

UM3506 I2C 通信手册

版本：V1.0



UNICMICRO

广芯微电子

广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

条款协议

本文档的所有部分，其著作权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

版本修订

版本	日期	描述
V1.0	2022.04.01	初始版本

目录

1	系统概述	1
1.1	功能框图	1
1.2	主要特性	1
2	通信协议	2
2.1	通信规则	2
2.2	通信协议定义	2
2.2.1	I2C 通信规则	2
2.2.2	寄存器总览	3
2.2.3	寄存器分述	5
2.3	固件升级协议	8
2.3.1	本地 FLASH 分区	9
2.3.2	固件升级流程	10
2.3.3	固件升级主、从设备寄存器操作规则	11
3	附录	14
3.1	16 位 CRC 算法	14

1 系统概述

UM3506 PD SoC 芯片定位于最新的 USB 接口解决方案，其符合 USB Power Delivery Rev 3.0 和 Type-C Rev 1.3 规范。它提供了一个灵活的可编程体系结构，可持续满足相关规范的不演进和 PD 应用之外的广泛功能扩展。

本文档旨在指导用户通过 I2C 方式访问 UM3506 系统，并定义通信协议的内容。

1.1 功能框图

图 1-1 是整个系统的结构框图

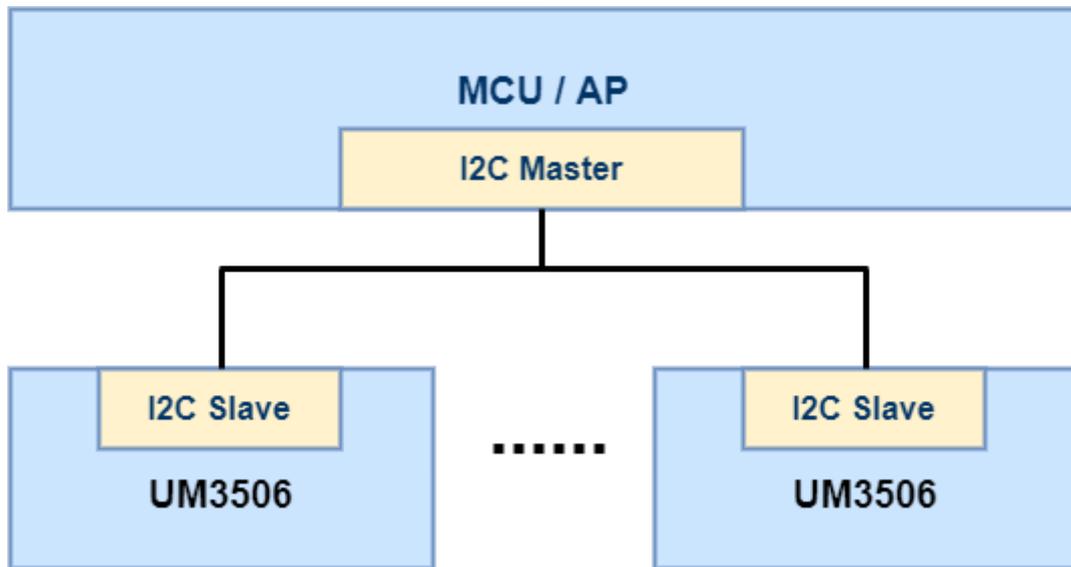


图 1-1 UM3506 通信系统框图

1.2 主要特性

- PD/Type-C 控制
- 系统策略控制
- 外设操作
- 升级操作
- 事件上报
 - 连接, OCP, OVP 等

2 通信协议

当前版本主从通信的数据通路采用 **I2C 通信** 实现。

2.1 通信规则

通信规则定义如下：

- 主设备发起，从设备响应主设备请求
- 主设备可以在任意时间发起数据传送请求。主设备的超时时间为 100ms。
- 如果从设备有事件产生，应通过中断引脚通知主设备

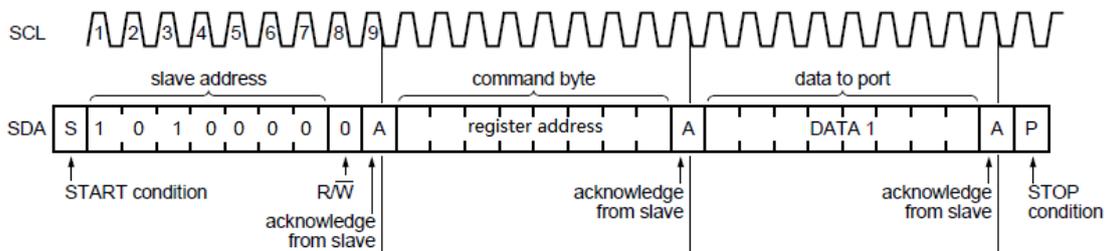
2.2 通信协议定义

通信双方交换的**通信消息**以 8 位寄存器操作为基础。读写数据的通信规则定义按照通常的寄存器读取方式。定义如下：

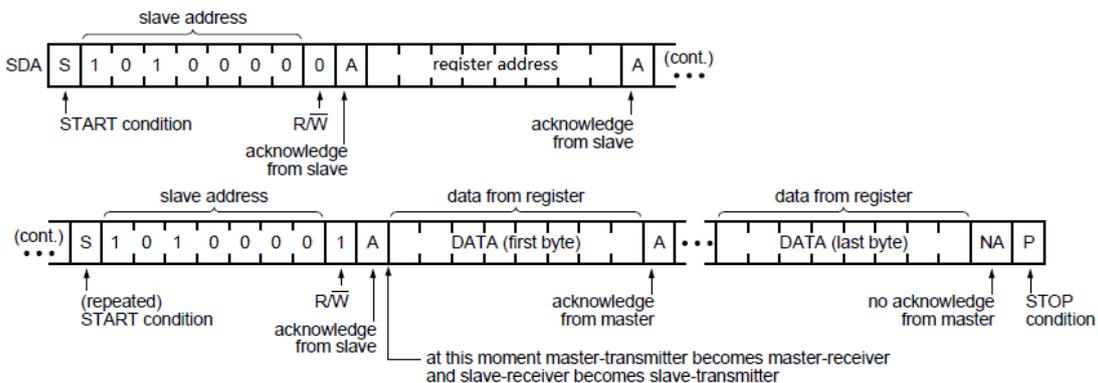
2.2.1 I2C 通信规则

与标准 I2C 操作一样，从设备地址为高 7 位有效，二进制定义为 **0b1010000 (0x50)**，第 0 位为读写控制位，可支持连续读写操作。详细描述如下。

2.2.1.1 写寄存器规则



2.2.1.2 读寄存器规则



2.2.2 寄存器总览

寄存器以及对应的数据定义如下：

地址	名称	操作	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
系统相关										
00h	软件版本	只读								
01h	系统控制 0	读/写	充电指示	温度异常保护	过流或短路保护	IDLE 模式指示	小电流模式指示	低电量保护	重启	关机
02h	功率配置 (SRC)	读/写	01h - FFh (Supply Power Profile)							
03h	功率配置 (SNK)	读/写	00h - FFh (Operating Power Profile)							
04h	电池容量	读/写	0 - 100							
05h	系统事件 0	读/写					从机请求系统控制 1	从机请求控制电源芯片	I2C 通信失败	操作失败
06h	系统事件 1	读/写							PD 状态改变	CC 状态改变
07h	系统控制 1	读/写	系统初始化完成	CC 连接 LED 使能	快充 LED 使能				擦除配置	保存配置
08h	当前功率	只读	0 - 100							
09h	温度 NTC 低字节	读/写	温度 NTC 低字节							
0Ah	温度 NTC 高字节	读/写	温度 NTC 高字节							
C 口控制										
10h	Type-C 控制	读/写	使能	Try. SRC 使能	VCONN 使能	检查 VBUS	数据角色：UFP/DFP/DRD		电源角色：SNK/SRC/DRP	
11h	PD 控制	读/写	使能	版本：R2/R3		VDM Modal	VDM SRC Startup	VDM	捕获模式	测试模式
12h	QC 控制	读/写	使能	版本：V2/V3/V4						
13h	高级控制 0	读/写	Hard reset	Soft reset	VCS 请求	DRS 请求	PRS 请求	错误恢复	CC2 无效	CC1 无效
14h	高级控制 1	读/写					New Power 请求	Goto min 请求	Snk Cap 请求	Src Cap 请求
15h	PDO 控制请求	读/写	RDO 获	PDO 获取	PDO 生	PDO 设	PDO 清	PDO 挡位：1-7		

			取		效	置	除			
16h	PDO 字节 0	读/写	PDO 字节 0							
17h	PDO 字节 1	读/写	PDO 字节 1							
18h	PDO 字节 2	读/写	PDO 字节 2							
19h	PDO 字节 3	读/写	PDO 字节 3							
C 口状态										
20h	Type-C 状态 0	只读			CC 选择	VCONN 供电	检测到 RA	UFP/DFP	SNK/SRC	连接状态
21h	PD 状态	只读				PDO 挡位: 1-7			电源协商完成	连接状态
22h	QC 状态	只读							版本: V2/V3	连接状态
23h	Type-C 状态 1	只读	前一次 CC 状态				当前 CC 状态			
24h	PE 状态	只读	状态值							
25h	C 口实时电流低字节	只读	C 口实时电流低字节							
26h	C 口实时电流高字节	只读	C 口实时电流高字节							
27h	C 口实时电压低字节	只读	C 口实时电压低字节							
28h	C 口实时电压高字节	只读	C 口实时电压高字节							
其他										
30h	电源芯片状态	只读	DM_L	AC_OK	INDET2	INDET1	VBUS_S HORT	OTP	EOC	Reserved
31h	电池电流低字节	只读	电池电流低字节							
32h	电池电流高字节	只读	电池电流高字节							
33h	电池电压低字节	只读	电池电压低字节							
34h	电池电压高字节	只读	电池电压高字节							
35h	电源芯片 CTRL0_SET	读/写	EN_OTG	Reserved	Reserved	VINREG _RATIO	FREQ_SET		DT_SET	
40h	芯片寄存器控制	读/写							读	写
41h	寄存器地址字节 0	读/写	芯片寄存器地址字节 0							

42h	寄存器地址字节 1	读/写	芯片寄存器地址字节 1					
43h	寄存器地址字节 2	读/写	芯片寄存器地址字节 2					
44h	寄存器地址字节 3	读/写	芯片寄存器地址字节 3					
45h	寄存器值字节 0	读/写	芯片寄存器值字节 0					
46h	寄存器值字节 1	读/写	芯片寄存器值字节 1					
47h	寄存器值字节 2	读/写	芯片寄存器值字节 2					
48h	寄存器值字节 3	读/写	芯片寄存器值字节 3					
49h	芯片 ADC 读取	只写	修改采样时间控制	采样时间控制 (0-7)			采样通道 (0-15)	
4Ah	芯片 ADC 值 0	只读	低 8 位值					
4Bh	芯片 ADC 值 1	只读	高 4 位值					
升级								
50h	升级命令	读/写	保留	保留	保留	保留	保留	升级命令
51h	升级状态	只读	完成	传输中	错误	Ready	APP 模式	当前模式
52h	固件 CRC 值 低字节	读/写	固件 CRC 值 低字节					
53h	固件 CRC 值 高字节	读/写	固件 CRC 值 高字节					
54h	24 位固件长度 字节 0	读/写	24 位固件长度 字节 0					
55h	24 位固件长度 字节 1	读/写	24 位固件长度 字节 1					
56h	24 位固件长度 字节 2	读/写	24 位固件长度 字节 2					
60h-7Fh	升级数据 (32 字节)	读/写	缓冲区 [0-31]					
80h-FFh	Vendor Defined Registers		As many as Vendor Defines					

2.2.3 寄存器分述

寄存器 00h: 软件版本

位	操作类型	位名称	默认值	描述
---	------	-----	-----	----

7-0	只读	软件版本	0	默认为0
-----	----	------	---	------

寄存器 01h: 系统控制 0

位	操作类型	位名称	默认值	描述
7	读/写	充放电指示	0	充电指示: 0: 放电, 1: 充电 (边充边放按照充电模式)
6		温度异常保护	0	温度异常保护: 0: 正常工作, 1: 保护
5		过流或短路保护	0	过流或短路保护: 0: 正常工作, 1: 保护
4		IDLE 模式指示	0	IDLE 模式指示: 0: 正常工作, 1: IDLE 标记
3		小电流模式指示	0	小电流模式指示: 0: 正常工作, 1: 小电流模式
2		低电量保护	0	低电量保护: 0: 正常工作, 1: 低电量
1		重启	0	系统重启: 0: 正常工作, 1: 重启
0		关机	0	系统关机: 0: 正常工作, 1: 关机

寄存器 02h: 功率配置 (Source)

位	操作类型	位名称	默认值	描述
7-0	读/写	功率配置索引	0	默认为0

Source 功率配置索引对应关系:

- 0: 无效的,
- 1: 自定义,
- 2: 0.5W_FIXED, 5V100mA,
- 3: 15W_FIXED, 5V3A,
- 4: 18W_FIXED, 5V3A, 9V2A, 12V1.5A, 15V1.2A,
- 5: 18W_PPS, 5V3A, 9V2A; 3.3-5.9V3A, 3.3-11V2A,
- 6: 27W_FIXED, 5V, 9V3A; 12V2.25A; 15V1.8A; 20V1.35A,
- 7: 27W_PPS, 5V, 9V3A; 15V1.8A; 20V1.35A; 3.3-11VA, 3.3-16V1.8A,
- 8: 30W_FIXED, 5V, 9V3A; 12V2.5A; 15V2A,
- 9: 30W_PPS, 5V, 9V3A; 15V2A; 3.3-11V3A, 3.3-16V2A,
- 10: 30W_FIXED_20V 5V, 9V3A; 12V2.5A; 20V1.5A,
- 11: 30W_PPS_20V, 5V, 9V3A; 15V2A; 3.3-11V3A, 3.3-20V1.5A,
- 12: 45W_FIXED, 5V, 9V, 15V3A; 20V2.25A,

13: 45W_PPS,	5V, 9V, 15V3A;20V2.25A;3.3-16V3A,
14: 60W_FIXED,	5V, 9V, 15V3A;20V3A,
15: 60W_PPS,	5V, 9V, 15V3A;20V3A;3.3-21V3A,
16: 65W_FIXED,	5V, 9V, 15V3A;20V3.25A,
17: 65W_PPS,	5V, 9V, 15V3A;20V3.25A;3.3-21V3.25A,
18: 72W_FIXED,	5V, 9V, 15V3A;20V3.6A,
19: 72W_PPS,	5V, 9V, 15V3A;20V3.6A;3.3-21V3.6A,
20: 75W_FIXED,	5V, 9V, 15V3A;20V3.75A,
21: 75W_PPS,	5V, 9V, 15V3A;20V3.75A;3.3-21V3.75A,
22: 80W_FIXED,	5V, 9V, 15V3A;20V4A,
23: 80W_PPS,	5V, 9V, 15V3A;20V4A;3.3-21V4A,
24: 87W_FIXED,	5V, 9V, 15V3A;20V4.35A,
25: 87W_PPS,	5V, 9V, 15V3A;20V4.35A;3.3-21V4.35A,
26: 90W_FIXED,	5V, 9V, 15V3A;20V4.5A,
27: 90W_PPS,	5V, 9V, 15V3A;20V4.5A;3.3-21V4.5A,
28: 96W_FIXED,	5V, 9V, 15V3A;20V4.8A,
29: 96W_PPS,	5V, 9V, 15V3A;20V4.8A;3.3-21V4.8A,
30: 100W_FIXED,	5V, 9V, 15V3A;20V5A,
31: 100W_PPS,	5V, 9V, 15V3A;20V5A;3.3-21V5A,

寄存器 03h: 功率配置 (Sink)

位	操作类型	位名称	默认值	描述
7-0	读/写	功率配置索引	0	默认为 0

Sink 功率配置索引对应关系:

- 0: 5V3A 15W
- 1: 跟随 Source 的最大功率
- 2: 9V2A 18W
- 3: 12V2A 24W
- 4: 14.8V2A 29.6W
- 5: 15V2A 30W
- 6: 20V2A 40W
- 7: 20V2.5A 50W
- 8: 20V3A 60W
- 9: 5V0.1A 0.5W
- 10: 20V5A 100W

11: PPS 自定义

12: 自定义

寄存器 04h: 电池容量

位	操作类型	位名称	默认值	描述
7-0	读/写	电池容量	0	电池容量百分比, 取值范围: 0 - 100

寄存器 50h: 固件升级命令

位	操作类型	位名称	默认值	描述
7-3	只读	保留	0	保留
2-0	读/写	固件升级命令	0	0: 空闲 1: 升级开始 2: 升级开始文件信息 3: 升级停止或中断

寄存器 51h: 固件升级状态

位	操作类型	位名称	默认值	描述
7	只读	完成	0	0: 固件传输未结束 1: 固件传输结束
6		传输中	0	0: 没有固件传输 1: 固件传输中
5		错误	0	0: 无错误 1: 出错
4		Ready	0	0: 不可以发送文件数据 1: 可以发送文件数据
3		APP 模式	0	0: Boot 模式 1: APP 模式
2-0		当前状态	0	当前操作, 定义同固件升级命令, 但只读, 作为状态查询

2.3 固件升级协议

固件升级功能, 是 UM3506 与远程主机基于 I2C 通信, UM3506 接收远程主机发送的固件数据在本地保存并正确引导启动。

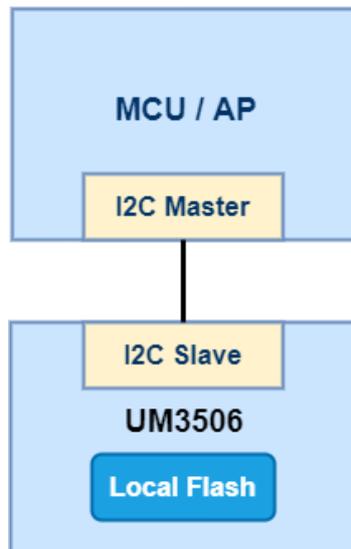
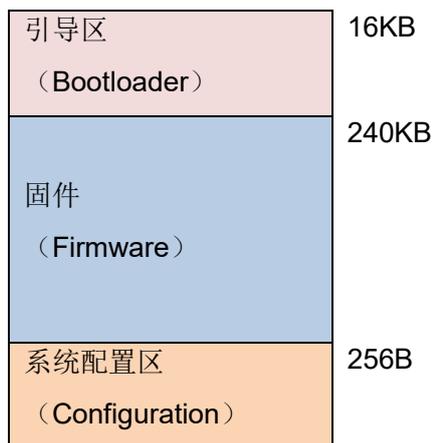


图 2-1 固件升级示意图

2.3.1 本地 FLASH 分区



2.3.2 固件升级流程

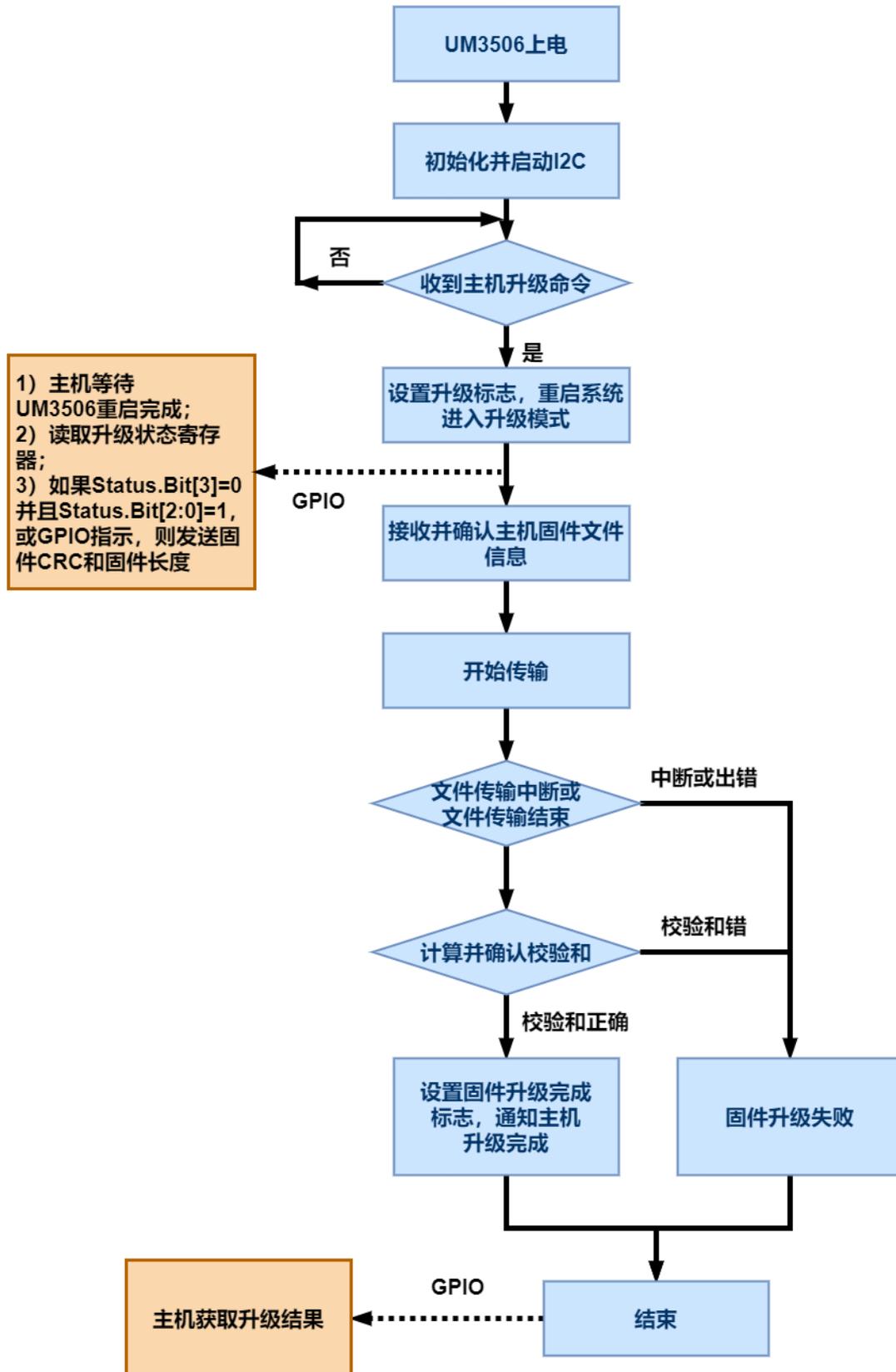


图 2-2 固件升级流程图

说明:

1. I2C 数据传输中断可以是 I2C 通信出错或者是传输超时。超时暂定为 1000 毫秒。
2. 数据传输过程中，主设备可以进行 I2C 读操作进行从设备状态查询。
3. 检验和使用 16 位 CRC 算法，详见附件 C 文件。
4. 本流程由寄存器定义的协议来实现。

2.3.3 固件升级主、从设备寄存器操作规则



图 2-3 APP 模式升级流程图

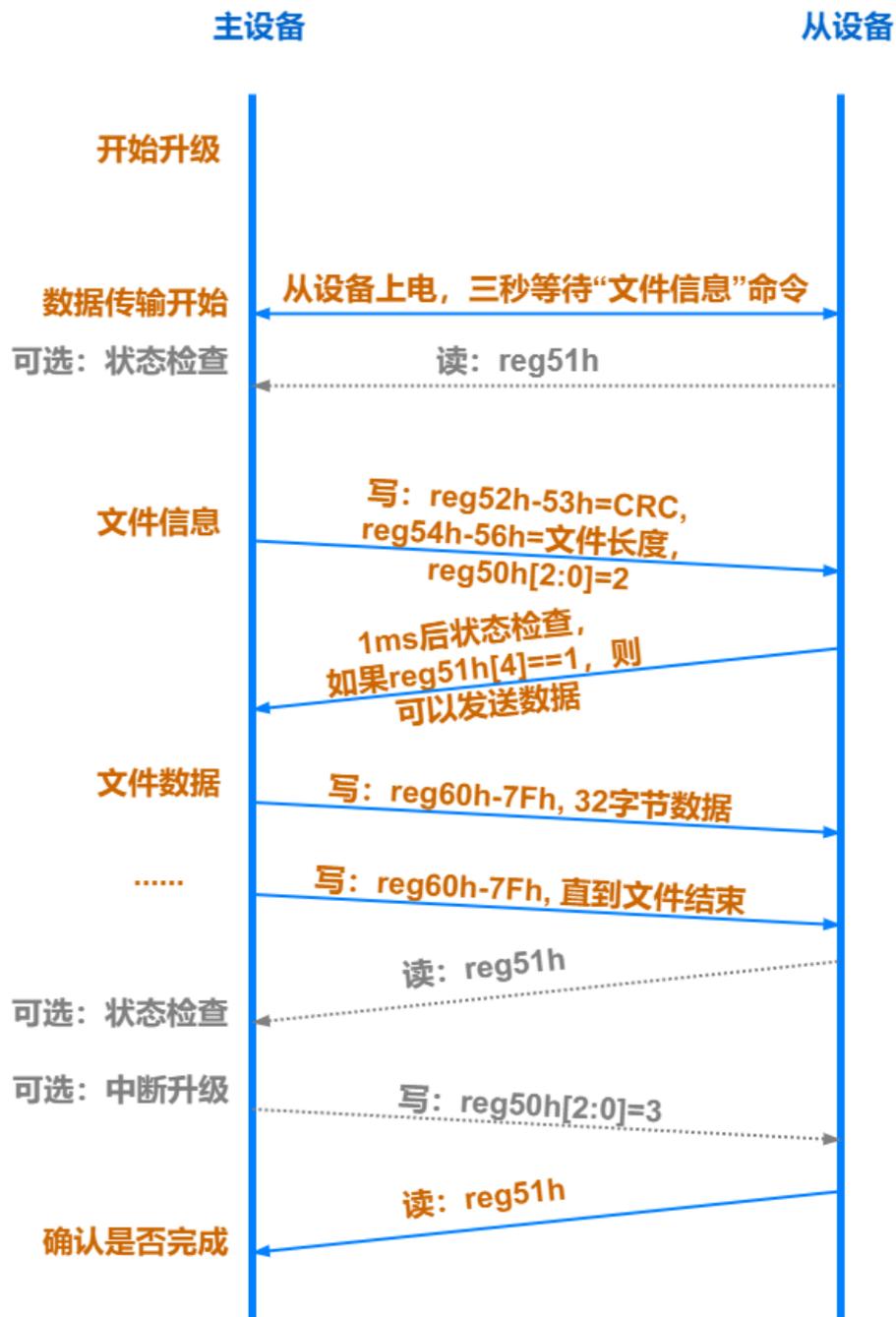


图 2-4 Boot 模式升级流程图

2.3.3.1 升级模式说明

- 1) App 模式：指系统正常运行时要进行的升级。

Boot 模式：指系统刚上电，还没有进入正常运行模式时，主设备有三秒时间可以通过发送文件信息命令开始升级。这种情况可以用于 app 模式下升级失败时的一种特殊补救模式，不是通常使用的模式。

- 2) 状态寄存器 51h.Bit[3]用于表明系统的模式，主设备可以在升级前读取再确认升级模式。

2.3.3.2 APP 模式固件升级开始规则

- 1) 主设备 I2C 写寄存器地址 50h, Bit[2:0]为 1。
- 2) 从设备收到升级命令后, 设置状态 Bit[2:0]为 1, 重新启动进入 Boot 升级模式; 主设备在发送升级命令后等待 1 秒, 检查状态 Bit[2:0]为 1 并且状态 Bit[3]为 0, 则表示从设备进入 Boot 升级模式。
- 3) 可选: 通过从设备的 GPIO 输出 1 来指示进入升级模式。
- 4) 可选: 主设备读取状态寄存器 51h 以确认是否有错

2.3.3.3 固件文件信息传输规则

- 1) 主设备 I2C 写寄存器地址 52h, 连续写 5 个字节, 其中:
第一、二个字节: 寄存器 52h、53h 为 16 位 CRC 值, 52h 为低地址, 53h 为高地址。
第三、四、五个字节: 寄存器 54h、55h、56h 为 24 位文件长度 (54h 为 24 位文件长度字节 0, 55h 为 24 位文件长度字节 1, 56h 为 24 位文件长度字节 2), 文件长度以字节为单位。
- 2) 主设备 I2C 写寄存器地址 50h, Bit[2:0]为 2, 触发升级开始文件信息。
- 3) 可选: 主设备读取状态寄存器 51h 以确认是否有错。

2.3.3.4 固件文件数据传输规则

在发送完文件信息之后, 就直接发送文件数据, 每次发送 32 字节。

- 1) 主设备读取状态寄存器 51h 以确认是否可以传送数据。
- 2) 数据传输开始, 主设备 I2C 写寄存器地址 60h, 连续写 32 个字节。重复此操作, 直到文件结束。
- 3) 如果主设备需要中断数据传输可以写寄存器地址 50h, Bit[2:0]为 3。
- 4) 最后, 主设备读取状态寄存器 51h 以确认文件是否有错或者完成。
- 5) 可选: 从设备置 GPIO 为低, 通知传输结束。

3 附录

3.1 16 位 CRC 算法

```

/**
 * \file    crc16.c
 * \brief   CRC-16:  $x^{16} + x^{15} + x^2 + 1$ 
 *
 * \par     Copyright (c) 2017 - 2022, Unicmicro Co., Ltd. All Rights Reserved.
 * \author
 * \date    2022-03-05
 * \version
 *
 * \history
 *   Date      Note
 *   2022-03-05  Original Creation
 *
 */

#include "crc16.h"

static const unsigned short crc16tab[256]= {
    0x0000U,0x1021U,0x2042U,0x3063U,0x4084U,0x50a5U,0x60c6U,0x70e7U,
    0x8108U,0x9129U,0xa14aU,0xb16bU,0xc18cU,0xd1adU,0xe1ceU,0xf1efU,
    0x1231U,0x0210U,0x3273U,0x2252U,0x52b5U,0x4294U,0x72f7U,0x62d6U,
    0x9339U,0x8318U,0xb37bU,0xa35aU,0xd3bdU,0xc39cU,0xf3ffU,0xe3deU,
    0x2462U,0x3443U,0x0420U,0x1401U,0x64e6U,0x74c7U,0x44a4U,0x5485U,
    0xa56aU,0xb54bU,0x8528U,0x9509U,0xe5eeU,0xf5cfU,0xc5acU,0xd58dU,
    0x3653U,0x2672U,0x1611U,0x0630U,0x76d7U,0x66f6U,0x5695U,0x46b4U,
    0xb75bU,0xa77aU,0x9719U,0x8738U,0xf7dfU,0xe7feU,0xd79dU,0xc7bcU,
    0x48c4U,0x58e5U,0x6886U,0x78a7U,0x0840U,0x1861U,0x2802U,0x3823U,
    0xc9ccU,0xd9edU,0xe98eU,0xf9afU,0x8948U,0x9969U,0xa90aU,0xb92bU,
    0x5af5U,0x4ad4U,0x7ab7U,0x6a96U,0x1a71U,0x0a50U,0x3a33U,0x2a12U,
    0xdbfdU,0xcbbdcU,0xfbbfU,0xeb9eU,0x9b79U,0x8b58U,0xbb3bU,0xab1aU,
    0x6ca6U,0x7c87U,0x4ce4U,0x5cc5U,0x2c22U,0x3c03U,0x0c60U,0x1c41U,
    0xedaeU,0xfd8fU,0xcdccU,0xddcdU,0xad2aU,0xbd0bU,0x8d68U,0x9d49U,
    0x7e97U,0x6eb6U,0x5ed5U,0x4ef4U,0x3e13U,0x2e32U,0x1e51U,0x0e70U,

```

```

0xff9fU,0xefbeU,0xdfddU,0xcffcU,0xbf1bU,0xaf3aU,0x9f59U,0x8f78U,
0x9188U,0x81a9U,0xb1caU,0xa1ebU,0xd10cU,0xc12dU,0xf14eU,0xe16fU,
0x1080U,0x00a1U,0x30c2U,0x20e3U,0x5004U,0x4025U,0x7046U,0x6067U,
0x83b9U,0x9398U,0xa3fbU,0xb3daU,0xc33dU,0xd31cU,0xe37fU,0xf35eU,
0x02b1U,0x1290U,0x22f3U,0x32d2U,0x4235U,0x5214U,0x6277U,0x7256U,
0xb5eaU,0xa5cbU,0x95a8U,0x8589U,0xf56eU,0xe54fU,0xd52cU,0xc50dU,
0x34e2U,0x24c3U,0x14a0U,0x0481U,0x7466U,0x6447U,0x5424U,0x4405U,
0xa7dbU,0xb7faU,0x8799U,0x97b8U,0xe75fU,0xf77eU,0xc71dU,0xd73cU,
0x26d3U,0x36f2U,0x0691U,0x16b0U,0x6657U,0x7676U,0x4615U,0x5634U,
0xd94cU,0xc96dU,0xf90eU,0xe92fU,0x99c8U,0x89e9U,0xb98aU,0xa9abU,
0x5844U,0x4865U,0x7806U,0x6827U,0x18c0U,0x08e1U,0x3882U,0x28a3U,
0xcb7dU,0xdb5cU,0xeb3fU,0xfb1eU,0x8bf9U,0x9bd8U,0xabbbU,0xbb9aU,
0x4a75U,0x5a54U,0x6a37U,0x7a16U,0x0af1U,0x1ad0U,0x2ab3U,0x3a92U,
0xfd2eU,0xed0fU,0xdd6cU,0xcd4dU,0xbdaaU,0xad8bU,0x9de8U,0x8dc9U,
0x7c26U,0x6c07U,0x5c64U,0x4c45U,0x3ca2U,0x2c83U,0x1ce0U,0x0cc1U,
0xef1fU,0xff3eU,0xcf5dU,0xdf7cU,0xaf9bU,0xbfbaU,0x8fd9U,0x9ff8U,
0x6e17U,0x7e36U,0x4e55U,0x5e74U,0x2e93U,0x3eb2U,0x0ed1U,0x1ef0U
};

```

```

unsigned short crc16_ccitt(const char *buf, int len)
{
    register int counter;
    register unsigned short crc = 0;
    for( counter = 0; counter < len; counter++)
        crc = (crc<<8) ^ crc16tab[((crc>>8) ^ *(char *)buf++)&0x00FF];
    return crc;
}

```