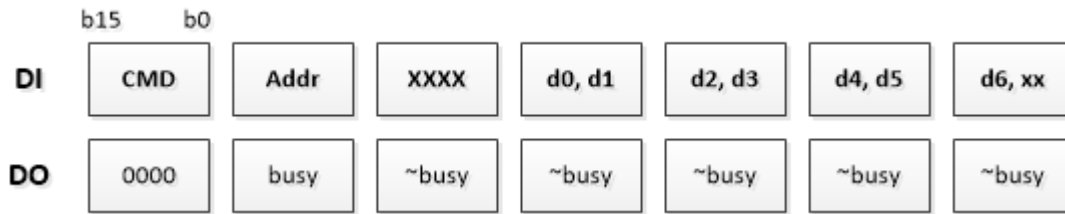


图 3-1. 16 位，大端字节序，写入方向

图 3-2 说明了在写入方向中采用大端字节序格式的 32 位数据（最高有效字节在线路上优先传输）。



图 3-2. 32 位，大端字节序，写入方向

CC33xx 配套 IC 还在读取方向上实现了外设到主机控制器的流控制，这意味着外设可以指示主机控制器停止接收（或何时开始接收）。它的工作原理与写入方向类似。

图 3-3 说明了在读取方向中采用大端字节序格式的 16 位数据（例如，最低有效字节在线路上优先传输）。



图 3-3. 16 位，大端字节序，读取方向

图 3-4 说明了在读取方向中采用小端字节序格式的 32 位数据（例如，最高有效字节在线路上优先传输）。

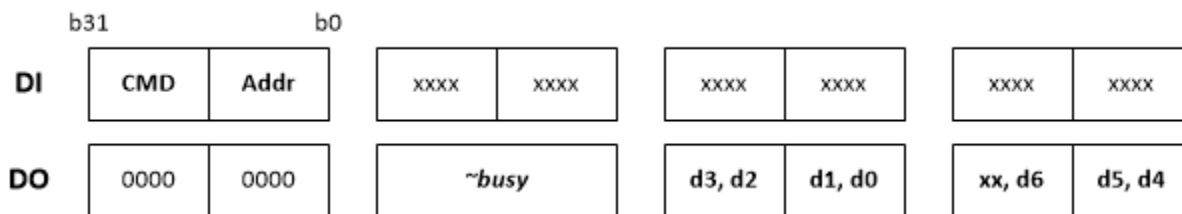


图 3-4. 32 位，大端字节序，读取方向



### 3.4 UART 接口

#### 3.4.1 UART 概述

表 3-7 列出了 UART 主机接口的特征。

**表 3-7. UART 接口特征**

特性	说明
UART 版本	4 线 (H4)
最大波特率	3M 波特
流控制	硬件 (RTS/CTS)
自动波特率检测	否
I/O 电压电平	1.8V

表 3-8 列出了 UART 主机接口的信号。

**表 3-8. UART 接口信号**

信号名称	吸电流	说明
TxD	外设	串行输出
RxD	控制器	串行输入
RTS ( 请求发送 )	外设	请求发送以发信号指示主机控制器停止/恢复传输
CTS ( 允许发送 )	控制器	允许发送以发信号指示外设停止/恢复传输

#### 3.4.2 UART 配置

表 3-9 总结了所有可能的配置。

**表 3-9. UART 接口配置**

特性	支持的配置
波特率	9600 波特、115200 波特或 3M 波特
自动波特率检测	否
流控制	硬件 (RTS/CTS)
数据位数	5-8 位
停止位	1、1.5 或 2 个停止位
奇偶校验	偶数、奇数或无奇偶校验
线路中断	检测和生成
自主模式	支持

#### 3.4.3 UART 流控制

通用异步接收器/发送器 (UART) 是一种传输协议，它以顺序方式将数据字节作为单个位流进行传输。在目标端，另一个 UART 将这些位重新组合成完整的字节。UART 由四条线路组成，分别用于数据传输 (Tx)、数据接收 (Rx)、保持主机传输的流控制 (RTS) 和保持器件传输的流控制 (CTS)。

在 UART 接口中，没有硬件定义的控制关系，每个实体都可以在全双工模式下独立地向另一方发送数据。硬件流控制利用两条硬件线路：RTS ( 请求发送 ) 和 CTS ( 允许发送 )，允许每一方在准备好处理数据时向另一方发出指示。这些电路在两个器件之间交叉耦合，因此一个器件上的 RTS 连接到另一器件上的 CTS，反之亦然。每个器件都使用其 RTS 来输出是否准备好接受新数据，并读取 CTS 以查看是否允许将数据发送到其他器件。

只要器件准备好接受更多数据，即可保持 RTS 线路有效。器件会在其接收缓冲区满之前的一段时间，将 RTS 置为无效。即使已将 RTS 置为无效，线路上和其他器件传输寄存器中仍可能有数据需要接收（在这种情况下，器件会在完成字节传输后停止传输）。另一个器件需要遵守流控制信号并暂停传输，直到 RTS 再次有效。

图 3-5 显示了主机和器件的硬件连接。

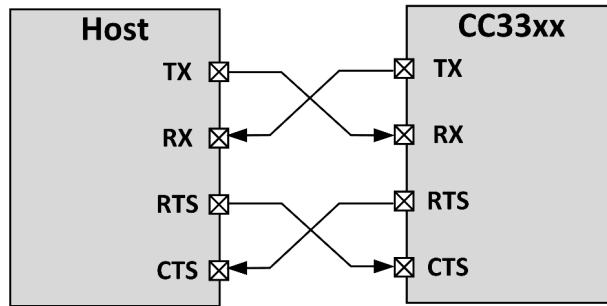


图 3-5. UART 接口流控制

### 3.5 引脚数选项

表 3-10 汇总了所有可能的引脚数选项。请注意，1 位 SDIO 模式尚未经过测试。

表 3-10. 引脚数选项

配置	Wi-Fi	BLE	中断	引脚数
双接口	4 位 SDIO	UART	OOB	6 (Wi-Fi) + 1 ( OOB 中断 ) + 4 (BLE) = 11 个引脚
双接口	1 位 SDIO	UART	OOB	3 (Wi-Fi) + 1 ( OOB 中断 ) + 4 (BLE) = 8 个引脚
共享接口	4 位 SDIO	不适用	IB	6 (Wi-Fi/BLE) = 6 个引脚
双接口	SPI	UART	OOB	4 (Wi-Fi) + 1 ( OOB 中断 ) + 4 (BLE) = 9 个引脚

## 4 主机通信

### 4.1 协议概述

无论使用什么硬件接口，流量都通过明确定义的协议传输。虽然每个硬件接口都使用不同的包装器来实现特定的硬件协议，但实际的有效负载是相同的。为了完整起见，本节列出了协议详细信息，但不需要开发人员/集成商进行任何移植工作。

该协议包含多个命令选项：

- 通用命令格式。在此命令中，不涉及 *host\_rx*

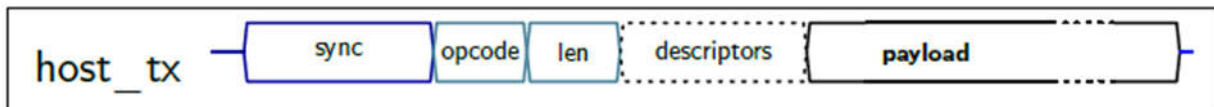


图 4-1. 通用命令

- *send()* 或 *sendto()* 命令格式。器件返回状态响应，指示命令成功



图 4-2. Send 或 Sendto 命令

- *recv()* 或 *recvfrom()* 命令格式，其中响应包含或不包含数据包

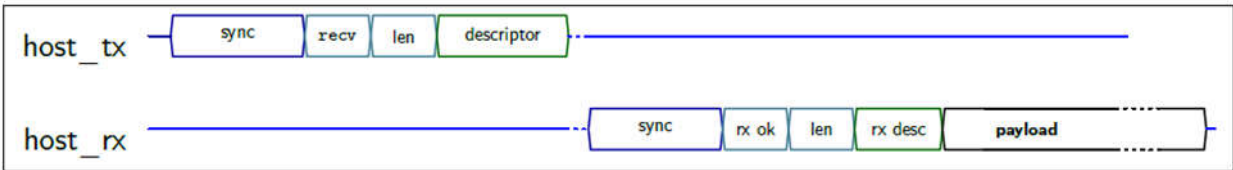


图 4-3. Recv 或 Recvfrom 命令

- *read\_status* 序列格式，其中主机请求器件提供即时状态

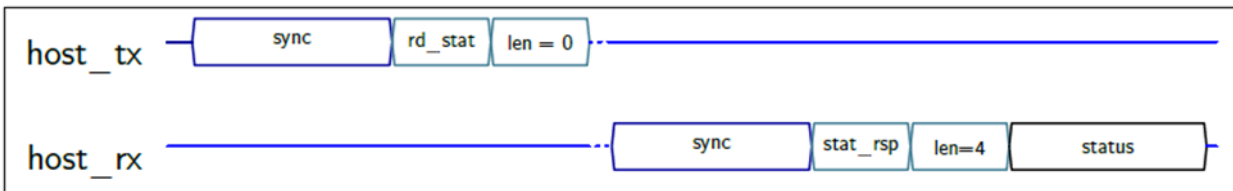


图 4-4. Read\_status 命令

- *cnys* 序列格式，其中主机请求器件提供单个最高优先级数据包

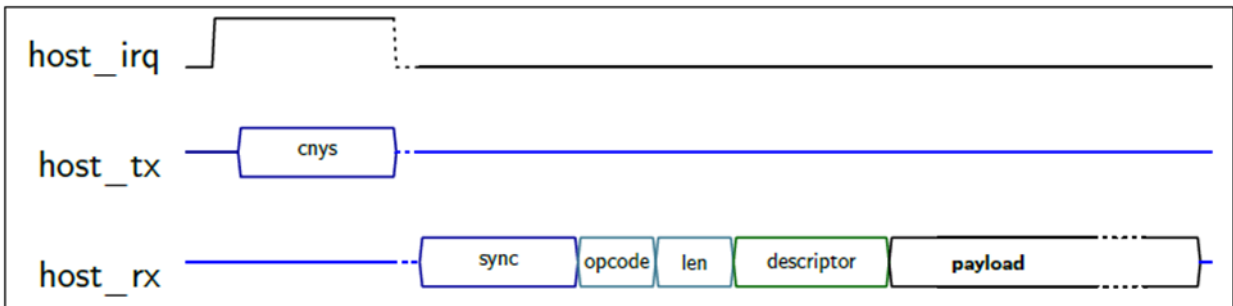


图 4-5. Cnys 命令

表 4-1 介绍了不同的命令字段。

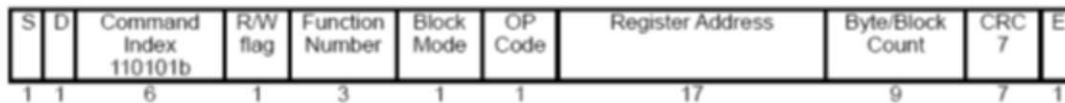
表 4-1. 协议结构

字段	说明
SYNC	用于同步命令或响应边界的常量模式
Opcode (操作码)	识别命令或响应 (取决于流量方向)
Length	此字段之后所有内容的长度
描述符	根据操作码/命令, 后面可能会有描述符来提供更多信息。例如, 对于 send() 操作, 描述符可以标识实际有效负载的套接字、标志和长度。还需要一个 Tx 状态响应字段
有效载荷	命令需要的任何其他内容
CNYS	触发器件以使用接收到的数据或事件进行响应的常量模式

## 4.2 SDIO 包装器

SDIO 包装器是一个标准 SDIO 命令。为了提供完整定义, 图 4-6 对 CMD53 结构 (在所有 Wi-Fi SDIO 事务中使用, 用于数据和控制) 进行了说明。

### CMD53



CMD53 access multiple bytes in any I/O function of SDIO card with single command

- ❖ D - Direction bit = 1
- ❖ Command Index = 0x35
- ❖ R/W flag – Determines the command type (Read = '0', Write= '1')
- ❖ Function Number – range between 0x0 and 0x7
- ❖ Block Mode – access performed in byte ('0') or block ('1') scale
- ❖ OP Code – access to fixed ('0') or incremental ('1') address
- ❖ Register Address – range between 0x0 and 0x1FFFF
- ❖ Byte/Block count – depend on Block Mode configuration
- ❖ Response R5

图 4-6. SDIO 包装器

### 4.3 SPI 包装器

SPI 包装器包括两个 16 位字的命令和地址。这些字段由主机驱动程序自动填充，如图 4-7 所示。

b47	b46	b45	b40	b39	b20	b19	b18	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b1	b0
start	tx	command	reserved	ops	fbrw	fbre	iod	ip	cs	ws	bs	de	wspi	CRC7	end				

- start** start bit = 0
- tx** transmission bit = 1
- command** command index, CMD0 = 0x00
- reserved** all 0's
- ops** Data output polarity select, 1 = Positive-edge driven flop, 0 = Negative-edge driven flop
- fbrw** Number of fixed-busy response words, 0 is an invalid entry
- fbre** Fixed-busy response enable
- iod** Host-Interrupt Open Drain, 1 = Open Drain, 0 = Push-Pull
- ip** Host-Interrupt Polarity, 1 = high, 0 = low
- cs** unused
- ws** data word size, two bytes = 0, four bytes = 1
- bs** data bit swizzle, b7/b15/b31 on the wire first = 0, b0 on the wire first = 1
- de** data endianness, little endian = 0, big endian, Byte 0 first = 1
- wspi** WLAN SPI mode = 1
- CRC7** CRC7, can be precalculated,  $G(x) = x^7 + x^3 + 1$
- end** end bit = 1

图 4-7. SPI 包装器

## 5 启动流程

主机处理器可以通过 **SPI** 或 **SDIO** 与配套 IC 通信。主机接口使用相同的硬件线路，器件根据预定义的顺序自动检测主机接口模式。对于主机接口，如果使用双接口模式，则该过程包括主主机接口（对于 Wi-Fi）和辅助主机接口（对于低功耗蓝牙）。

图 5-1 显示了从主机端开始的主接口初始化流程。

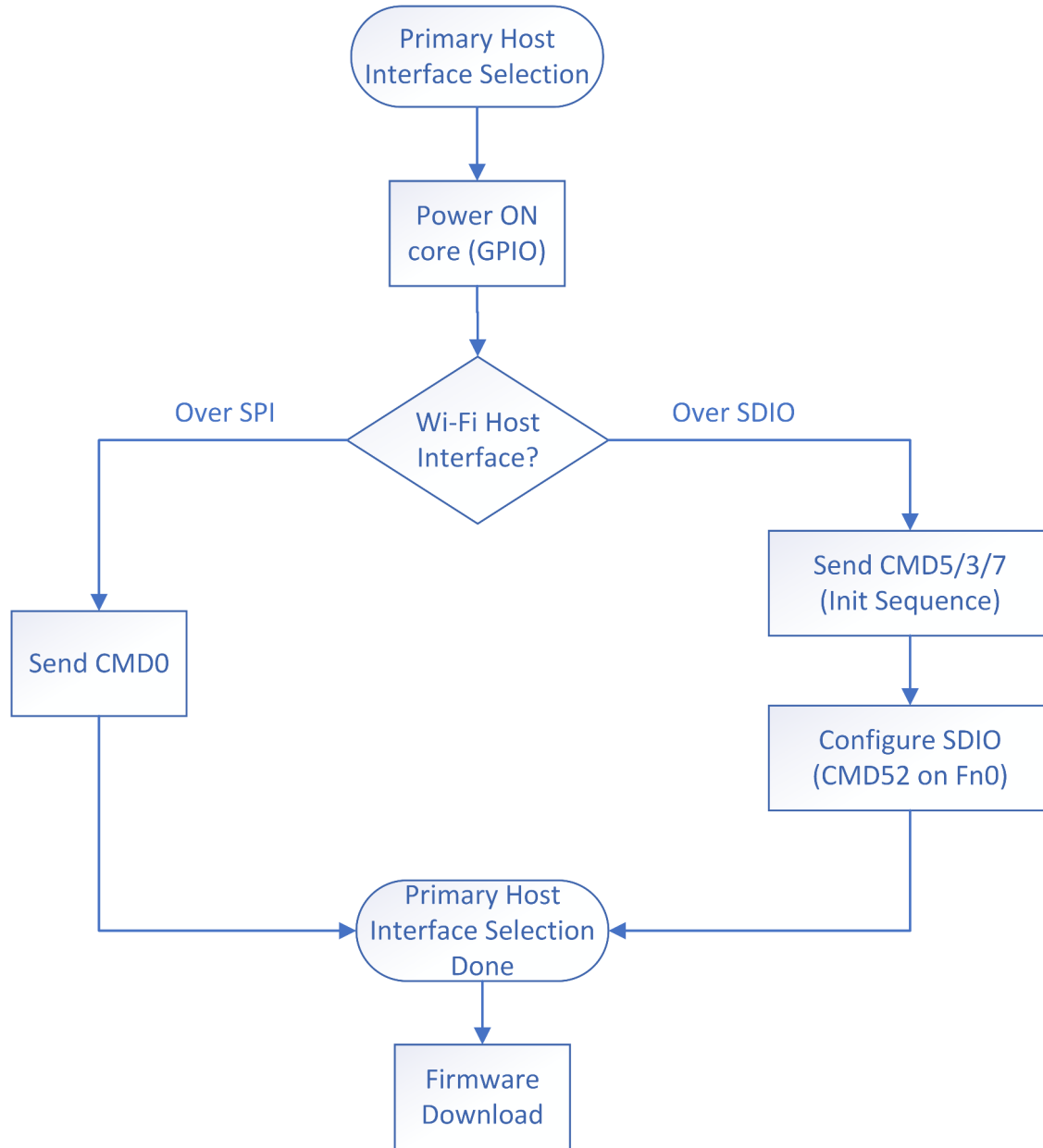


图 5-1. 主接口初始化

器件上电后，主机负责发送正确的命令序列。对于 **SPI** 模式，发送操作码为 0 (CMD0) 的单个 **SDIO** 命令，以便将器件设置为 **SPI** 模式。对于 **SDIO** 模式，该序列包括 **CMD5/3/7**，然后发送一系列 **CMD52** 命令，以设置器件的其他配置参数。

如果需要低功耗蓝牙，则可能需要执行辅助接口初始化流程。低功耗蓝牙可以在共享模式或双模式（通过 UART）下使用。从器件端，如果使用 UART，则在正确配置 UART 之前，器件应将 RTS 线路配置为手动控制的 GPIO 输出，并将 RTS 置为有效，以进一步保护内核，使其不接收任何主机命令，直到 UART 已配置并准备就绪。图 5-2 说明了从器件端开始的辅助接口初始化流程。

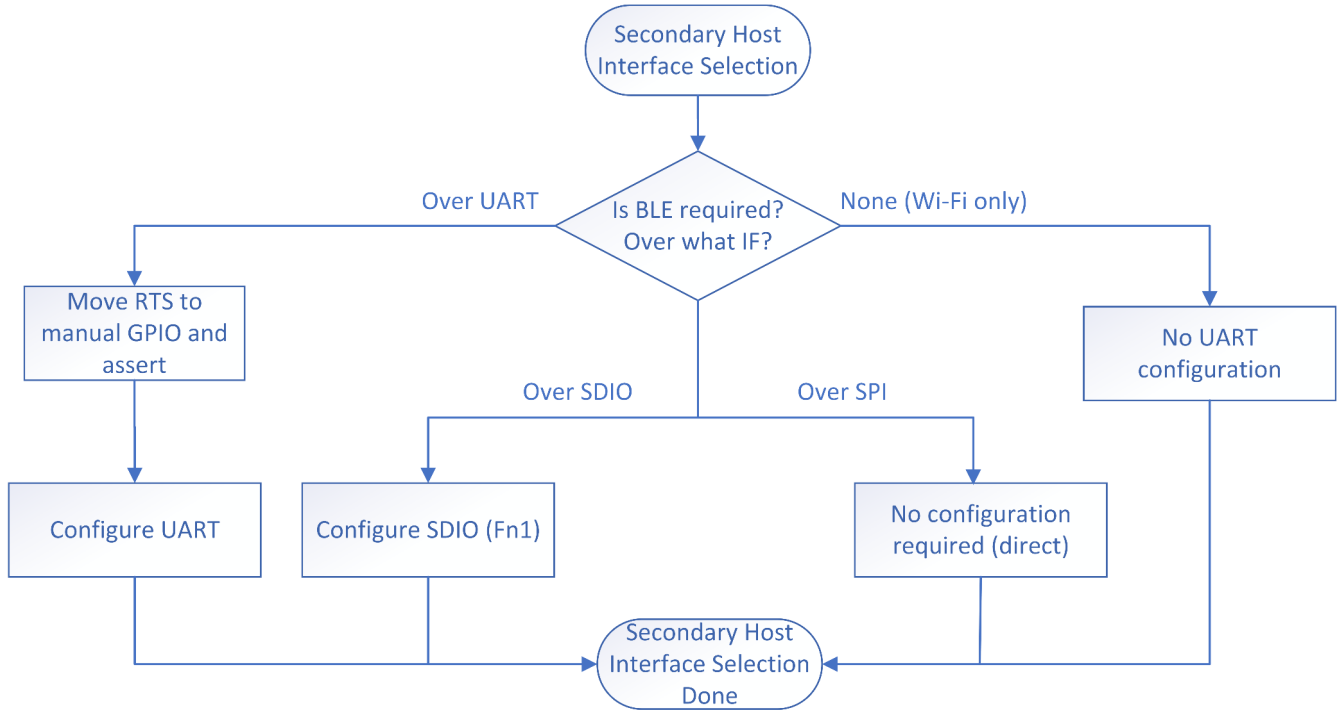


图 5-2. 辅助接口初始化

## 5.1 SDIO

在初始化到 SDIO 模式期间，不涉及数据线路，仅使用时钟和命令行。但是，如果 SDIO DAT0 线路被置为有效（设置为低电平），该过程可能会暂停并失败。这会被解读为 BUSY 信号，主机会推迟其传输，直到线路被置为无效。除非有外部下拉，否则器件端不应发生这种情况。

初始化过程中涉及的主要命令包括：

- CMD5：此命令用于查询器件所需的电压范围
- CMD3：此命令用于为器件分配逻辑地址
- CMD7：此命令用于根据逻辑地址选择所需器件，并将其启用以进行数据传输
- CMD52：此命令用于访问公共 I/O 区域、函数 0。使用一个命令/响应对读取/写入一个字节

## 5.2 SPI

图 5-3 显示了 SDIO CMD0 命令格式。

- 此 48 位命令必须在上电或复位后提供给器件
- 此命令在线路上传输的第一位始终是位 47
- 此命令由 3x16 位字组成，基于 SDIO CMD0

- 要将器件置于 WSPI 模式，必须设置命令的第 8 位

b47	b46	b45	b40	b39	b20	b19	b18	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b1	b0
start	tx	command	reserved	ops	fbrw	fbre	iod	ip	cs	ws	bs	de	wspi	CRC7	end				

<b>start</b>	start bit = 0
<b>tx</b>	transmission bit = 1
<b>command</b>	command index, CMD0 = 0x00
<b>reserved</b>	all 0's
<b>ops</b>	Data output polarity select, 1 = Positive-edge driven flop, 0 = Negative-edge driven flop
<b>fbrw</b>	Number of fixed-busy response words, 0 is an invalid entry
<b>fbre</b>	Fixed-busy response enable
<b>iod</b>	Host-Interrupt Open Drain, 1 = Open Drain, 0 = Push-Pull
<b>ip</b>	Host-Interrupt Polarity, 1 = high, 0 = low
<b>cs</b>	unused
<b>ws</b>	data word size, two bytes = 0, four bytes = 1
<b>bs</b>	data bit swizzle, b7/b15/b31 on the wire first = 0, b0 on the wire first = 1
<b>de</b>	data endianness, little endian = 0, big endian, Byte 0 first = 1
<b>wspi</b>	WLAN SPI mode = 1
<b>CRC7</b>	CRC7, can be precalculated, $G(x) = x^7 + x^3 + 1$
<b>end</b>	end bit = 1

图 5-3. SDIO CMD0 命令



## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司