目录

[SPI命令概述 2](#_Toc114611891)

[SPI帧结构 2](#_Toc114611892)

[主机框架结构 3](#_Toc114611893)

[响应帧结构 3](#_Toc114611894)

[位和字节顺序 4](#_Toc114611895)

[SPI命令集CMD 5](#_Toc114611896)

[命令参数PARAM 7](#_Toc114611897)

[驱动标识符DRID 8](#_Toc114611898)

[驱动选择 DRP 9](#_Toc114611899)

[司机分配顺序 9](#_Toc114611900)

[小电流驱动器选择 LCDRP 10](#_Toc114611901)

[LF 数据位顺序 11](#_Toc114611902)

[设备状态标志 STAT 12](#_Toc114611903)

[多状态标志处理 15](#_Toc114611904)

[所有命令使用的状态标志事件 16](#_Toc114611905)

[循环冗余校验CRC8 17](#_Toc114611906)

[命令帧时序 17](#_Toc114611907)

[直接帧响应 17](#_Toc114611908)

[非阻塞命令 18](#_Toc114611909)

[命令限制 19](#_Toc114611910)

[命令参数限制 21](#_Toc114611911)

[通用命令 21](#_Toc114611912)

[GET\_VERSION 21](#_Toc114611913)

[命令 21](#_Toc114611914)

[响应 21](#_Toc114611915)

[状态标志 22](#_Toc114611916)

[CONFIG\_DEVICE 22](#_Toc114611917)

[命令 22](#_Toc114611918)

[响应 23](#_Toc114611919)

[状态标志 23](#_Toc114611920)

[CONFIG\_ADVANCED 23](#_Toc114611921)

[命令 23](#_Toc114611922)

# SPI命令概述

NJJ29C0B 是一款联合低频驱动器和接收器 IC，带有嵌入式控制器。 NJJ29C0B 由主机控制器通过 SPI 命令控制，该命令是预定义 SPI 命令集的一部分，包含用于设备配置、设置和数据交换的命令。 主机控制器作为 SPI 主机，而 NJJ29C0B 作为 SPI 从机

## SPI帧结构

SPI 命令在 SPI 帧中传输。 一个 SPI 帧最多包含 256 个字节。 RX 和 TX 的默认值为 0x00，对应空闲模式下的 SPI 接口。

SPI 消息的第一个数据字节被解释为 SPI 消息长度 (LEN)。 LEN 在 0x01 和 0xFF 之间有效，并指定随后的数据字节数（通常在一个 SPI 消息中最多 255 个数据字节，一些命令支持更少的数据字节）。 数据字节可以包含任何内容（包括值 0x00）。

表 1. SPI 帧示例

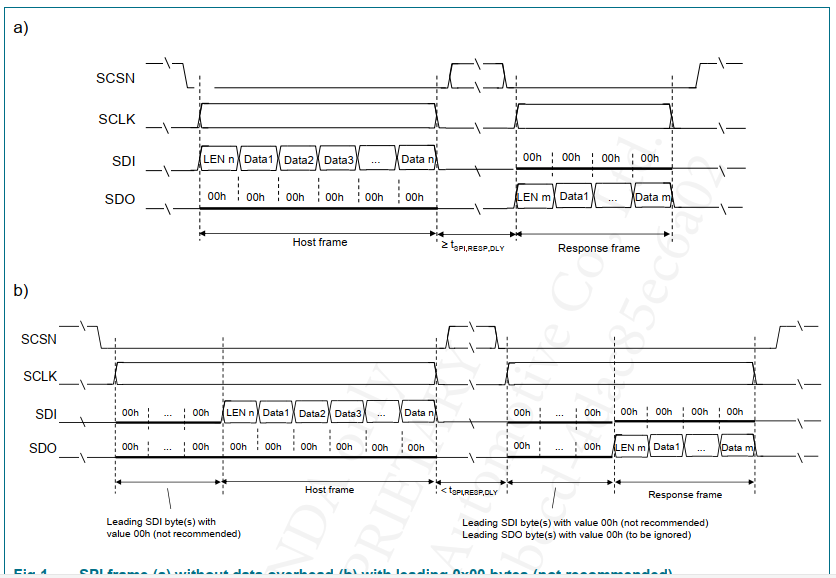
LEN 数据

0x04 0xF0 0x00 0x10 0x34

任何值为 0x00 的前导字节都将被丢弃。 任何时候都允许从前导 0x00 字节更改为任何其他值。

如果 NJJ29C0B 在预期时间内没有接收到指定数量的数据字节，则 RX 复位，第一个不同于 0x00 的字节被解释为下一个主机命令帧的长度。 如果接收到的数据字节多于预期，则忽略这些数据字节并清除包含这些字节的 RX 缓冲区。

从 NJJ29C0B 读取数据时，应使用 0x00 字节作为 TX 数据。 类似地，0x00 字节在接收时被发送。 图 1 显示了由主机帧和响应帧组成的 SPI 帧。 当既不打算 RX 也不打算 TX 操作时，建议不发送值为 0x00 的前导字节以减少设备负载。



### 主机框架结构

从主机到 NJJ29C0B 的命令帧结构如表 2 所示。

表 2. 从主机到 NJJ29C0B 的 SPI 命令结构

LEN CMD 参数 CRC8

1 字节 1 字节 0…253 字节 1 字节

CMD 保存应采取哪些操作的信息。 PARAM 是可选的，它的大小取决于 CMD。 PARAM 可以包含 0 到 253 个字节。

每个 SPI 主机帧在帧结束时由 CRC8 保护

### 响应帧结构

来自主机的每个命令帧总是由来自 NJJ29C0B 的直接响应帧确认。

在收到对前一个命令的直接响应（例如，通过发送 0x00 字节）或相应的最大响应延迟时间过去之前，不应发送下一个主机命令帧。

NJJ29C0B到主机的响应帧结构如表3所示

表 3. NJJ29C0B 到主机的 SPI 响应结构

LEN CMD 统计 参数 CRC8

1 字节 1 字节 1 字节 0…252 字节 1 字节

CMD 保存从主机帧到 NJJ29C0B 的回显值。 STAT 字节包含设备状态标志，这些标志表示应用程序故障或命令执行期间发生的特殊事件。 PARAM 编码取决于 CMD。

PARAM 可以包含 0 到 252 个字节。 每个 SPI 响应帧在帧结束时由 CRC8 保护

## 位和字节顺序

每个 SPI 字节以最低有效位在前（LSBit 在前）传输。 例如，字节 0xA2 (0b1010 0010) 作为位序列通过 SPI 接口以 0b0100 0101 的顺序传输

表 4. 示例：0xA2 的 LSBit 和 MSBit

MSBit LSBit

1 0 1 0 0 0 1 0

如果一个值包含 2 个或更多字节，则以最低有效字节在前（LSByte 在前）传输。 每个字节以最低有效位在前（LSBit 在前）传输。 例如，16 位值 0xC5A2 (0b1100 0101 1010 0010) 作为位序列通过 SPI 接口以 0b0100 0101 1010 0011 的顺序传输

表 5. 示例：0xC5A2 MSByte LSByte 的 LSBit、MSBit、LSByte 和 MSByte

MSBit LSBit MSBit LSBit

1 1 0 0 0 1 0 1 1 0 1 0 0 0 1 0

为了清楚起见，低位和高位字节的位置在每个相应的 SPI 命令中明确指定。 每个低字节标签以“\_L”结尾（例如“VALUE\_L”），而每个高字节标签以“\_H”结尾（例如“VALUE\_H”）。

表 6. 具有 2 字节值的示例 SPI 命令序列

LEN CMD 参数 CRC8

LEN 0xXX VALUE\_L VALUE\_H CRC8

对于 3 字节长度的值，中间字节在末尾标有“\_M”（例如“VALUE\_M”）

表 7. 具有 3 字节值的示例 SPI 命令序列

LEN CMD 参数 CRC8

LEN 0xXX VALUE\_L VALUE\_M VALUE\_H CRC8

由于并非所有值都需要 SPI 字节结构给出的所有可用位，因此这些位按升序从 0（LSByte 的 LSBit）到 MSByte 的 MSBit 编号

表 8. 值 0xC5A2 示例中的位编号

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

1 1 0 0 0 1 0 1 1 0 1 0 0 0 1 0

如果一个值由几个位组成，则记为 VALUE[m:n]。 下表显示了由 8、16 和 24 位组成的值的位表示法示例。

表 9. VALUE8BIT（1 个字节）

字节符号 位符号 描述

VALUE8BIT VALUE8BIT[7:0] 8 位值

表 10. VALUE16BIT（2 字节）

字节符号 位符号 描述

VALUE16BIT VALUE16BIT[15:0] 16 位值

VALUE16BIT\_H VALUE16BIT[15:8] 16 位值高字节

VALUE16BIT\_L VALUE16BIT[7:0] 16 位值低字节

表 11. VALUE24BIT（3 字节）

字节符号 位符号 描述

VALUE24BIT VALUE24BIT[23:0] 24 位值

VALUE24BIT\_H VALUE24BIT[23:16] 24 位值高字节

VALUE24BIT\_M VALUE24BIT[15:8] 24 位值中字节

VALUE24BIT\_L VALUE24BIT[7:0] 24 位值低字节

如果一个值包含的位少于完成一个字节边界所需的位，则可以使用其余位。 例如，由 23 位组成的值标记为 VALUE[22:0]。 在这种情况下，代表高字节的 MSBit 的 VALUE[23] 可以用于其他用途

## SPI命令集CMD

CMD 字节确定要采取的设备操作。 表 12 总结了 SPI 命令集

表 12. SPI 命令集[1][2]

命令 CMD 编码 说明

一般命令

GET\_VERSION 0xF0 读取设备版本标识符

CONFIG\_DEVICE 0x09 配置设备参数

CONFIG\_ADVANCED 0xD2 配置高级设备参数

设备重置

SET\_POR 0xF4 执行上电复位

GET\_POR\_STATUS 0x07 读取设备上电复位状态标志

CLEAR\_POR\_STATUS 0x08 清除设备复位后设置的 POR 状态标志

工作状态

START\_SLEEP 0x50 将设备设置为睡眠状态

START\_SLEEP\_FORCED 0x4F 将设备设置为 SLEEP 状态，与设备状态无关

升压转换器

CONFIG\_BOOST 0x45 配置升压转换器

低频天线驱动器

CONFIG\_LF\_DRIVER 0x40 配置低频驱动模式、波特率、电流、抖动、频率、转移

并联低电流驱动器

CONFIG\_LC\_DRIVER 0x44 配置低电流低频驱动

电报定序器

SET\_LF\_DATA 0x41 在 RAM 中存储要通过主驱动程序传输的数据

SET\_LC\_DATA 0x47 在 RAM 中存储要通过低电流驱动器传输的数据

START\_LF\_TRANSMIT 0x42 使用预加载的配置和数据开始 LF 传输

START\_LF\_TRANSMIT\_DATA 0x46 使用预加载配置和附加数据开始 LF 传输

STOP\_LF\_TRANSMIT 0x43 停止低频传输

防盗器

CONFIG\_IMMO\_DRIVER 0x60 配置防盗天线驱动

CONFIG\_IMMO\_BPLM 0x61 配置二进制脉冲长度调制器时序

CONFIG\_IMMO\_RECEIVER 0x62 配置防盗接收器

START\_IMMO 0x63 启动 immo 驱动并发送恒定载波信号

STOP\_IMMO 0x64 停止 immo 驱动程序传输的恒定载波信号

START\_IMMO\_TRANSCEIVE 0x65 写入应答器并读取应答器的响应

START\_IMMO\_TRANSMIT 0x66 向转发器写入数据

GET\_IMMO\_RESPONSE 0x67 从接收数据缓冲区读取数据

CLEAR\_IMMO\_STATUS 0x68 清除防盗器状态标志

SET\_IMMO\_MASK 0x69 屏蔽防盗器状态标志以触发 INT 引脚

天线参数

MEAS\_ANT\_IMP 0x48 测量天线阻抗值（变体 1）

MEAS\_ANT\_IMP\_ADVANCED 0xD5 测量天线阻抗值（变体 2）

SET\_ANT\_IMP 0x49 设置天线阻抗值

GET\_ANT\_IMP 0x4A 读取天线阻抗值

GET\_ANT\_IMP\_EFFECTIVE 0xD4 读取有效天线阻抗值

设备保护

GET\_PROT\_STATUS 0x58 读取设备保护状态标志（故障）

CLEAR\_PROT\_STATUS 0x59 清除设备保护状态标志

SET\_PROT\_MASK 0x5A 屏蔽设备保护标志以触发 INT 引脚

设备诊断

START\_DIAG 0x4C 启动升压转换器、LF 驱动器和天线的诊断

GET\_DIAG\_STATUS 0x4D 读取设备诊断状态标志

CLEAR\_DIAG\_STATUS 0x4E 清除设备诊断状态标志

SPI接口

CONFIG\_SPI 0xF1 配置SPI接口

ECHO\_SPI 0x01 检查 SPI 链路的正确物理操作

唤醒端口

CONFIG\_WUP 0x10 配置唤醒端口

GET\_WUP\_STATUS 0x13 读取唤醒状态标志

CLEAR\_WUP\_STATUS 0x14 清除唤醒状态标志

SET\_WUP\_MASK 0x15 屏蔽唤醒状态标志以触发 INT 引脚

WUP 事件触发轮询

CONFIG\_WUP\_POLLING 0x51 配置通过唤醒端口唤醒的驱动程序序列

定时器触发轮询

CONFIG\_TIMER\_POLLING 0x52 配置轮询定时器和轮询方案

START\_TIMER\_POLLING 0x53 将设备设置为 SLEEP 状态并启动定时器触发轮询

温度指示

CONFIG\_TEMP 0x18 配置温度警告阈值

GET\_TEMP\_STATUS 0x19 读取温度状态标志

CLEAR\_TEMP\_STATUS 0x1A 清除温度状态标志

SET\_TEMP\_MASK 0x1B 用于触发 INT 引脚的掩码温度状态标志

操作状态标志

GET\_OP\_STATUS 0x55 读取操作状态标志

CLEAR\_OP\_STATUS 0x56 清除操作状态标志

SET\_OP\_MASK 0x57 屏蔽操作状态标志以触发 INT 引脚

程序下载

DOWNLOAD\_PROG 0x30 将程序存储在 RAM 中以供执行

START\_PROG 0x31 执行下载到 RAM 中的程序

GET\_PROG\_SIG 0x33 读取程序签名

[1] 命令集可能会发生变化

[2] 除非另有说明，否则只允许使用指定的命令

在详细的 SPI 命令描述中，标记为 RFU 的位和字节保留供将来使用。 为了将来的兼容性，写操作应分别为 RFU 位或字节分配“0”或“0x00”。 RFU 位或字节的任何读操作分别产生“0”或“0x00”

## 命令参数PARAM

大多数 SPI 命令包含参数，代表设置和/或数据。 这些参数编码在 SPI 帧中包含的 PARAM 字节中

参数：

PARAM 包含一个或多个命令参数 A、B、C、...。

表 13. 不同类型参数的 SPI 命令

LEN CMD 参数 CRC8

LEN 0xXX A B C … CRC8

参数可以是相同类型 A（例如数据字节），重复 A1、A2、...

表 14. 具有相同类型参数的 SPI 命令

LEN CMD 参数 CRC8

LEN 0xXX A1 A2 … CRC8

为了更清楚起见，在本文档中，重复 A1、A2、……的参数由其名称表示，后跟大括号 {Ai} 中的“i”。 SPI 消息长度 LEN 用于确定相应命令中的参数数量。 参数的最小数量应为 1

表 15. 具有相同类型参数的简化表示法的 SPI 命令

LEN CMD 参数 CRC8

LEN 0xXX {Ai} CRC8

一些命令允许重复参数序列 A1、B1、C1、A2、B2、C2、... 包含不同类型的参数。 参数按给定顺序依次重复。 每个参数序列代表一个参数集。 重复参数的顺序不受约束，因此顺序 A1、B1、C1、A2、B2、C2 等价于顺序 A2、B2、C2、A1、B1、C1

表 16. 具有顺序重复参数的 SPI 命令

LEN CMD 参数 CRC8

LEN 0xXX A1 B1 C1 A2 B2 C2 A3 B3 C3 … CRC8

在简化的符号中，这通过将顺序重复的参数嵌入大括号 {Ai, Bi, Ci, ...} 中来表示。 SPI 消息长度 LEN 用于确定相应命令中的参数序列（参数集）的数量。 重复参数的序列数可以是1

表 17. 以简化符号顺序重复参数的 SPI 命令

LEN CMD 参数 CRC8

LEN 0xXX {爱笔字} CRC8

如果在同一命令中以不同的值多次传输相同的参数，则最后传输的设置将覆盖任何先前的设置。

### 驱动标识符DRID

一些 LF 驱动相关的 SPI 命令有一个驱动标识符 DRID 来选择请求的驱动，后跟这个通道的参数

表 18. LF 驱动程序相关命令的 SPI 命令结构

参数

DRID

DRID：

驱动程序（通道）ID 标识为其分配参数集的 LF 通道

表 19. DRID

位 符号 访问 值 说明

7 到 3 RFU W0 保留供将来使用

2 到 0 DRID W 驱动程序标识符

1. 低频驱动器 1
2. 001 低频驱动器 2
3. 010低频驱动器3
4. 011 低频驱动器 4
5. 100 低频驱动器 5
6. 101 低频驱动器 6
7. 110 保留供将来使用
8. 111 保留供将来使用

### 驱动选择 DRP

有些命令需要同时选择多个驱动程序。 这些命令通过设置参数 DRP 中的相应位来选择相应的驱动程序

表 20. LF 驱动器选择参数

参数

DRP RFU

DRP：

选择 LF 驱动器 Tx

表 21. DRP

位 符号 访问 值 说明

7 到 6 RFU W0 保留供将来使用

5 DR6P W 选择低频驱动器 6

0 无选择

1 选择

4 DR5P W 选择低频驱动器 5

0 无选择

1 选择

3 DR4P W 选择低频驱动器 4

0 无选择

1 选择

2 DR3P W 选择低频驱动器 3

0 无选择

1 选择

1 DR2P W 选择低频驱动器 2

0 无选择

1 选择

0 DR1P W 选择低频驱动器 1

0 无选择

1. 选择

### 司机分配顺序

在某些情况下，设备必须自己将相应的专用驱动程序分配给通过主机命令传输的参数序列。 在这种情况下，驱动程序按驱动程序编号升序分配给参数序列，从最低编号开始（表 22）

表 22. 驱动程序分配顺序

编号 驱动程序分配

1 个低频驱动器 1

2 低频驱动器 2

3 低频驱动器 3

4 低频驱动器 4

5 低频驱动器 5

6 低频驱动器 6

例子：

如果需要在一个命令中分配驱动程序 1、3 和 4，则应用以下顺序：

• 1st：LF 驱动器 1

• 2nd：LF 驱动器 3

• 3rd：LF 驱动器 4

### 小电流驱动器选择 LCDRP

启动 LF 传输的命令允许激活与主驱动器并行的低电流驱动器。 通过设置参数 LCDRP 中的相应位，只允许选择未被选择为主驱动器的通道的低电流驱动器。

激活主驱动器时，选定的低电流驱动器自动开始发射

表 23. 低电流驱动器选择参数

参数

LCDRP RFU

LCDRP：

LCDRP 选择低电流 LF 驱动器 Tx

表 24. LCDRP（复位值 0x00）

位 符号 访问 值 说明

7 到 6 RFU W0 保留供将来使用

5 LCDR6P W 选择低频驱动器 6

0 无选择

1 选择

4 LCDR5P W 选择低频驱动器 5

0 无选择

1 选择

3 LCDR4P W 选择低频驱动器 4

0 无选择

1 选择

2 LCDR3P W 选择低频驱动器 3

0 无选择

1 选择

1 LCDR2P W 选择低频驱动器 2

0 无选择

1 选择

0 LCDR1P W 选择低频驱动器 1

0 无选择

1. 选择

### LF 数据位顺序

发送或接收 LF 数据的序列通过将数据拆分为完整字节和最后一个字节来处理，如果依赖于 LF 协议，则位数偏离 8 位的倍数

#### LF 发射

以下 SPI 命令指定要通过设备 LF 接口传输的数据

• SET\_LF\_DATA

• SET\_LC\_DATA

• START\_LF\_TRANSMIT\_DATA

• START\_IMMO\_TRANSMIT

• START\_IMMO\_TRANSCEIVE（传输部分）

数据通过 SPI 接口以字节为导向（每个数据值由一个字节表示）传输，从第一个数据字节开始到最后一个数据字节。

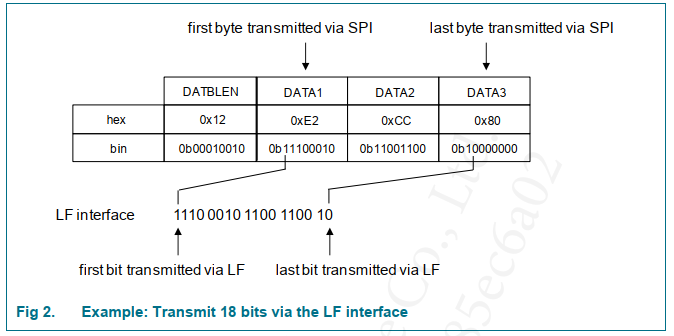
最后一个可能不完整的字节保持有效数据 MSB 对齐（例如，对于 5 位 0b10101xxx，最后 3 位 xxx 应设置为“0”并被嵌入式系统软件忽略）。 数据以 MSB 先发送

例子：

应通过 LF 接口传输 18 位序列 (DATBLEN = 0x12)

0b1110 0010 1100 1100 10

在这个例子中，首先 DATA1 被完全发送，然后 DATA2 被发送。 由于指定的总数据长度为 18 位，最后 DATA3 的前 2 位是 MSB 对齐的 Pr，其余 6 位 DATA3 的低位用零填充（图 2）



#### LF接收

一个命令处理通过设备 LF 接口接收的数据

• GET\_IMMO\_RESPONSE

数据通过 SPI 接口按字节接收（每个数据值由一个字节表示），从第一个数据字节开始一直到最后一个数据字节。

最后一个可能不完整的字节保持有效数据 LSB 对齐（例如，对于 5 位 0bxxx10101，高前 3 位设置为“0”，主机控制器应忽略）。

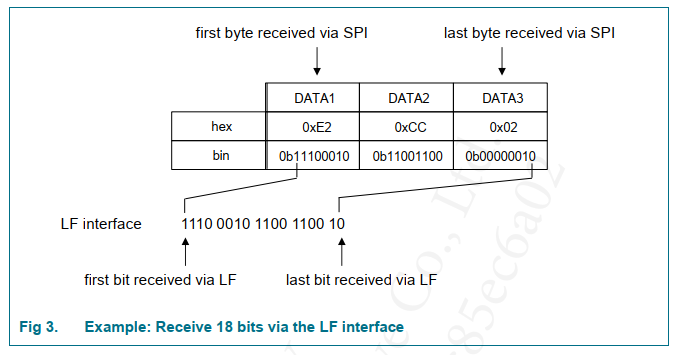
接收时序始终是 8 位的倍数。 嵌入式系统软件等待通过 LF 接口接收到的最后一个完整字节，并用零替换未使用的位。

例子：

应通过 LF 接口接收 18 位序列

0b1110 0010 1100 1100 10

在该示例中，首先完整接收 DATA1，然后接收 DATA2。 由于指定的总数据长度为 18 位，最后 DATA3 的前 2 位是 LSB 对齐的，而 DATA3 的其余 6 个高位用零填充（图 3）。



## 设备状态标志 STAT

每个命令响应都包含设备状态标志。 器件状态标志编码在 STAT 字节中，包括指示器件复位的标志状态。

统计：

设备状态标志可以分为两组

• 动态状态标志 STAT[7:4] 和

• 静态状态标志 STAT[3:0]

动态状态标志

• SF\_CMD

• SF\_LAST\_OP

• SF\_CRC 和

• SF\_PAR

表示最后一个 SPI 命令的结果并动态更改它们的值。

静态标志

• SF\_OP

• SF\_WUP

• SF\_PROT 和

• SF\_POR

表示相应组中汇总事件的组标志（操作状态标志组、唤醒状态标志组、保护状态标志组和上电复位状态标志组）。 一旦设置，这些标志将保持它们的值，直到主机控制器清除相应组中的相应详细状态标志

表 25. STAT（复位值 0x01）

位 符号 访问 值 说明

7 SF\_CMD R 命令码解释状态

0 有效命令

1 命令未知或（暂时）不允许

6 SF\_LAST\_OP R LF 场激活状态

0 激活成功

1 激活失败

5 SF\_CRC R CRC 状态

0 无错误

1 错误

4 SF\_PAR R 参数解释状态

0 可行

1 失败

3 SF\_OP R 运行状态

0 无请求

1 个请求

2 SF\_WUP R 唤醒状态

0 无请求

1 个请求

1 SF\_PROT R 保护状态

0 无请求

1 个请求

0 SF\_POR R 设备复位状态

0 无复位

1. 重置

SF\_CMD：

如果命令不允许（此时），则设置 SF\_CMD，因为

• 使用了未知的 SPI 命令编码（枚举器故障）

• 在实际设备操作状态下不允许发送 SPI 命令

• NJJ29C0B 处理非阻塞命令，主机控制器在 NJJ29C0B 设置操作状态标志之前询问不允许的 SPI 命令

如果提供的 SPI 响应设置了状态标志 SF\_CMD，则未执行先前发送的命令主机帧的预期功能。

SF\_LAST\_OP：

SF\_LAST\_OP 如果在激活 LF 字段时在以下操作期间发生错误，则设置

• 升压转换器启动

• LF 驱动器电源启动

• LF 驱动器启动

• 电流控制回路启动

• PLL 启动

• Immo LDO 诊断

• Immo ADC 校准（仅适用于 immo 操作）

因此，SF\_LAST\_OP 表示无法执行预期的驱动程序操作。

对于大多数可以上报SF\_LAST\_OP的SPI命令，SF\_LAST\_OP在直接响应中不设置，而是在后续命令的响应中上报。 一个例外是命令 START\_IMMO，它在直接响应中报告 SF\_LAST\_OP。

由于除了 SF\_LAST\_OP 之外，SF\_PROT 也与 LF 驱动程序操作有关，以下条件显示根据故障条件设置哪个标志

SF\_LAST\_OP 已设置

• 对于命令 START\_LF\_TRANSMIT、START\_LF\_TRANSMIT\_DATA、START\_TIMER\_POLLING：

如果在 tLF 内 LF 激活期间，START 接收到 STOP\_LF\_TRANSMIT 命令，则在后续命令的响应中报告 SF\_LAST\_OP 标志。

• 对于命令 MEAS\_ANT\_IMP 和 MEAS\_ANT\_IMP\_ADVANCED：

如果 VBAT 大于 18V，则不执行天线阻抗测量，并且在后续命令的响应中报告 SF\_LAST\_OP 标志。

• 对于命令 MEAS\_ANT\_IMP\_ADVANCED：

如果一个或多个天线的阻抗测量不成功，例如，由于天线连接断开，则在后续命令的响应中报告 SF\_LAST\_OP 标志。

• 对于激活 LF 字段的所有命令：

如果在激活 LF 字段的操作过程中发生错误（并且没有保护），则在后续命令的响应中报告 SF\_LAST\_OP（例外：START\_IMMO 在直接响应中报告 SF\_LAST\_OP）

SF\_PROT 已设置

• 对于激活 LF 字段的所有命令：

如果在 LF 激活开始之前发生保护，则在后续命令的响应中报告 SF\_PROT 并且没有 SF\_LAST\_OP 标志。

SF\_LAST\_OP 和 SF\_PROT 均已设置

• 对于激活 LF 字段的所有命令：

当启用任何硬件块后引发设备保护标志时，SF\_LAST\_OP 和 SF\_PROT 都会在后续命令的响应中报告（例外：START\_IMMO 在直接响应中报告 SF\_LAST\_OP 和 SF\_PROT）。

SF\_CRC：

如果附加到 SPI 命令的 CRC8 错误，则设置 SF\_CRC

需要注意的是，如果在命令响应中设置了 SF\_CRC，则回显的命令代码不能被视为正确。

如果提供的 SPI 响应设置了状态标志 SF\_CRC，则未执行先前发送的命令主机帧的预期功能

SF\_PAR：

如果无法正确解释通过 SPI 命令传输的参数，则设置 SF\_PAR。 这包括超出其有效范围的参数值和不一致的长度，尤其是在级联帧上。 如果 RFU 位设置为 1，SF\_PAR 也会设置。

此外，如果参数集不可行，则设置 SF\_PAR，例如，如果发送 CLEAR\_XX\_STATUS 命令而没有标记为清除的状态标志。

如果 SPI 响应设置了状态标志 SF\_PAR，则相应命令主机帧的预期功能未执行

SF\_OP：

如果非阻塞命令已完成其操作并因此设置其专用操作状态标志，则设置 SF\_OP。

如果提供的 SPI 响应设置了状态标志 SF\_OP，则应询问命令 GET\_OP\_STATUS 以检查哪个操作已完成。

SF\_WUP：

如果在启用的唤醒端口检测到有效的唤醒事件，则设置 SF\_WUP。

如果在状态标志 SF\_WUP 设置的情况下提供 SPI 响应，则应询问命令 GET\_WUP\_STATUS 以检查哪个唤醒端口已触发。

SF\_PROT：

如果触发了设备保护事件，则设置 SF\_PROT。设备通过硬件停用升压转换器和 D\* 类驱动程序（如果已激活）。之后，它进入 ERROR 状态。

如果提供的 SPI 响应设置了状态标志 SF\_PROT，则应询问命令 GET\_PROT\_STATUS 以检查触发了哪个保护事件。

SF\_POR：

如果发生设备上电复位 (POR)，则设置 SF\_POR。释放上电复位后，SPI 响应提供状态标志 SF\_POR 设置。

如果提供的 SPI 响应设置了状态标志 SF\_POR，则应询问命令 GET\_POR\_STATUS 以检查哪个事件触发了上电复位

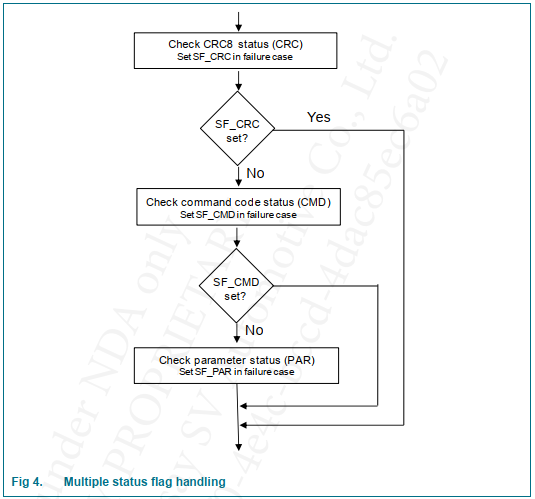
### 多状态标志处理

取决于 SPI 命令和参数，在某些情况下，可能适合设置多个设备状态标志。 在这些情况下，将按以下顺序检查和设置状态标志（图 4）。

• SF\_CRC 总是首先检查，因为在故障情况下，SPI 命令的剩余内容不可靠

• SF\_CMD 在 SF\_CRC 之后处理并否决 SF\_PAR，因为参数评估以 SPI 命令的精确知识为前提

• 如果之前未检测到 SF\_CRC 和 SF\_CMD 问题，则会评估 SF\_PAR



如果检测到故障，则中止参数检查。 例如，当主机控制器在禁止设备操作状态下询问 SPI 命令时，设置 SF\_CMD，而不检查 SPI 命令参数 PARAM。

### 所有命令使用的状态标志事件

如果发生以下故障或事件之一，所有 SPI 命令通常使用以下设备状态标志

表 26. 所有命令使用的状态标志事件

状态字节 状态位 报告的故障或事件

STAT SF\_CMD 命令不允许（在当前设备状态下）[1]

SF\_LAST\_OP LF 电源路径激活期间失败（由之前直接发送的命令引起）

LF 传输中止（由之前直接发送的命令 STOP\_LF\_TRANSMIT 导致，在启动 LF 之后和 tLF 之前询问时，START 已经过去）

SF\_CRC SPI CRC 错误

SF\_PAR 无效的 SPI 帧长度

保留供将来使用 (RFU) 位或字段值设置为 > 0[2]

SF\_OP 接收到的所有 LF 数据（由之前发送的非阻塞命令引起）

发送的所有LF数据（由之前发送的非阻塞命令引起）

天线阻抗测量完成（由之前发送的非阻塞命令引起）

天线诊断完成（由之前发送的非阻塞命令引起）

SF\_WUP WUP 触发

SF\_PROT 保护触发

SF\_POR 设备复位触发

将剩余的设备状态标志事件分配给相应的 SPI 命令在详细的 SPI 命令描述中给出

## 循环冗余校验CRC8

每个 SPI 命令附加一个 8 位循环冗余校验值 (CRC8)。 CRC8 计算包括所有数据字节，包括 LEN 直到最后一个 PARAM 字节。

CRC8：

CRC8 根据以下算法计算。

CRC8 计算

字 wCRC；

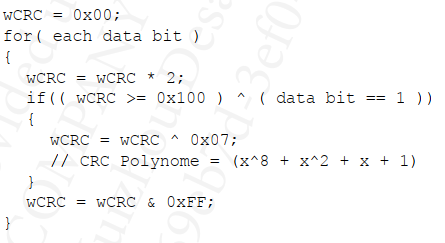


表 27. CRC8 计算示例

输入（十六进制） CRC8（十六进制）

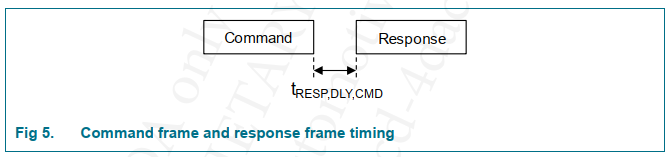
00 08 3F 00 4A

## 命令帧时序

### 直接帧响应

直接响应帧在接收到主机帧的 CRC8 后延迟时间发送。 该响应延迟时间 tRESP,DLY,CMD 取决于 SPI 命令和命令参数，因为需要检查正确的命令长度、CRC8 和命令参数（图 5）。

重要的是要注意集成 µController 的任何其他处理活动，如保护处理、活动 LF 接口（例如 LF 通道更改）、WUP 事件可能会扩大响应延迟时间。



### 非阻塞命令

一些 SPI 命令在 NJJ29C0B 上启动需要延长处理时间并且具有操作状态标志的操作。表 28 总结了这些非阻塞命令

表 28. 带有操作状态标志的非阻塞命令

非阻塞命令 操作状态标志

START\_LF\_TRANSMIT SF\_TXREADY

START\_LF\_TRANSMIT\_DATA SF\_TXREADY

START\_IMMO\_TRANSMIT SF\_TXREADY

START\_IMMO\_TRANSCEIVE SF\_RXREADY

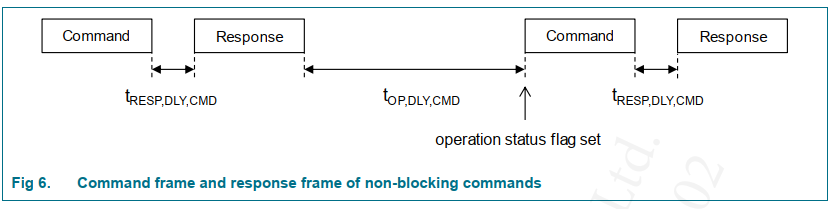
MEAS\_ANT\_IMP SF\_IMPMEAS

MEAS\_ANT\_IMP\_ADVANCED SF\_IMPMEAS

START\_DIAG SF\_DIAG

接收到这些命令之一后，准备发送 SPI 响应并启动已启动的 NJJ29C0B 操作（例如诊断）。 在命令响应延迟时间 tRESP,DLY,CMD 之后，应用程序可以询问指示命令已执行的直接响应。 发送直接 SPI 响应后，启动的操作正在进行（图 6）。

操作延迟时间 tOP,DLY,CMD 取决于命令和命令参数。 操作完成由专用操作状态标志指示，主机控制器可以检查该标志，或者可以将哪个设置分配给设备 INT 引脚。



发送非阻塞命令后，操作完成前，允许查询以下 SPI 命令

• GET\_POR\_STATUS

• GET\_PROT\_STATUS

• GET\_DIAG\_STATUS

• GET\_WUP\_STATUS

• GET\_TEMP\_STATUS

• GET\_OP\_STATUS

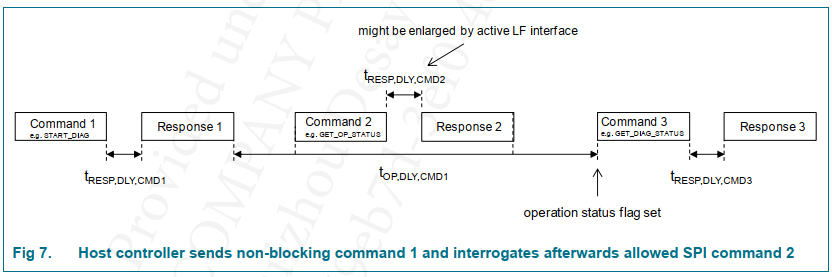
• START\_SLEEP\_FORCED

• SET\_POR

• STOP\_LF\_TRANSMIT（仅在发送命令 START\_LF\_TRANSMIT 或 START\_LF\_TRANSMIT\_DATA 之后）

• STOP\_IMMO（仅在发送命令 START\_IMMO\_TRANSMIT 或 START\_IMMO\_TRANSCEIVE 后）

特别是 GET\_OP\_STATUS 用于检查启动的操作何时完成（如果 INT 引脚未用于此目的）（图 7）



如果主机控制器在 NJJ29C0B 设置操作状态标志之前询问不同的 SPI 命令，则不执行该命令，并在设置状态标志 SF\_CMD 的情况下提供 SPI 响应

由于 SPI 命令的处理需要一定的处理时间，其中启动的 NJJ29C0B 操作会延迟，在运行过程中询问多个 SPI 命令可能会影响操作执行时间。

启动操作完成后，允许再次发送所有 SPI 命令

## 命令限制

某些 SPI 命令仅在特殊设备条件下才允许使用。 表 29 总结了依赖于器件工作状态的 SPI 命令限制。

如果询问了在相应设备操作状态中不允许的 SPI 命令，则不处理此命令，并且直接 SPI 响应包含设置的设备状态标志 SF\_CMD

表 29. SPI 命令限制

Command IDLE state PKE state IMMO state DIAG state ERROR state 运行中的状态 完成后的状态

一般命令

GET\_VERSION 允许 不允许 不允许 不允许 允许 无变化 无变化

CONFIG\_DEVICE 允许 不允许 不允许 不允许 允许 无变化 无变化

CONFIG\_ADVANCED允许 不允许 不允许 不允许 允许 无变化 无变化

设备重置

SET\_POR 允许 允许 允许 允许 允许 冷启动 IDLE

GET\_POR\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_POR\_STATUS允许 不允许 不允许 不允许 允许 无变化 无变化

工作状态

START\_SLEEP 允许 不允许 不允许 不允许 允许 无变化 SLEEP

START\_SLEEP\_FORCED允许 允许 允许 允许 允许 无变化 SLEEP

升压转换器

CONFIG\_BOOST 允许 不允许 不允许 不允许 允许 无变化 无变化

低频天线驱动器

CONFIG\_LF\_DRIVER允许 不允许 不允许 不允许 允许 无变化 无变化

并联低电流驱动器

CONFIG\_LC\_DRIVER允许 不允许 不允许 不允许 允许 无变化 无变化

电报定序器

SET\_LF\_DATA 允许 不允许 不允许 不允许 允许 无变化 无变化

SET\_LC\_DATA 允许 不允许 不允许 不允许 允许 无变化 无变化

START\_LF\_TRANSMIT允许 不允许 不允许 不允许 不允许 PKE IDLE

START\_LF\_TRANSMIT\_DATA允许 不允许 不允许 不允许 不允许 PKE IDLE

STOP\_LF\_TRANSMIT不允许 允许 不允许 不允许 不允许 无变化 IDLE

防盗器

CONFIG\_IMMO\_DRIVER允许 不允许 不允许 不允许 允许 无变化 无变化

CONFIG\_IMMO\_BPLM允许 不允许 不允许 不允许 允许 无变化 无变化

CONFIG\_IMMO\_RECEIVER 允许 不允许 不允许 不允许 允许 无变化 无变化

START\_IMMO 允许 不允许 不允许 不允许 不允许 无变化 IMMO

STOP\_IMMO 不允许 不允许 允许 不允许 不允许 无变化 IDLE

START\_IMMO\_TRANSCEIVE 不允许 不允许 允许 不允许 不允许 不允许 无变化 无变化

START\_IMMO\_TRANSMIT 不允许 不允许 允许 不允许 不允许 不允许 无变化 无变化

GET\_IMMO\_RESPONSE 允许 不允许 允许 不允许 允许 无变化 无变化

CLEAR\_IMMO\_STATUS 允许 不允许 允许 不允许 允许 无变化 无变化

SET\_IMMO\_MASK 允许 不允许 不允许 不允许 允许 无变化 无变化

天线阻抗

MEAS\_ANT\_IMP 允许 不允许 不允许 不允许 不允许 不允许 PKE IDLE

MEAS\_ANT\_IMP\_ADVANCED 允许 不允许 不允许 不允许 不允许 不允许 PKE IDLE

SET\_ANT\_IMP 允许 不允许 不允许 不允许 允许 无变化 无变化

GET\_ANT\_IMP 允许 不允许 不允许 不允许 允许 无变化 无变化

GET\_ANT\_IMP\_EFFECTIVE 允许 不允许 不允许 不允许 允许 无变化 无变化

设备保护

GET\_PROT\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_PROT\_STATUS允许 不允许 不允许 不允许 允许 无变化 如果所有设备保护标志都被清除，则为 IDLE。 其他没有变化。

SET\_PROT\_MASK 允许 不允许 不允许 不允许 允许 无变化 无变化

设备诊断

START\_DIAG 允许 不允许 不允许 不允许 不允许 DIAG IDLE

GET\_DIAG\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_DIAG\_STATUS允许 不允许 不允许 不允许 允许 无变化 无变化

SPI接口

CONFIG\_SPI 允许 不允许 不允许 不允许 允许 无变化 无变化

ECHO\_SPI 允许 不允许 不允许 不允许 允许 无变化 无变化

唤醒端口

CONFIG\_WUP 允许 不允许 不允许 不允许 允许 无变化 无变化

GET\_WUP\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_WUP\_STATUS允许 不允许 不允许 不允许 允许 无变化 无变化

SET\_WUP\_MASK 允许 不允许 不允许 不允许 允许 无变化 无变化

WUP 事件触发轮询

CONFIG\_WUP\_POLLING允许 不允许 不允许 不允许 允许 无变化 无变化

定时器触发轮询

CONFIG\_TIMER\_POLLING允许 不允许 不允许 不允许 允许 无变化 无变化

START\_TIMER\_POLLING 允许 不允许 不允许 不允许 不允许 无变化 PTIME = 0 的 PKE 状态。 PTIME > 0 的 POLLING/IDLE/PKE 状态的周期性重复

温度指示

CONFIG\_TEMP 允许 不允许 不允许 不允许 允许 无变化 无变化

GET\_TEMP\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_TEMP\_STATUS允许 不允许 不允许 不允许 允许 无变化 无变化

SET\_TEMP\_MASK 允许 不允许 不允许 不允许 允许 无变化 无变化

操作状态标志

GET\_OP\_STATUS 允许 允许 允许 允许 允许 无变化 无变化

CLEAR\_OP\_STATUS 允许 不允许 允许 不允许 允许 无变化 无变化

SET\_OP\_MASK 允许 不允许 不允许 不允许 允许 无变化 无变化

程序下载

DOWNLOAD\_PROG 允许 不允许 不允许 不允许 允许 无变化 无变化

START\_PROG 允许 不允许 不允许 不允许 允许 无变化 无变化

GET\_PROG\_SIG 允许 不允许 不允许 不允许 允许 无变化 无变化

## 命令参数限制

以下命令描述中使用的所有代表模拟特性的命令参数（例如电流、电压、阻抗、电感、品质因数、相位、时间、频率、比特率、温度和百分比）都是象征性的典型值，可能与给出的确切值有偏差 在参数说明中

# 通用命令

## GET\_VERSION

GET\_VERSION 提供版本标识符。

### 命令

表 30. GET\_VERSION 命令

LEN CMD CRC8

0x02 0xF0 CRC8

### 响应

表 31. GET\_VERSION 响应

LEN CMD 统计 参数 CRC8

0x0B 0xF0 STAT PI\_L PI\_H HW\_L HW\_H ROM\_L ROM\_H RAM\_L RAM\_H CRC8

PI：

PI 包含产品标识符。

表 32. PI（复位值 0xXXXX）

位 符号 访问 值 说明

15 到 0 PI[15:0] R 0xXXXX 产品标识符

HW：

HW 包含硬件版本标识符。

表 33. 硬件（复位值 0xXXXX）

位 符号 访问 值 说明

15 到 0 HW[15:0] R 0xXXXX 硬件版本

ROM：

ROM 包含 ROM 版本标识符。

表 34. ROM（复位值 0xXXXX）

位 符号 访问 值 说明

15 到 0 ROM[15:0] R 0xXXXX ROM 版本

RAM：

RAM 包含 RAM 版本标识符。 如果不执行 RAM 下载，则返回 RAM 版本号 0xFFFF。

表 35. RAM（复位值 0xFFFF）

位 符号 访问 值 说明

15 到 0 RAM[15:0] R 0xXXXX RAM 版本

### 状态标志

在故障情况下，根据事件或故障条件设置通用设备状态标志。

## CONFIG\_DEVICE

CONFIG\_DEVICE 配置设备参数。

### 命令

表 36. CONFIG\_DEVICE 命令

LEN CMD 参数 CRC8

0x03 0x09 DEVPAR CRC8

DEVPAR：

表 37. DEVPAR（默认值 0x01）

位 符号 访问 值 说明

7 到 2 RFU W0 保留供将来使用

1 PREAMB W 自动发送 NXP 前导码和代码违规模式

0 禁用

1 启用

0 VIO W VIO 电压电平

0 3.3 伏

1. 5 伏

PREAMB：

如果启用了 PREAMB，则每次驱动程序自动启动 LF 传输时，都会在传输更多数据之前发送 NXP 前导码和代码违规（同步）模式。 在前导之后发送的数据被附加而不中断数据流。

VIO：

如果请求更改电平，则在发送响应之前更改电压

### 响应

表 38.CONFIG\_DEVICE 响应

LEN CMD 状态 CRC8

0x03 0x09 状态 CRC8

### 状态标志

在故障情况下，根据事件或故障条件设置通用设备状态标志。

## CONFIG\_ADVANCED

设备参数的初始设置经过优化，可满足典型应用的需求。 对于某些设备参数，使其适应应用程序可能是有利的

因此，CONFIG\_ADVANCED 允许用更新的参数覆盖这些设备参数的初始配置值

### 命令

表 39. CONFIG\_ADVANCED 命令

LEN CMD 参数 CRC8

LEN 0xD2 {CAPARi {VALUEij}} CRC8

CAPARi, VALUEij：

CAPARi 选择与 VALUEij 适配的 CONFIG\_ADVANCED 参数

表 40. CONFIG\_ADVANCED 参数

CAPARi 函数 VALUEij 类型 描述

0x04 DRVMINTON VALUE8BIT LF 驱动器最小开启时间

0x12 CONFIG\_CHIRP CHIRPBLANK VALUE8BIT 啁啾消隐样本数

CHIRPSTEPINC VALUE8BIT 频率步进增量数

CHIRPSTEPAFTPK VALUE8BIT 峰值后的频率步进数

CHIRPSTEPBCK VALUE8BIT 跳回的频率步数

CHIRPCYCLE VALUE8BIT 每个频率的周期数

#### DRVMINTON

天线电流是在 D\* LF 驱动器开启期间测量的。 在电流检测电路中对镜像信号进行采样之前，需要稳定驱动脉冲。 此建立时间可能受 TX 引脚连接到 GND 的外部电容器的影响，具体取决于应用。

LF 驱动器最小接通时间的初始值经过优化以涵盖典型应用。 在某些应用程序中，更新 LF 驱动程序最小开启时间的默认值可能是有利的。 CONFIG\_ADVANCED DRVMINTON 允许覆盖 LF 驱动器最小开启时间的默认设置

表 41. DRVMINTON（复位值 0x05）

位 符号 访问 值 说明

7 to4 RFU W0 保留供将来使用

3to0 DRV\_MIN\_TON[3:0] W LF 驱动器最小导通时间

0000 200 纳秒

0001 250 纳秒

……

1110 900 纳秒

1111 950 纳秒

DRV\_MIN\_TON：

参数 DRV\_MIN\_TON 调整将执行有效天线驱动器电流测量的时间间隔（参考有效驱动器周期的中间）。 如果有效驱动周期等于或大于 2 x (DRV\_MIN\_TON + 100 ns)，则选择稳压 FB 反馈操作

#### CONFIG\_CHIRP

借助频率啁啾信号快速准确地确定天线谐振频率，该频率啁啾信号由快正向啁啾和慢速反向啁啾组成。

快进啁啾每 6 个周期取一个样本值，并将频率增加 2 个频率步长（步长 f = fC / 256 = 488.28125 Hz）。一旦检测到峰值，快进啁啾终止。

基于对谐振频率的第一次估计，慢速反向啁啾在快啁啾的终止频率附近开始。慢速反向啁啾在采样之前对每个频率应用 8 或 9 个周期。之后，频率缓慢回退（步长 f = fC / 256 = 488.28125 Hz），直到确定样本最大值。

啁啾信号经过优化以涵盖典型应用。优化是在考虑操作延迟时间和结果准确性的情况下完成的。在某些应用中，针对所应用的天线定制啁啾可能是有利的。

CONFIG\_ADVANCED 允许调整相关的啁啾信号参数。

啁啾：

啁啾信号开始时会出现过冲，这是由偏离谐振频率的频率驱动的天线引起的。 为避免基于此过冲错误地检测到峰值，默认情况下会丢弃第一个样本值（空白）。

空白的第一个样本的初始数量经过优化以涵盖典型应用。 在某些应用程序中，更新默认值可能是有利的。

因此，CONFIG\_ADVANCED CHIRPBLANK 允许覆盖空白第一个样本数的默认设置

表 42. CHIRPBLANK（复位值 0x02）

位 符号 访问 值 说明

7 到 3 RFU W0 保留供将来使用

2 到 0 CHIRP\_BLANK[2:0] W 啁啾消隐样本数

0x00 保留供将来使用

0x01 1

0x02 2

0x03 3

0x04 4

0x05 保留供将来使用

…

0x07 保留供将来使用

啁啾：

为了加快天线谐振频率测量，在快进啁啾期间，啁啾频率每 6 个周期增加 2 个频率步长（步长 f = fC / 256 = 488.28125 Hz）。

频率步长增量的初始数量经过优化以涵盖典型应用。

在某些应用程序中，更新默认值可能是有利的

因此，CONFIG\_ADVANCED CHIRPSTEPINC 允许覆盖频率步进增量数的默认设置。

请注意，配置值 0x01 会增加命令 MEAS\_ANT\_IMP\_ADVANCED 的执行时间

表 43. CHIRPSTEPINC（复位值 0x02）

位 符号 访问 值 说明

7 到 4 RFU W0 保留供将来使用

3 到 0 CHIRP\_STEPINC[3:0] W 频率步进增量数

0x00 保留供将来使用

0x01 1

…

0x0A 10

0x0B 保留供将来使用

…

0x0F 保留供将来使用

CHIRPSTEPFTPK：

如果啁啾信号幅度变为最大并且如果在以下频率设置下的信号值较小，则检测到天线谐振峰值。 可以配置在慢速反向啁啾期间检查第二个条件的频率步长数。

峰值后的初始频率步数经过优化以涵盖典型应用。 在某些应用程序中，更新此数字可能是有利的。

因此，CONFIG\_ADVANCED CHIRPSTEPAFTPK 允许覆盖峰值后频率步进数的默认设置。

请注意，配置大于 0x01 的值会增加命令 MEAS\_ANT\_IMP\_ADVANCED 的执行时间

表 44. CHIRPSTEPAFTPK（复位值 0x01）

位 符号 访问 值 说明

7 到 3 RFU W0 保留供将来使用

2 到 0 CHIRP\_STEPAFTPK[2:0] W 峰值之后的频率步进数

0x00 保留供将来使用

0x01 1

…

0x05 5

0x06 保留供将来使用

0x07 保留供将来使用

啁啾：

啁啾信号在少数周期内保持在一个频率设置。 因此，最大电流将在应用谐振频率后延迟出现。 在最大电流时，啁啾信号已经移动到更高的频率

为了补偿这种影响，啁啾频率在慢速反向啁啾开始之前跳回 5 个频率步长（步长 f = fC / 256 = 488.28125 Hz）。

初始频率反步数经过优化以涵盖典型应用。 在某些应用程序中，更新此数字可能是有利的。

因此，CONFIG\_ADVANCED CHIRPSTEPBCK 允许覆盖频率反步数的默认设置

表 45. CHIRPSTEPBCK（复位值 0x05）

位 符号 访问 值 说明

7 到 4 RFU W0 保留供将来使用

3 到 0 CHIRP\_STEPBCK[3:0] W 跳回的频率步数

0x00 保留供将来使用

0x01 1

…

0x05 5

…

0x0A 10

0x0B 保留供将来使用

…

0x0F 保留供将来使用

CHIRPCYCLE：

在用快进啁啾检测到峰值后，开始慢速反向啁啾。

每个频率的初始周期数经过优化以涵盖典型应用。 在某些应用程序中，更新此数字可能是有利的。

因此，CONFIG\_ADVANCED CHIRPCYCLE 允许覆盖每个频率周期数的默认设置。

请注意，推荐值为 0x08（与复位值不同），较大的值会增加命令 MEAS\_ANT\_IMP\_ADVANCED 的执行时间