

国芯微离线语音对接协议

国芯微离线语音对接协议

- 1.版本历史
- 2.协议概述
- 3.串口设置
- 4.协议交互流程
- 5.命令详情
 - 5.1.MCU 版本查询 (0x01): 模组-->MCU
 - 5.2.离线语音 MIC 配置 (0x02): 模组-->MCU
 - 5.3.离线语音 SPK 配置 (0x03): 模组-->MCU
 - 5.4.唤醒引脚设置 (0x04): 模组-->MCU【暂未支持】
 - 5.5.MIC 数据传输 (0x05): MCU-->模组
 - 5.6.录音模式开关 (0xF5): 模组-->MCU
 - 5.7.使用 MCU 进行 SPK 播放 (0x06): 模组-->MCU
 - 5.8.通知 MCU 模组播放状态 (0xF6): 模组-->MCU
 - 5.9.离线语音配置命令 (0x07): 模组-->MCU
 - 5.10.离线语音唤醒事件上报 (0x08): MCU-->模组
 - 5.11.离线语音休眠事件上报 (0x09): MCU-->模组
 - 5.12.离线语音芯片状态查询 (0x0A): 模组-->MCU【暂未支持】
 - 5.13.离线语音录音播音测试 (0x0B): 模组-->MCU
 - 5.14.离线语音音频回路测试 (0x0C): 模组-->MCU【暂未支持】
 - 5.15.离线语音 GPIO 测试 (0x0D): 模组-->MCU【暂未支持】
 - 5.16.设置自定义指令词 (0x11): 模组-->MCU
 - 5.17.MCU事件上报 (0x12): MCU-->模组
- 6.固件升级
 - 6.1.OTA - 双备份
 - 6.2.OTA - 裸片升级
- 7.FAQ
 - 7.1.麦克风增益设置
 - 7.2.播放音量设置
 - 7.3.校验和函数实现举例
 - 7.4.MCU事件上报 (0x12) 说明
 - 7.5.上行音频常见问题 (MCU->模组)
 - 7.6.下行音频常见问题 (模组->MCU)
 - 7.6.1.流控方式介绍 (推荐使用GPIO流控)
 - 7.6.2.帧格式版本说明 (SPK配置0x03命令)
 - 7.6.2.连续播放两条音频时, 播放丢字问题。
 - 7.6.3.播放数据格式参数
 - 7.6.4.ACK播放流程
 - 7.6.4.GPIO播放流程

1.版本历史

作者	日期	修订版本	修改描述
zhangyl	2025.09.18	1.3.7	修改vad灵敏度设置协议 增加环境声压阈值配置协议，可通过配置此阈值过滤远处的低噪人声 修改SPK播放（0x03）命令，扩展字段增加静音脚和电平配置 对唤醒，上报超时时间设置范围进行说明 修改自定义唤醒词协议（0x11），增加AEC阈值
zhangyl	2025.08.19	1.3.6	自定义唤醒词设置命令，修改子命令号为0x11，增加使用默认指令词控制字段 MCU上报事件命令，修改子命令号为0x12 增加对两种流控方式的说明 增加离线语音配置（0x07）命令新增示例控制方式，并对之前的示例控制方式进行优化，增加VAD复位命令
zhangyl	2025.08.05	1.3.5	SPK配置（0x03）命令修改，增加设置SPK播放协议版本字段； SPK播放（0x06）命令修改，增加新的传输格式，以支持传输多帧编码数据； MCU上报事件（0x12）命令修改，增加播放完成事件
zhangyl	2025.07.28	1.3.4	增加上报音频数据增益配置；增加自定义唤醒词设置命令；增加MCU事件上报命令
zhangyl	2025.07.04	1.3.3	波特率设置改为1M；增加FAQ
zhangyl	2025.06.21	1.3.2	修正协议部分版本号错误；mic+ref连通性测试协议修改；完善mic上报音频参数介绍
zhangyl	2025.06.05	1.3.1	增加录音模式开关协议；vad灵敏度设置；修改默认增益推荐设置为26dB；支持降噪配置
zhangyl	2025.06.03	1.3.0	修复 SPK 应答格式错误；增加常见配置命令示例
zhangyl	2025.05.29	1.2.2	增加离线语音配置命令说明；去掉唤醒和语音上报时间限制
zhangyl	2025.05.20	1.2.1	修改MIC配置范围，单位db；增加模组通知MCU自己的播放状态协议
zhangyl	2025.05.19	1.2.0	新增 mic、spk 开/关功能；唤醒事件、使能/失能vad、开关唤醒、查询状态
zhangyl	2025.05.17	1.1.0	SPK播放增加包序号
zhangyl	2025.05.15	1.0.0	首版

2.协议概述

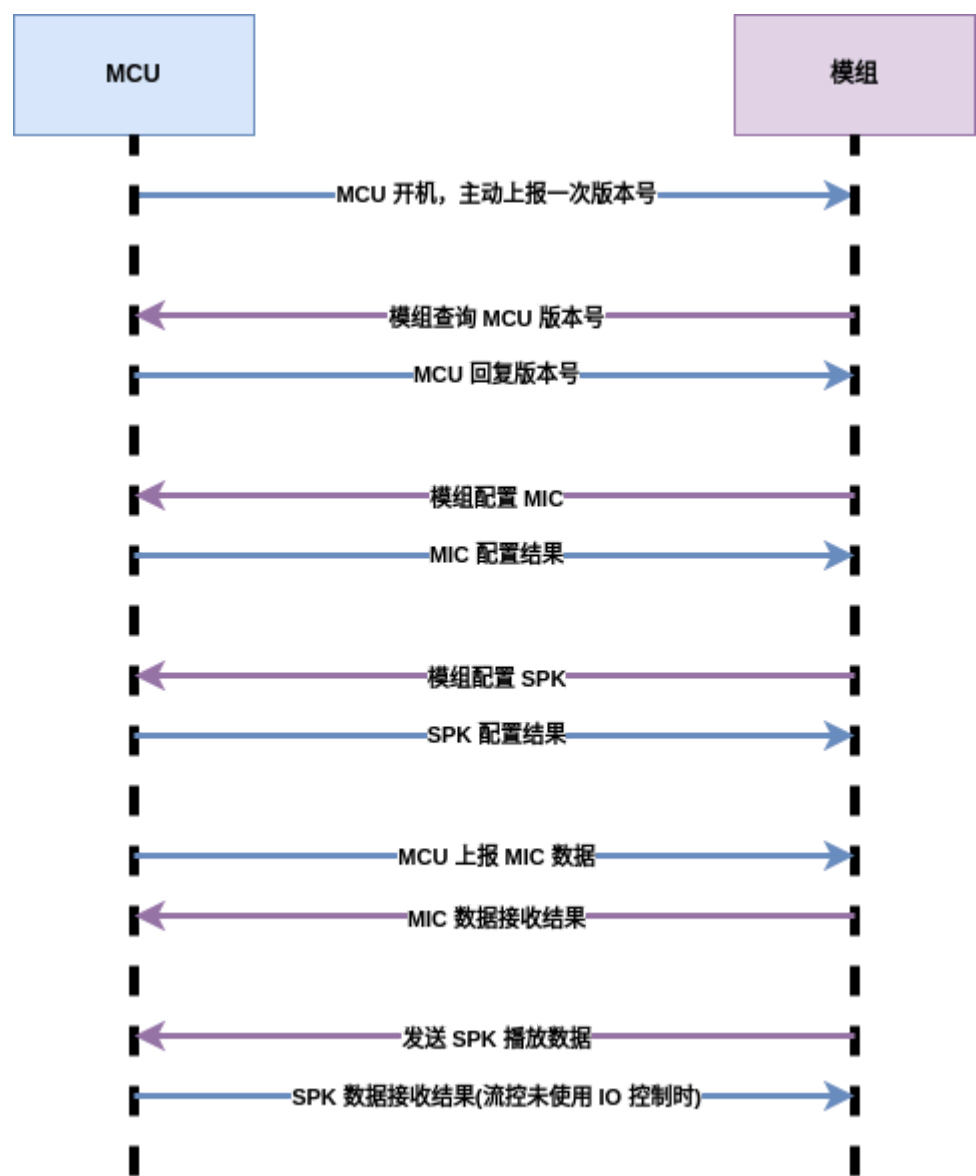
离线语音芯片（下称MCU）和 上位机模组（下称模组）通过串口协议进行通信。以下描述中大于1字节的数据均采用**大端传输**。该协议支持以下功能：

- MCU 版本查询
- MIC 配置
- 喇叭配置
- 唤醒引脚配置
- MIC 数据传输
- 喇叭播放数据传输
- 唤醒事件
- 使能/失能 VAD（语音活动检测）
- 开关唤醒
- 查询状态
- OTA

3.串口设置

- 波特率：1M
- 数据帧格式：8N1

4.协议交互流程



5.命令详情

5.1.MCU 版本查询（0x01）：模组-->MCU

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0001
数据	1	0x01(子命令)
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

• MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	4+N
数据	1	0x01(子命令)
	1	0：MCU上电首次发送；1：收到模组查询返回
	2	保留
	N	x.x.x（版本号为字符串，点分十进制形式，最大10字节，格式为 x.x.x，x 为 0-99，如1.0.0）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- 上电后，MCU 主动上报一次版本号，0 表示 MCU 上电首次发送，防止模组无法感知 MCU 重启；1 表示收到模组查询后返回。
- 如果模组回复成功，MCU 首次上电标志设置为 1，否则不设置。

• 模组回复 MCU 版本号（可选）

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x01(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

```
mcu<--模组
55 aa 00 92 00 01 01 93 #模组查询版本号
mcu-->模组
55 aa 03 92 00 09 01 01 ff ff 31 2e 33 2e 31 8e #mcu反馈固件版本号，为1.3.1
mcu<--模组
55 aa 00 92 00 02 01 00 94 #（可选）模组回复已成功获取版本号，也可以不发送
```

python脚本的 init_voice_control 函数提供了PC测试例子

5.2.离线语音 MIC 配置（0x02）：模组-->MCU

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x000D
数据	1	0x02(子命令)
	4	16000（采样率）
	1	16（采样位深）
	1	1（通道）
	1	26（MIC 增益：0-32，单位：dB）
	1	0：PCM；1：speex； 2：opus（音频格式，当前仅支持 opus 格式，40ms 或者 20ms 一帧。20ms一帧需要固件重新编译支持，并且是一次串口传输上报两帧）
	4	针对40ms：最小82字节，最大82*21字节（模组单次可接收的最大数据量，不包含协议固定字段，仅包含数据部分的长度，单位字节，调节的时候使用82 * n 来配置，n为整数） 针对20ms：最小84字节，最大84*21字节（模组单次可接收的最大数据量，不包含协议固定字段，仅包含数据部分的长度，单位字节，调节的时候使用84 * n 来配置，n为整数）

字段	长度	说明
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- MIC 增益默认设置26dB。
 - MCU 上报 16K，16bit，单通道的 opus 裸数据，每次上报40ms数据量。
 - MCU 单次传输的数据不超出设置的模组单次可接收的最大数据量，每次传输为完整帧数据。
 - MCU 会根据设置的模组单次可接收的最大数据量、当前的音频缓存，上报不同数据量音频，若模组接收不能及时处理，请设置此值为最小值82，这样基本保持在每40ms上报一次数据。如果设置为0，将默认使用520字节。mcu内部缓存音频为82*21字节。如果设置的值越小，那么接收延迟可能会越大。假如设置为82字节，那么接收延迟最大可能会有800ms
- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x02(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

```
mcu<--模组
55 aa 00 92 00 0d 02 00 00 3e 80 10 01 1a 02 00 00 07 0c 9e #配置mic音频输出opus压缩，16K
16bit 单声道，麦克风增益26dB，模组单词可接收的最大数据量为82*22字节
mcu-->模组
55 aa 03 92 00 02 02 00 98 #mic配置成功
```

python脚本的 configure_mic 函数提供了PC测试例子
可参考：《7.5.上行音频常见问题（MCU->模组）》有补充信息描述

5.3.离线语音 SPK 配置 (0x03)：模组-->MCU

- 模组发送

字段	长度	值	说明
帧头	2	0x55AA	固定帧起始标识
版本	1	0x00	协议版本号
命令字	1	0x92	标识协议类型
数据长度	2	0x000C 0x0011 ：需要设置帧格式版本V2或使用预留扩展时使用	数据域总长度（含子命令，不包含校验和）
数据域			
▸ 子命令	1	0x03	标识SPK配置指令
▸ 采样率	4	16000	音频采样率（单位：Hz）
▸ 采样位深	1	16	音频位深度（单位：bit）
▸ SPK音量	1	0-100	扬声器音量值
▸ 数据格式	1	0 ：PCM 2 ：Opus 1 ：Speex 3 ：MP3	
▸ 单次最大数据	4	1152	模组单次发送音频数据的最大长度（单位：字节）
▸ 帧格式版本（可选）	1	1 ：V1（此字段不存在时默认使用此版本） 2 ：V2 用于后续 SPK播放指令 (0x06) 的数据格式解析	仅当数据长度!=0x000C时存在
▸ 功放静音管脚配置（可选）	1	MCU实际使用的功放静音管脚	未设置默认使用P7
▸ 静音管脚静音电平（可选）	1	0：静音时为低电平 1：静音时为高电平	未设置默认0
▸ 预留扩展（可选）	2	0	仅当数据长度!=0x000C时存在
校验和	1	SUM%256	帧头到数据域结束的字节累加和，对256取余

说明：

- opus播放时不可以带头信息传输，使用采样率16K，码率16kbps，单通道。
 - PCM格式播放时必须使用GPIO流控方式。
 - 使用V1帧格式版本，模组每次只能推一帧的数据量给MCU。
 - 使用V2帧格式版本，模组每次可根据需求推送多帧数据给MCU，主要为解决ACK方式播放卡顿问题增加，若当前播放无问题可忽略。
- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0006
数据	1	0x03(子命令)
	1	执行结果：0x00（成功）；0x01（SPK 单次发送最大数据长度协商失败）；0x02（硬件配置失败）
	4	512（若模组给 MCU 的最大单次发送数据 MCU 无法接受，返回错误，并返回可接受的大小。成功时可无此字段，仅 0x01 时有效）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

```
mcu<--模组
55 aa 00 92 00 0c 03 00 00 3e 80 10 64 02 00 00 02 00 d6 #模组设置SPK 16K 16bit 单声道，opus
格式音频，每次传输最大512字节
mcu-->模组
55 aa 03 92 00 06 03 00 00 00 02 08 a7 #SPK配置成功
```

python脚本的 configure_spk 函数提供了PC测试例子

如果需要同时支持opus和pcm播放，那么播放相应音频格式的时候，需要先使用该命令配置正确 SPK

可参考：《7.6.下行音频常见问题（模组->MCU）》有补充信息描述

5.4.唤醒引脚设置（0x04）：模组-->MCU【暂未支持】

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	N+1
数据	1	0x04(子命令)
	N	唤醒引脚配置（JSON 格式字符串），如 P1 上升沿唤醒，P2 高电平维持500ms唤醒： <code>{"wk_list": [{"io": 1, "trigger": "edge", "mode": "rise"}, {"io": 2, "trigger": "level", "mode": "high", "time_ms": 500}]}</code> ，包含字符串结尾 <code>\0</code>
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x04(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.5.MIC 数据传输（0x05）： MCU-->模组

- MCU 发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	2+N
数据	1	0x05(子命令)
	1	0：收音开始（vad_start）；1：收音数据（vad_data）；2：收音结束（vad_end）
	N	MIC 采样数据
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- MIC 采样数据仅在 vad_data（1）指令中传输，vad_start（0）和 vad_end（2）无采样数据。
 - MIC 数据传输仅在使用 UART 时使用，传输通道配置在离线语音配置命令（0x07）中设置。
- 模组返回（可选）

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x05(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

```

mcu-->模组
55 aa 03 92 00 02 05 00 9b #收音开始 (vad_start)
mcu-->模组
55 aa 03 92 00 52 05 01 bb 42 01 7a 2e 10 eb 04 b9 a4 ef 3b 3b 7b 89 30 9a 11 a9 6e f4 61
6a de 9a 09 2a 6e e0 fa 94 78 7e c3 9f 61 3f 39 32 fa 35 08 06 e1 28 4d 5b ff e8 d5 76 2f
33 d5 8f 8c 9e 64 5e c1 fb a1 58 80 2d f1 44 a3 40 42 99 c7 bc 6b ec 30 a0 e9 31 00 dd #收
音数据 (vad_data)
mcu-->模组
55 aa 03 92 00 02 05 02 9d #收音结束 (vad_end)

```

python脚本的 record 函数提供了PC测试例子，可以支持40ms和20mc opus的音频录音

```

# 仅录音功能测试
response = send_command_and_wait(ser, 0x00, 0x92, 0x07, bytes([0x01, 0x01]),
timeout=1) # 开mic
print("MIC开关设置:", "成功" if response and response[2][0] == 0x07 and response[2][2]
== 0x00 else "失败")
record(ser)

```

5.6.录音模式开关（0xF5）：模组-->MCU

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	3
数据	1	0xF5(子命令)
	1	0：关闭录音模式（退出仅上传MIC数据）；1：开启录音模式（进入仅上传MIC数据）
	1	[0, 3] 降噪等级，值越大降噪越明显
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- 进入此模式后不再响应其它命令，需要在接收到退出命令后才能正常接收其它命令，因此需要提前设置超时时间的有关参数（0x07）。
- 进入此模式会将唤醒，VAD功能自动关闭，MIC 打开，将直接开始上报数据。
- 退出此模式会将唤醒，VAD功能，MIC 自动打开，若降噪等级需要更改或有其它需求记得重新配置。

- MCU 的 MIC 数据上报使用 0x05 命令
- 示例:
 - 开启录音模式, 设置降噪等级0命令 `55aa00920003f501008a`
 - 关闭录音模式, 设置降噪等级0命令 `55aa00920003f5000089`
- MCU返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0002
数据	1	0xF5(子命令)
	1	执行结果: 0x00 (成功); 0x01 (失败)
校验和	1	从帧头开始, 按字节求和, 得出的结果对 256 求余

5.7.使用 MCU 进行 SPK 播放 (0x06): 模组-->MCU

- 模组发送
 - V1版本。由于 Opus 编码帧为变长格式, 为明确界定帧边界, 此版本规定每个数据包仅可传输一个帧 (不可包含多帧)。

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	6+N
数据	1	0x06(子命令)
播放状态	1	0: 开始播放; 1: 播放数据; 2: 停止播放
	4	包序号 (自增, 每次从0开始)
	N	SPK 播放数据
校验和	1	从帧头开始, 按字节求和, 得出的结果对 256 求余

- V2版本。**需要SPK配置 (0x03) 命令设置使用此版本**。解决V1版本每个数据包只能传输一帧的限制, 支持一个数据包包含多个Opus编码帧。

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	6+N
数据	1	0x06(子命令)
播放状态	1	0：开始播放；1：播放数据；2：停止播放
	4	包序号（自增，每次从0开始）
	1	包含的播放帧个数
	-	每个播放帧包含以下字段（重复上字节设置次数）
	2	播放帧数据长度
	N	SPK 播放数据
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- 播放状态：
 - 开始播放（0）：不能带 SPK播放数据。MCU 收到该包进入播放准备状态。
 - 播放数据（1）：带 SPK播放数据。MCU 接收该数据进行播放。
 - 停止播放（2）：不能带 SPK播放数据。MCU 收到该包退出播放准备状态。
 - 备注：一般云端下发 **tts start**，就可以发送 **开始播放（0）命令**，云端下发 **tts stop**，需要将本地缓存的音频全部发送给 MCU 后，再发 **停止播放（2）命令**。
- SPK 播放数据仅在使用 UART 传输时使用，需根据流控 IO 或 ACK 控制发送频率。传输通道在**离线语音配置命令（0x07）**中设置。
- 常见问题可见 7.5
- MCU返回（ACK响应，流控IO未开启时使用）：

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	6
数据	1	0x06 (子命令)
	1	执行结果：0x00 (成功)；0x01 (失败)

字段	长度	说明
	4	包序号（本次接收到的包序号）
校验和	1	从帧头开始，按字节求和，结果对256求余

- 说明：
- 流控IO未使用时才会返回，应答失败时可以延时播放帧长（例如每帧40ms时可以延时40）发送下一帧数据。
 - 流控IO使用时，会通过切换流控IO电平状态控制模组发送频率（模组判断IO拉高表示可以发送，IO拉低表示不能发送）。

5.8.通知 MCU 模组播放状态（0xF6）：模组-->MCU

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	2+N
数据	1	0xF6(子命令)
	1	0：开始播放；2：停止播放
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

- 说明：
- 通知 MCU 当前模组的播放状态。用于喇叭接在模组上，或者喇叭接在MCU上，但是通过i2s播放，并同时需要MCU做 AEC 算法处理的场景下。MCU会根据这个通知来开启或者关闭AEC算法。

- MCU返回：

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	2
数据	1	0xF6 (子命令)
	1	执行结果：0x00 (成功)；0x01 (失败)

字段	长度	说明
校验和	1	从帧头开始，按字节求和，结果对256求余

5.9.离线语音配置命令（0x07）：模组-->MCU

- 模组发送

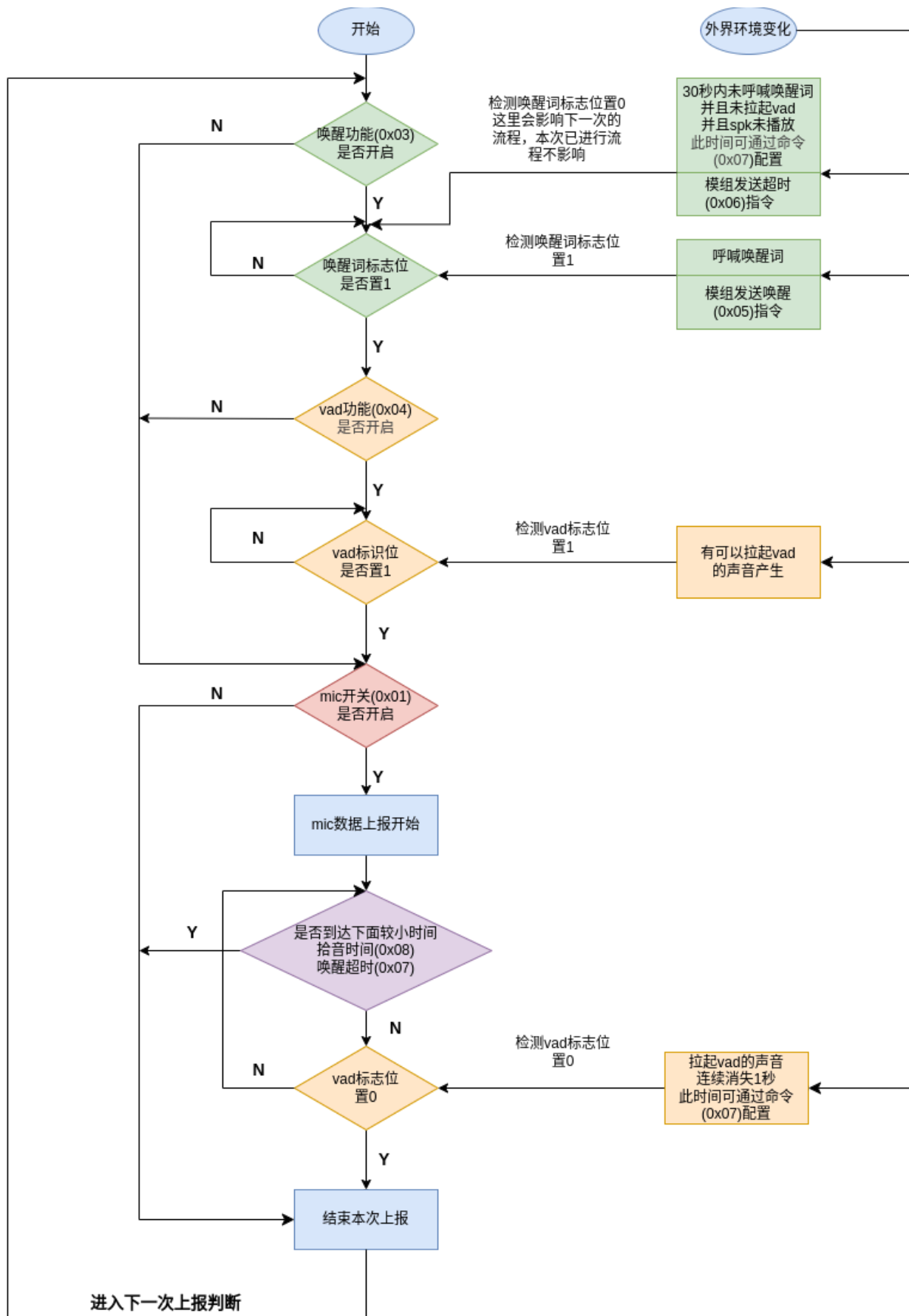
字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	2+N
数据	1	0x07(子命令)
	1	配置命令
	N	配置数据，见下面的详细描述
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

配置命令和配置数据：

配置命令	配置数据	备注
数据传输通道设置（0x00）	0x00（uart）	默认使用 UART，I2S暂不支持
	0x01（I2S）	
MIC 开关控制（0x01）	0x00（关闭）	默认开启状态
	0x01（开启）	
SPK 开关控制（0x02）	0x00（关闭）	默认开启状态
	0x01（开启）	
唤醒词开关控制（0x03）	0x00（关闭）	默认开启状态
	0x01（开启）	
VAD 控制（0x04）	0x00（关闭）	默认开启状态，使用固件默认 VAD 超时时间1秒

配置命令	配置数据	备注
	0x01 开启并使用固件默认时间	
	> 0x01 开启并设置时间长度1Byte	设置静默帧数触发超时（1帧=40ms）
唤醒通知 (0x05)	0x01	唤醒（模组通知语音芯片设备被唤醒）
超时通知 (0x06)	0x01	超时（模组通知语音芯片已超时）
设置静默超时时间（0x07）	1或4Byte大端 取值范围 1到4000000	默认30s，超时需重新唤醒 设置数据需大于0，小于4000000，长度1字节或4字节 意味着在没有唤醒，vad为0,未播放音频超过30s后，下一次需要唤醒才可以上报数据
设置最大拾音时间（0x08）	1或4Byte大端 取值范围 1到4000000	默认30s，超时停止拾音 设置数据需大于0，小于4000000，长度1字节或4字节 意味着最长连续上报30s数据
VAD灵敏度 (0x09)	取值范围0-100，2字节	数值越大，灵敏度越低。默认值可能随固件版本变化。
	第1字节设置 播放 时灵敏度	
	第2字节设置 非播放 时灵敏度	
降噪设置 (0x0A)	0-3（代表降噪等级）	开启降噪，并设置降噪等级，默认降噪等级0，越大降噪越明显
	0xFF（关闭降噪）	关闭降噪
ASR音频增益设置（0x0B）	>0，1Byte，单位dB	设置上报的音频数据放大多少dB，默认12dB
VAD复位 (0x0C)	0x01	vad复位，将重置当前vad状态为0，若当前正在上传将停止录音
设置环境声压阈值（0x0D）	>0，1Byte，单位dB	默认35dB，稍微有声音即可达到，可通过配置此阈值过滤远处的低噪人声。VAD检测流程是先经过VAD模型，再经过环境声压过滤，最终再给到应用层

说明：



- 语音上报对应有三级功能控制，分别是Mic总开关(0x01)，唤醒功能(0x03)，Vad功能(0x04)。可根据协议进行控制开关。
- 语音上报是需要达到一定条件才会上传的，在三个控制全部打开时，优先级由高到低排序为：Mic开关 -> 唤醒标志 -> Vad标志。

- Vad和唤醒标志，会根据外界环境动态变化。可根据协议进行超时时间配置，Vad超时配置命令(0x07)，唤醒超时配置命令(0x05)。
- 可以根据上面流程图自行控制 MCU 语音上报
- 控制示例:【未设置时使用的上面表格中的默认参数，或本次程序运行已设值】
 - 1.想要唤醒后直接上报音频，并持续上报音频60秒，可以进行如下配置：
 - 注：若唤醒超时时间小于最大拾音时间，唤醒超时后本次拾音完就会结束，达不到最大拾音时间。
 - 55aa0092000307083cdf # 最大拾音时间60s
 - 55aa0092000307073cde # 唤醒超时时间60s
 - 55aa009200030701019d # 打开Mic总开关
 - 55aa009200030704009f # 关闭Vad功能
 - 55aa009200030703019f # 打开唤醒功能
 - 2.按键长按对话模式：若外部有一个按键，在按键长按时上报数据对话，且最长持续上报60秒。
 - 55aa0092000307083cdf # 设置最大拾音时间60秒
 - 55aa009200030701009c # 关闭mic
 - 55aa009200030703009e # 关闭唤醒
 - 55aa009200030704009f # 关闭vad
 - 55aa009200030701019d # 长按按键时打开mic
 - 55aa009200030701009c # 松开按键时关闭mic。
 - 3.按键对话模式：若外部有一个按键，需每次按键，vad拉起后上报数据对话，间隔1s未说话vad下去，按键后5秒没有说话退出唤醒状态。
 - 55aa00920003070705a7 # 设置静默（唤醒）超时5s
 - 55aa009200030701009c # 关闭mic
 - 55aa00920003070419b8 # 设置打开vad并设置由拉起到下去的间隔1s
 - 55aa009200030703009e # 关闭唤醒
 - 55aa009200030701019d # 按键按下时打开mic
 - 55aa009200030701009c # 语音上报结束后，即收到 vad上报 end 时，关闭mic
 - 4.唤醒对话模式：若要求每次对话都需要唤醒，vad拉起后上报数据对话，间隔1s未说话vad下去，唤醒后5秒没有说话退出唤醒状态。
 - 55aa00920003070705a7 # 设置静默（唤醒）超时5s
 - 55aa00920003070419b8 # 设置打开vad并设置拉起到下去的间隔1s
 - 55aa009200030703019f # 打开唤醒
 - 55aa00920003070601a2 # 设置超时通知
 - 55aa009200030701019d # 打开mic
 - 55aa00920003070601a2 # 语音上报结束后，即收到 vad上报 end 时，发送超时，退出唤醒状态
 - 5.随意对话模式：若要求首次需要唤醒，唤醒后通过vad拉起上报数据对话，间隔1s未说话vad下去，唤醒后30秒没有说话退出唤醒状态。

- 55aa0092000307071ec0 # 设置静默（唤醒）超时30s
- 55aa00920003070419b8 # 设置打开vad并设置由拉起到下去的间隔1s
- 55aa009200030703019f # 打开唤醒
- 55aa00920003070601a2 # 设置超时通知
- 55aa009200030701019d # 打开mic
- 6.唤醒问答模式: 第一次需要唤醒，回答期间可使用唤醒词打断，一问一答方式。
 - 55aa0092000307071ec0 # 设置唤醒超时30s
 - 55aa00920003070419b8 # 打开vad，并设置vad间隔1s
 - 55aa009200030703019f # 打开唤醒
 - 55aa00920003070601a2 # 设置超时通知
 - 55aa009200030701019d # 打开mic
 - 55aa00920003070601a2 # 语音上报结束，即收到 vad上报 end 时，设置超时通知
 - 55aa009200030701009c # 播报开始时关闭mic
 - 55aa009200030701019d # 播报结束时打开mic
 - 55aa00920003070501a1 # 播报结束后主动通知唤醒
 - 55aa009200030701019d # 唤醒词打断后主动打开mic
- 7.vad灵敏度控制:
 - 55aa0092000307092dd1 # 设置vad阈值为45

• MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0003
数据	1	0x07(子命令)
	1	配置命令
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.10.离线语音唤醒事件上报（0x08）：MCU-->模组

• MCU 发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	1+4
数据	1	0x08(子命令)
	4	唤醒词条的事件ID
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- 唤醒词触发后，MCU 通过此指令通知模组。
- 模组返回（可选）

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x08(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.11.离线语音休眠事件上报（0x09）：MCU-->模组

- MCU 发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	1+4
数据	1	0x09(子命令)
	4	休眠事件ID（例如：15）

字段	长度	说明
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- 语音芯片超时休眠后，通过此指令通知模组。主要用在短指令的生效时间上。比如先喊唤醒词，然后才能喊2级短指令词，持续时间15秒。15秒超时后，要继续喊短指令必须先喊唤醒词。这个15秒超时到了就会上报该语音休眠事件。
- 模组返回（可选）

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x09(子命令)
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.12.离线语音芯片状态查询（0x0A）：模组-->MCU【暂未支持】

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0001
数据	1	0x0A(子命令)
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03

字段	长度	说明
命令字	1	0x92
数据长度	2	0x0005
数据	1	0x0A(子命令)
	4	状态标志位（大端）
	1	执行结果：0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

状态标志位定义：

Bit位	配置数据	备注
Bit0	MIC开关状态	0：关闭；1：开启
Bit1	唤醒开关状态	0：关闭；1：开启
Bit2	VAD是否启动	0：关闭；1：开启

5.13.离线语音录音播音测试（0x0B）：模组-->MCU

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x0B(子命令)
	1	N（录音时间，单位秒）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- MCU 收到该命令后启动录音，录完自动播放，用于工厂测试，可通过按键触发。
- MCU 返回

字段	长度	说明
帧头	2	0x55aa

字段	长度	说明
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0001
数据	1	0x0B(子命令)
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.14.离线语音音频回路测试（0x0C）：模组-->MCU【暂未支持】

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0003
数据	1	0x0C(子命令)
数据	1	0x00（麦克通道）；0x01（回采通道）；0xFF（结束测试）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

说明：

- MCU 收到该命令后启动音频回路测试，播放测试时长的测试通路音频，会在播放完成后返回。

- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x0C(子命令)
	1	0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.15.离线语音 GPIO 测试（0x0D）： 模组-->MCU【暂未支持】

- 模组发送

字段	长度	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x0D(子命令)
数据	1	0x01（(T1，测试5-8分组和GX_P10是否接地)）0x02（T2，仅测试GX_P10是否接地)
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

- MCU 返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	0x0002
数据	1	0x0D(子命令)
	1	0x00（成功）；0x01（失败）
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

5.16.设置自定义指令词（0x11）： 模组-->MCU

- 模组发送

字段	长度	值	说明
帧头	2	0x55AA	固定帧起始标识
版本	1	0x00	

字段	长度	值	说明
命令字	1	0x92	
数据长度	2	1+N	数据总长度（含子命令，不包含校验和）
子命令	1	0x11	标识自定义指令词设置
	1	0x00 0x01	标识是否使用默认的唤醒词 为0代表当前设置指令词将覆盖掉固件默认的词，只会有当前设置的词 为1代表当前设置指令词+固件默认的词
	4	0x00	预留扩展
	2	0x000N	设置的指令词个数（N）
	-	-	每个词条包含以下字段（重复 N 次）
	4	100-8191	事件ID ，唤醒后通过0x08命令上报
	4	<1000	阈值 ，非AEC下触发唤醒的阈值（需脚本生成）
	4	<1000	AEC阈值 ，AEC下触发唤醒的阈值（需脚本生成）
	4	0/1	类型 ，0=普通指令词, 1=主唤醒词
	1+X	<39字节	指令词 ，结构: [长度 X] + [UTF-8字符串]（无结束符）
	1+Y	≤16个元素	标签 ，结构: [长度 Y] + [short数组]（如[14,38,...]，需脚本生成）
校验和	1	SUM%256	帧头到数据域结束的字节累加和，对256取余

示例：

```
55aa009200641100000000000000200000006500000349000001a4000000010ce5b08fe5a5bde5b08fe5a5bd10000e0026000b0025000e0026000b002500000066000003380000019c000000010ce5b08fe5a5bde5908ce5ada610000e0026000b002500060033000e0023e6
```

- 将不使用默认指令词，重置唤醒词列表，设置成下面两个唤醒词

```
[USER_KWS]default_enable:0
[USER_KWS]set user_kws_num:2
[UDKL]Demo Kws List [Total:2]:
[UDKL]KWS: 小好小好 | KV: 101 | TRH: 841 | AEC_TRH: 420 | Major: 1 | L: [14 38 11 37 14 38 11 37 ]
[UDKL]KWS: 小好同学 | KV: 102 | TRH: 824 | AEC_TRH: 412 | Major: 1 | L: [14 38 11 37 6 51 14 35 ]
```

说明：

- MCU 不记录唤醒词，**掉电、每次设置命令都将重置指令词列表**，需要主控模组保存唤醒词的数据，发

- 给MCU。
- MCU返回

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	6
数据	1	0x11 (子命令)
	1	执行结果：0x00 (成功)；0x01 (失败)
校验和	1	从帧头开始，按字节求和，结果对256求余

5.17.MCU事件上报（0x12）：MCU-->模组

- MCU 发送（模组是否使用可选）

字段	长度	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x92
数据长度	2	1+4
数据	1	0x12(子命令)
	4	事件定义值
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

事件定义值定义：（具体可见7.6章节）

- 0x01：播放开始
- 0x02：SPK FIFO 播放空
- 0x03：播放结束（收到模组SPK 停止播放事件才反馈）

6.固件升级

当前固件升级有两种方式

- 裸片升级：支持空片升级，类似PC的固件烧录工具。按住BOOT键上电复位，进入升级模式，此时进行串口交互来升级固件。

- 双备份：不支持空片升级，需要预先在flash里烧录好支持OTA升级功能的固件。默认会将 8006 的 2M Flash 分成两份，存储两份固件，启动时判断固件的编译日期，启动最新编译日期的固件。

6.1.OTA - 双备份

- 概述

OTA（Over-the-Air）即空中下载技术，通过网络远程为设备更新和升级软件程序。

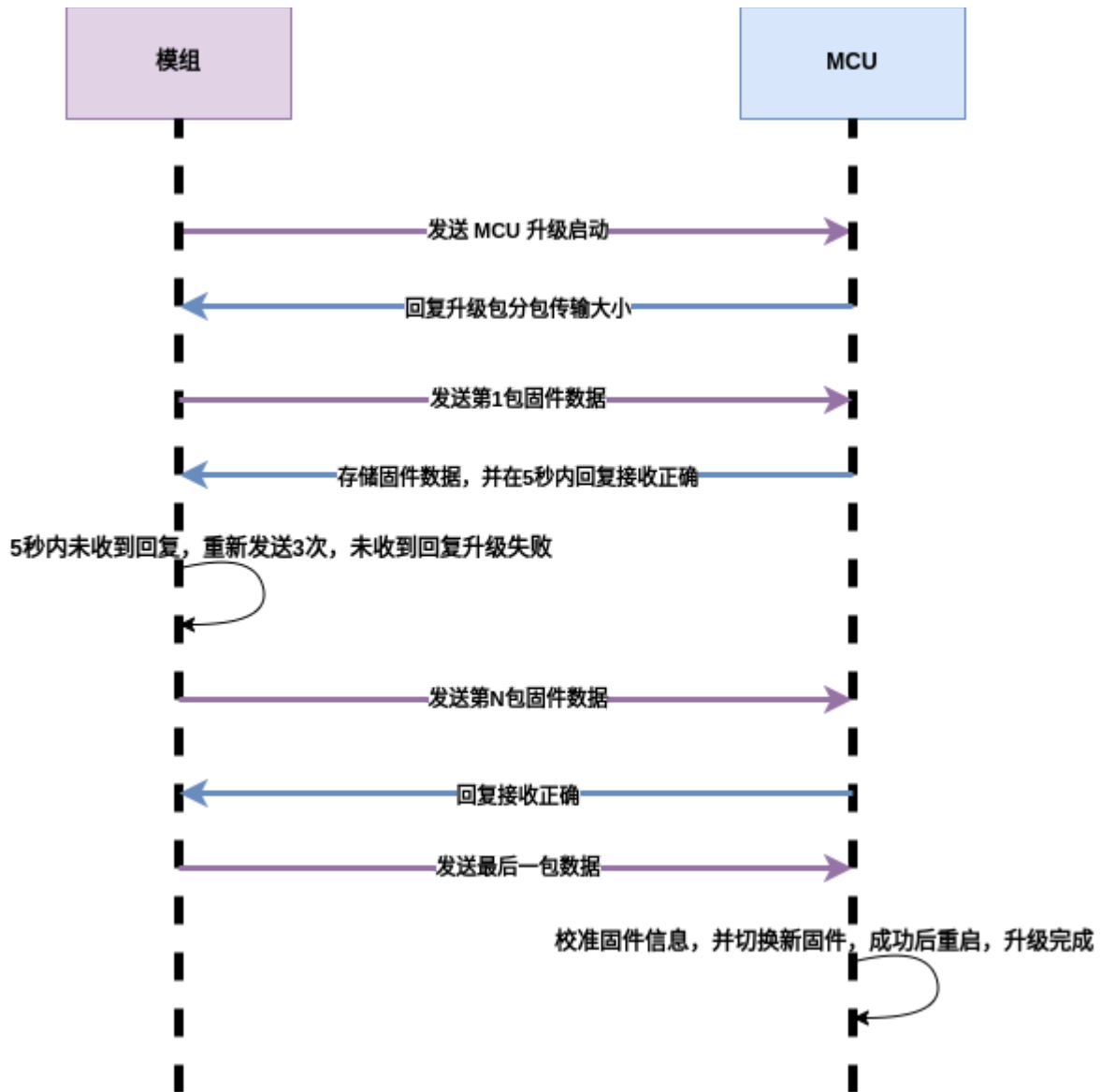
支持 MCU OTA 功能。先将需要更新的固件文件上传至服务器，然后 Wi-Fi 模组通过OTA协议对文件进行分包传输，最后 MCU 接收升级包并写入本地闪存，最终实现固件的升级。

Zone	Addr	Length
bootloader	0x0 ~ 0x0fff	4KB
app bank A	0x1000	1020K
app bank B	0x100000	1020K

具体flash区域规划，可以参考：`ovp\ovp_aiot\ovp\common\ota\app_bin_hdr.h`

- 功能流程

MCU OTA 功能需要 Wi-Fi 模组支持，设备配网成功后链接服务器，通过平台配置或升级检测，可以启动 OTA 流程。升级流程如下：



Wi-Fi 模组发送完所有的升级包，发送版本号查询指令 0x01，MCU 需要在一分钟回复产品信息中的 MCU 软件版本号（升级后的版本号），版本号需要和升级的版本号保持一致。

可以使用开发包里的python测试脚本“wifi_uart_pc_script”进行OTA的测试。

```
# 双备份OTA测试，注意 WIFI 大模型包中固件默认不支持
firmware = "app.bin"      # 编译输出的app.bin路径
test_ota(ser, firmware)   # 正常升级
```

支持OTA功能的固件，需要在编译8006/8005固件的时候，在 make menuconfig 里勾选“Enable dual bank upgrade”。OTA升级的固件，必须是编译出的 app.bin 文件，它不包含 bootloader信息。

升级效果如下：

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

```
RESTART: G:\1-GX8006开发包\4-GX8006标准固件以及串口协议移植指南
\wifi_uart_pc_script_v1.1.0\main.py
输出目录准备就绪
获取版本结果: 1 1
获取版本号: 1.3.0
MIC 配置成功
SPK 配置成功
SPK 配置成功
开始OTA升级测试...
OTA启动成功, 分包大小: 1024字节
进度: 1024/767240 (0.1%)
进度: 2048/767240 (0.3%)
进度: 3072/767240 (0.4%)
进度: 4096/767240 (0.5%)
进度: 5120/767240 (0.7%)
进度: 6144/767240 (0.8%)
进度: 7168/767240 (0.9%)
进度: 8192/767240 (1.1%)
进度: 9216/767240 (1.2%)
进度: 10240/767240 (1.3%)
进度: 11264/767240 (1.5%)
进度: 12288/767240 (1.6%)
进度: 13312/767240 (1.7%)
进度: 14336/767240 (1.9%)
进度: 15360/767240 (2.0%)
进度: 16384/767240 (2.1%)
进度: 17408/767240 (2.3%)
进度: 18432/767240 (2.4%)
进度: 19456/767240 (2.5%)
进度: 20480/767240 (2.7%)
进度: 21504/767240 (2.8%)
进度: 22528/767240 (2.9%)
```

注意：使用PC烧录工具烧录了支持OTA双备份的固件，如果下次还要用PC烧录工具烧录固件，要先把flash全部擦除再烧录，避免启动时编译日期判断机制导致运行的不是烧录的固件。

说明

- MCU OTA 升级成功条件是固件包传输完成，设备重启上报新的版本号，两者同时完成才会提示成功。
- 指令列表

OTA 流程涉及以下协议指令：

命令字	命令说明
0x0a	MCU 升级启动（升级包大小通知）
0x0b	MCU 升级包传输

- MCU 升级启动（升级包大小通知）

启动 OTA 升级时，模组首先使用命令 `0x0a` 通知 MCU 固件包的总大小，此时 MCU 需回复模组固件分包传输大小。

- 升级包分包传输大小

回复升级包分包传输值	升级包分包大小
<code>0x00</code>	默认 256 字节
<code>0x01</code>	512 字节
<code>0x02</code>	1024 字节

- 模组发送

字段	字节数	说明
帧头	2	<code>0x55aa</code>
版本	1	<code>0x00</code>
命令字	1	<code>0x0a</code>
数据长度	2	<code>0x0004</code> 或 <code>0x0008</code>
数据	4	固件包字节数，unsigned int，大端
数据	1	MCU 类型： <code>10~19</code>
数据	3	预留扩展
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

`55 aa 00 0a 00 04 00 00 68 00 75`

表示固件包长度 26624，即 26KB。

- MCU 返回

字段	字节数	说明
帧头	2	<code>0x55aa</code>
版本	1	<code>0x03</code>
命令字	1	<code>0x0a</code>
数据长度	2	<code>0x0001</code>
数据	1	分包传输大小： <code>0x00</code> ：256 字节 <code>0x01</code> ：512 字节 <code>0x02</code> ：1024 字节
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

55 aa 03 0a 00 01 00 0d

表示 MCU 支持 256 字节分包传输。

- MCU 升级包传输
模组通过该指令按包传输固件包给到 MCU，MCU 接收到传输分包进行接收处理。若 MCU 5s 内未回复模组，模组将进行重发，若 3 次重发后仍未收到 MCU 回复，则认为升级失败。
- 升级包传输数据格式：包偏移 + 包数据。
- MCU 若收到该帧数据长度为 4 字节，并且包偏移 $\geq$ 固件大小，则包传输结束。
- 模组发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0b
数据长度	2	0x0004+N
数据	4+N	- data[0]-data[3]：包偏移（固定） - data[4]-data[n]：数据包内容
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

对于固件大小 530 字节（最后一包可不回复）：

- 第一包，包偏移 0x00000000，数据包长度 256 字节：
55 aa 00 0b 01 04 00000000 xx...xx XX
- 第二包，包偏移 0x00000100，数据包长度 256 字节：
55 aa 00 0b 01 04 00000100 xx...xx XX
- 倒数第二包，包偏移 0x00000200，数据包长度 18 字节：
55 aa 00 0b 00 16 00000200 xx...xx XX
- 最后一包，包偏移 0x00000212，数据包长度 0 字节：
55 aa 00 0b 00 04 00000212 xx...xx XX

- MCU 返回

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x0b
数据长度	2	0x0000

字段	字节数	说明
数据	0	无
校验和	1	从帧头开始，按字节求和，得出的结果对 256 求余

示例：

```
55 aa 03 0b 00 00 0d
```

6.2.OTA - 裸片升级

请访问AI文档中心：<https://document.nationalchip.com/software/fornax/SDK%E5%BC%80%E5%8F%91%E6%8C%87%E5%8D%97/%E6%A0%B7%E4%BE%8B%E5%8F%82%E8%80%83%E6%8C%87%E5%8D%97/%E4%B8%B2%E5%8F%A3%E8%A3%B8%E6%9C%BA%E5%8D%87%E7%BA%A7%E5%8F%82%E8%80%83%E4%BE%8B%E7%A8%8B/>

7.FAQ

7.1.麦克风增益设置

使用离线语音 MIC 配置 (0x02) 命令，修改 MIC 增益。

7.2.播放音量设置

使用离线语音 SPK 配置 (0x03)：模组-->MCU命令，修改 SPK 音量。

7.3.校验和函数实现举例

```
/**
 * @brief 计算校验和
 *
 * @param data 字节数组指针（包含帧头、版本、命令字、数据长度和数据等完整内容）
 * @param len 参与计算的字节长度（从帧头到最后一个数据字节）
 * @return uint8_t 计算出的校验和（0-255范围）
 */
uint8_t calculate_checksum(const uint8_t *data, uint16_t len) {
    uint16_t sum = 0; // 使用16位避免溢出

    for (uint16_t i = 0; i < len; i++) {
        sum += data[i]; // 累加所有字节
    }

    return (uint8_t)(sum % 256); // 对256求余返回校验和
}
```

7.4.MCU事件上报（0x12）说明

事件定义值定义：

- 播放事件
 - 0x01：播放开始，代表MCU真正的播放开始，因为SPK数据发送在MCU需要缓存一些数据，故模组发送spk start后，MCU还未真正的开始播放。
 - 0x02：播放空，代表MCU还未接收到模组发送的spk end（一条音频的结束），但已经播空。
 - 0x03：播放结束，代表MCU真正的播放结束，因为SPK数据发送在MCU缓存了一些数据，故模组发送spk end后，MCU未真正的结束播放。
- 播放事件使用举例
 - 连续播放两条音频，二次播放间隔较短时，发送的第二条音频的 开始播放（0） 指令将清空缓存导致播放丢字。这时模组等待播放结束事件上报后再下发下一条可避免。

7.5.上行音频常见问题（MCU->模组）

- MIC 增益默认设置26dB。
- MCU 上报 16K，16bit，单通道的 opus 裸数据，每次上报40ms数据量。
- MCU 会根据设置的模组单次可接收的最大数据量、当前的音频缓存，上报不同数据量音频，若模组接收不能及时处理，请设置此值为最小值82，这样基本保持在每40ms上报一次数据。

参数	设置值	含义
采样率	16000 Hz	音频采样频率，决定每秒采集的样本数。16000 Hz适合语音编码，覆盖人声频率范围（300-3400 Hz），支持宽带语音（8 kHz带宽），适用于VoIP。
采样位深	16bit	每个音频样本的位数，决定动态范围和量化精度。16位是标准位深，提供足够的动态范围，适合语音和音乐编码。
声道数	1 (单声道)	音频通道数量。单声道仅使用一个通道，减少数据量和计算负载，适合语音通信场景。
应用模式	APPLICATION_AUDIO	高保真音频模式。
比特率	16000 bps	每秒传输的比特数，决定音频质量和文件大小。16000 bps（16 kbps）适合宽带语音，提供清晰语音质量，同时保持低带宽需求。
复杂度	4	编码算法的复杂程度（0-10）。值为4表示中等复杂度，平衡音质和计算资源消耗。
VBR	不启用	使用恒定比特率（CBR），简化编码。
帧长	40 ms	40ms帧，16000Hz采样下每帧640点，每帧输出字节数固定 80 字节。

参数	设置值	含义
带宽	宽带(WB, 8kHz)	编码音频的频率范围。宽带（8 kHz）覆盖人声完整频率，适合高质量语音编码，解码算力需求适中。

7.6.下行音频常见问题（模组->MCU）

7.6.1.流控方式介绍（推荐使用GPIO流控）

- GPIO流控：
 - PCM 格式播放时必须使用此方式
 - 根据GPIO电平状态判断是否发送数据
 - 高电平 时发送数据
 - 低电平 时停止发送
 - 为防止MCU接收内存溢出，每次发送包中间可能增加10ms的间隔。
- ACK流控：
 - 根据MCU返回的应答包判断是否发送下一包
 - 0 时代表MCU接收成功，模组可以继续发送后续数据
 - 1 时代表MCU接收失败，可以延时播放帧长（例如每帧40ms时可以延时40ms，也可以适当增加延时80ms，按实际情况来）发送下一帧数据。
 - 由于应答机制，这里可以进行多次重发确保不丢包。

7.6.2.帧格式版本说明（SPK配置0x03命令）

- 使用V1帧格式版本，模组每次只能推一帧的数据量给MCU。
- 使用V2帧格式版本，模组每次可根据需求推送多帧数据给MCU，主要为解决ACK方式播放卡顿问题增加，若播放无问题可忽略。

7.6.2.连续播放两条音频时，播放丢字问题。

- 原因：因为发给MCU的播放数据，在MCU是有缓存的，所以在连续播放两条音频，二次播放间隔较短时，发送的第二条音频的开始播放（0）指令将清空缓存导致播放丢字。
- 解决：通过MCU上报事件（0x12），等待播放结束（0x03）上报后再下发下一条音频。

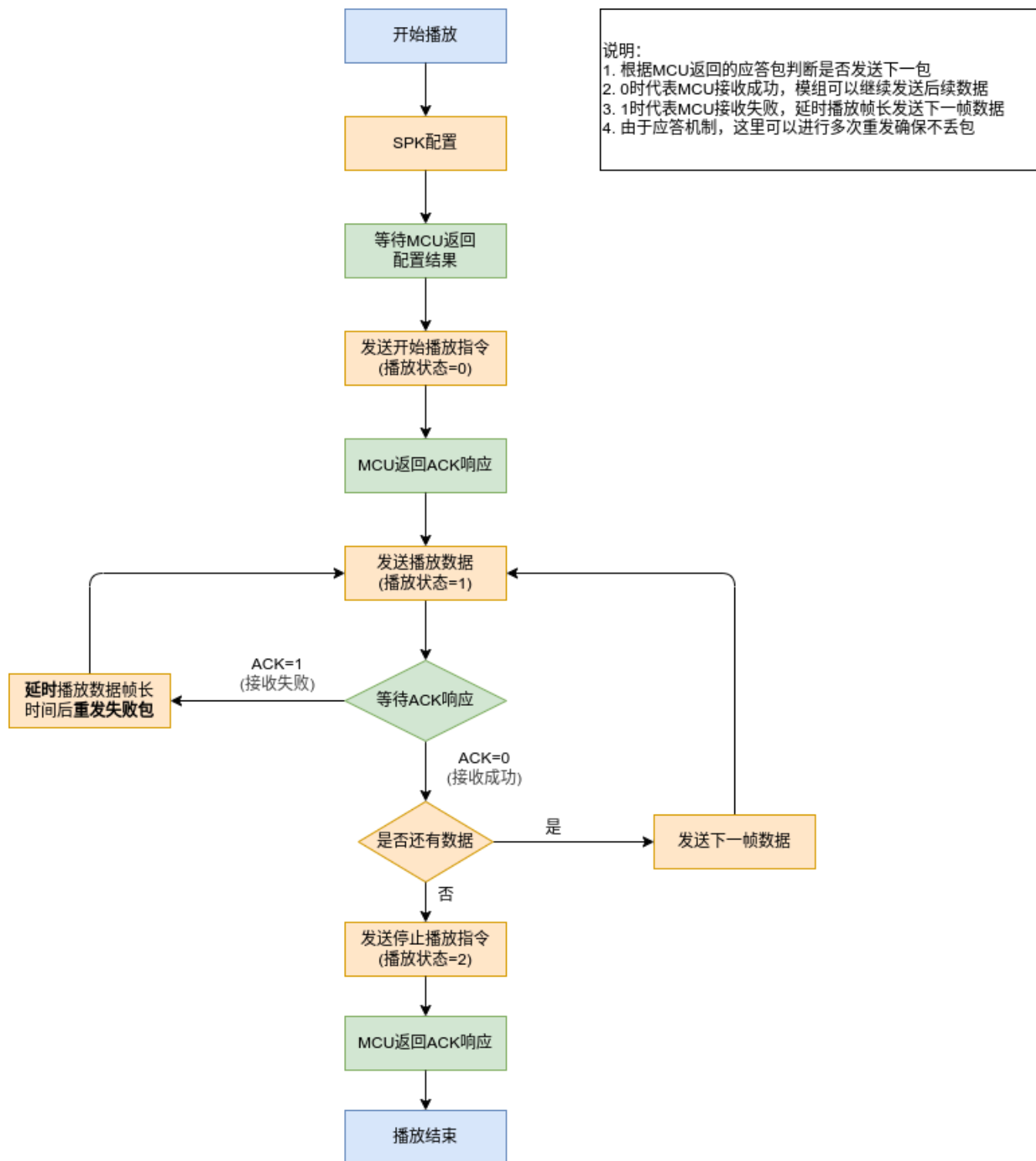
7.6.3.播放数据格式参数

- opus播放时不可以带头信息传输，使用采样率16K，码率16kbps，单通道。
- 具体参数如下。

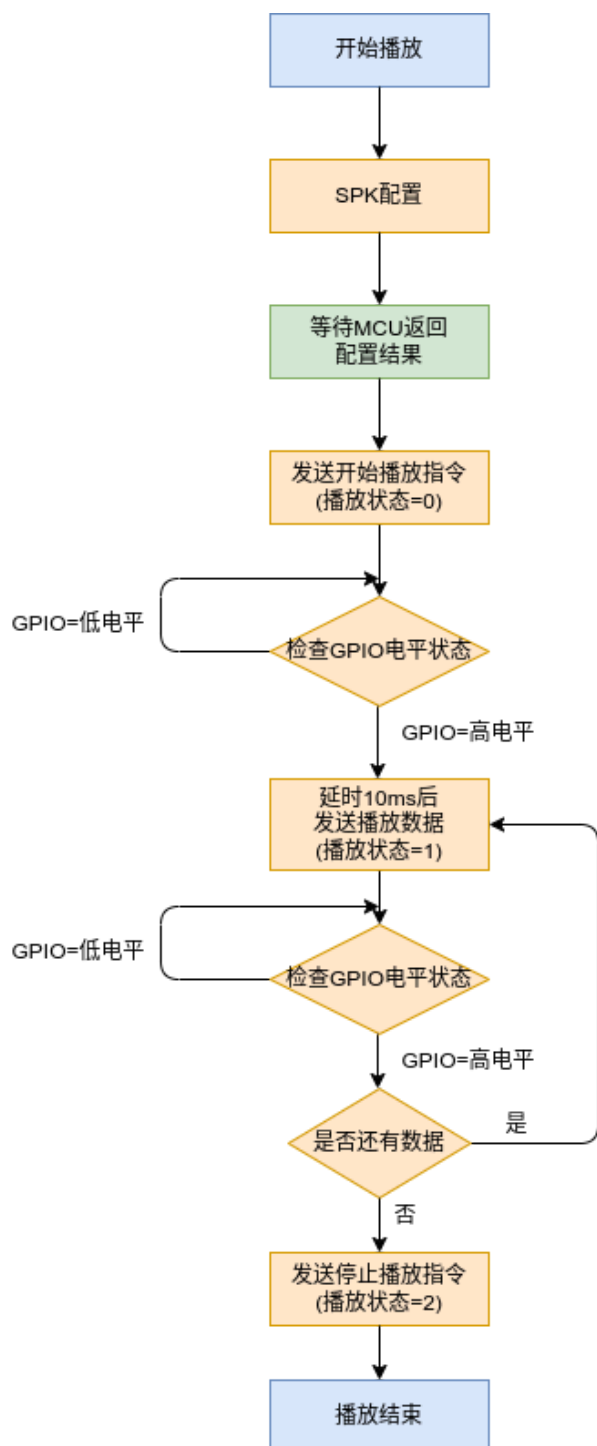
参数	设置值	含义
采样率	16000 Hz	音频采样频率，决定每秒采集的样本数。16000 Hz适合语音编码，覆盖人声频率范围（300-3400 Hz），支持宽带语音（8 kHz带宽），适用于VoIP。

参数	设置值	含义
采样位深	16bit	每个音频样本的位数，决定动态范围和量化精度。16位是标准位深，提供足够的动态范围，适合语音和音乐编码。
声道数	1 (单声道)	音频通道数量。单声道仅使用一个通道，减少数据量和计算负载，适合语音通信场景。
应用模式	APPLICATION_VOIP	编码器优化模式，专为语音通信（VoIP）设计，适合实时通话。
比特率	16000 bps	每秒传输的比特数，决定音频质量和文件大小。16000 bps（16 kbps）适合宽带语音，提供清晰语音质量，同时保持低带宽需求。
复杂度	4	编码算法的复杂程度（0-10）。值为4表示中等复杂度，平衡音质和计算资源消耗。
VBR	启用/不启用	可变比特率（VBR）允许根据音频内容动态调整比特率，以优化质量。启用VBR通常提供更一致的音质；禁用则使用恒定比特率（CBR），简化编码。
帧长	40/20/60 ms	每帧音频的持续时间，影响延迟和编码效率。
带宽	宽带(WB, 8kHz)	编码音频的频率范围。宽带（8 kHz）覆盖人声完整频率，适合高质量语音编码，解码算力需求适中。

7.6.4.ACK播放流程



7.6.4.GPIO播放流程



说明:

1. 根据GPIO电平状态判断是否发送数据
2. 高电平时发送数据
3. 低电平时停止发送
4. 高低电平仅代表模组是否发送数据, 无法区分包收取是否正确
5. 为防止MCU的接收内存溢出, 每个发送包之间要有10ms的间隔