



UCChip UHF READER

串口协议

适用范围

UCM601/2/6/8 系列模组

文档密级

| | |
|------------------------|--------|
| 文档名称 | 密级 |
| UCChip UHF READER 串口协议 | 公开 |
| 文档版本 | 共 55 页 |
| V1.1.3 | |

版本历史

| 版本 | 日期 | 变更描述 |
|--------|------------|---|
| V1.0.0 | 2022.09.08 | 初始版本 |
| V1.0.1 | 2022.10.26 | 修改获取固件格式，增加型号识别 修改天线序列号 最小从 0x01 开始 修改频率设置格式 增加 1 个字节作为频率间隔 修改射频链路选项 修改标签回复命令 RSSI 和 Freq 的长度 |
| V1.0.2 | 2022.12.03 | 修复读写协议回复格式错误 |
| V1.0.3 | 2022.12.08 | 修复文档中没有停止盘存命令 |
| V1.0.4 | 2022.12.14 | 修改读协议发送长度错误 |
| V1.0.5 | 2022.12.22 | 添加 SM7 相关命令 |
| V1.0.6 | 2023.02.17 | 添加自定义盘存命令,更改 RSSI 计算公式 |
| V1.0.7 | 2023.03.16 | 修改盘存命令上报格式，删除标签筛选命令 |
| V1.0.8 | 2023.04.04 | 单个指令解析中添加了错误码说明 |
| V1.0.9 | 2023.04.12 | 新增加天线切换盘存命令 |
| V1.1.0 | 2023.05.05 | 新增国标命令，新增复旦微国标命令，修改部分命令的错误回复 |
| V1.1.1 | 2023.06.26 | 新增 4、8 通道读写器输出功率设置命令；优化多天线轮询盘存描述 |
| V1.1.2 | 2023.07.17 | 修改数据包定义描述和恢复默认参数指令描述 更新指令集列表 新增天线检测指令、超温告警指令和保存参数指令 修改设置工作天线指令数据包描述 新增第 8 节出厂默认参数配置表 |
| V1.1.3 | 2023.11.30 | 文档整体重新排版 增加部分命令参考示例 修改 8 天线读取当前输出功率的长度 |

| | | |
|--------|-----------|---|
| | | 修改设置单天线读写器射频输出功率备注信息 修改数据包格式定义 checksum 备注信息 修改查询盘存缓存返回 cmd 命令 新添加 CW 相关命令 新添加 Session 和 Target 命令 新添加盘存心跳包和上报相关命令 新添加 TX ON 和 TX OFF 相关命令 新添加选择标签相关命令 |
| V1.1.4 | 2024.2.29 | 取消了天线检测相关命令 |
| V1.1.5 | 2024.3.11 | 增加设置读取工作天线-多天线 修改 80H 盘存标签 Data 字段备注 修改 89H 盘存标签（实时）Data 字段备注 修改 8AH 盘存标签（自定义）Data 字段备注 增加 8AH 盘存标签（自定义）匹配参数相关备注命令说明 |
| V1.1.6 | 2024.5.15 | 新添加 GPIO 设置和读取命令 |
| V1.1.7 | 2024.6.24 | 新添加读取 UUID 命令 |
| V1.1.8 | 2024.7.12 | 移除了使用 8A 命令设置不匹配时标签的 CRC 校验返回 |
| V1.1.9 | 2024.9.2 | 修改 checksum 描述，加入 LRC 校验描述 |
| V1.2.0 | 2025.2.14 | 修改设置读写器工作频率范围的描述。 |
| V1.2.1 | 2025.3.3 | 勘误 |
| V1.2.1 | 2025.5.10 | 删除设置临时功率范围错误的描述 |

前言

本文档是御芯微超高频 RFID 读写器的串口通信协议定义书，主要提供给使用御芯微超高频 RFID 读写器系列产品的开发人员。协议规定了读写器与上位机或其他控制平台的通信格式，开发人员在保留读写器原有出厂固件或使用了读写器二次开发版本但保留了原有串口协议的情况下可依照本文档进行开发。

目录

| | |
|--------------------------------------|----|
| 1. 文档概述 | 3 |
| 1.1. 名称 | 3 |
| 1.2. 功能 | 3 |
| 1.3. 参考文档 | 3 |
| 1.4. 使用范围 | 3 |
| 2. 数据包格式定义 | 3 |
| 2.1. 上位机发送数据包格式定义 | 3 |
| 2.2. 读写器返回数据包格式定义 | 4 |
| 2.3. 指令集定义 | 5 |
| 3. 指令解析 | 7 |
| 3.1. 3EH——设置 CW 开关状态 | 7 |
| 3.2. 3FH——获取 CW 开关状态 | 7 |
| 3.3. 42H——复旦微 GB 加密通信写 | 8 |
| 3.4. 43H——复旦微 GB 加密通信读 | 10 |
| 3.5. 45H——读标签 GB | 11 |
| 3.6. 46H——写标签 GB | 13 |
| 3.7. 47H——锁定标签 GB | 15 |
| 3.8. 49H——灭活标签 GB | 16 |
| 3.9. 4AH——保存参数 | 17 |
| 3.10. 4BH——恢复读写器默认配置 | 18 |
| 3.11. 4CH——读写器应用软件升级 | 18 |
| 3.12. 4DH——基带固件升级 | 19 |
| 3.13. 4EH——设置心跳开关和周期 | 20 |
| 3.14. 4FH——获取心跳开关和心跳周期 | 20 |
| 3.15. 5AH——获取 Session 和 Target | 21 |
| 3.16. 5BH——设置 Session 和 Target | 22 |
| 3.17. 5CH——获取 TX RF 占空比 | 23 |
| 3.18. 5DH——设置 TX RF 时间 | 23 |
| 3.19. 5FH——设置 4 天线读写器射频输出功率 | 24 |
| 3.20. 60H——查询读写器 GPIO 状态 | 25 |
| 3.21. 61H——设置读写器 GPIO 状态 | 26 |
| 3.22. 66H——设置读写器临时射频输出功率 | 27 |
| 3.23. 69H——设置射频链路的通讯速率 | 28 |
| 3.24. 6AH——查询射频链路的通讯速率 | 29 |
| 3.25. 70H——复位读写器 | 30 |
| 3.26. 71H——设置串口通讯波特率 | 30 |
| 3.27. 72H——查询读写器固件版本 | 31 |
| 3.28. 73H——设置读写器地址 | 32 |
| 3.29. 74H——设置读写器工作天线 | 33 |
| 3.30. 75H——查询当前工作天线 | 34 |
| 3.31. 76H——设置单天线读写器射频输出功率 | 35 |
| 3.32. 77H——查询读写器当前输出功率 | 36 |
| 3.33. 78H——设置读写器工作频率范围 | 37 |

| | | |
|-------|-----------------------------------|----|
| 3.34. | 79H—查询读写器工作频率范围 | 39 |
| 3.35. | 7AH—设置蜂鸣器开关 | 40 |
| 3.36. | 7BH—查询当前设备的工作温度 | 41 |
| 3.37. | 80H—盘存标签 | 42 |
| 3.38. | 81H—读标签 6C | 43 |
| 3.39. | 82H—写标签 6C | 45 |
| 3.40. | 83H—锁定标签 6C | 47 |
| 3.41. | 84H—灭活标签 6C | 49 |
| 3.42. | 85H—匹配 ACCESS 操作的 EPC 号 | 50 |
| 3.43. | 86H—查询匹配的 EPC 状态 | 51 |
| 3.44. | 87H—多天线轮询盘存 | 52 |
| 3.45. | 89H—盘存标签(实时上传标签数据) | 55 |
| 3.46. | 8AH—盘存标签(自定义读取数据) | 56 |
| 3.47. | 8BH—自定义 session 和 target 盘存 | 59 |
| 3.48. | 8CH—停止盘存 | 61 |
| 3.49. | 8DH—选择标签 | 61 |
| 3.50. | 8EH—获取标签 | 62 |
| 3.51. | 90H—查询盘存缓存 | 64 |
| 3.52. | 91H—查询并重置盘存缓存 | 65 |
| 3.53. | 92H—查询盘存缓存标签数量 | 65 |
| 3.54. | 93H—重置盘存缓存 | 66 |
| 3.55. | 95H—SM7 加密写(可选项) | 67 |
| 3.56. | 96H—SM7 解密读(可选项) | 68 |
| 3.57. | 97H—SM7 PK 秘钥更新(可选项) | 69 |
| 3.58. | 98H—复旦微 GB 双向认证 | 70 |
| 3.59. | E1H—读写器温度过高告警 | 72 |
| 3.60. | F3H—读取 UUID | 72 |
| 4. | 错误码表 | 73 |
| 5. | 频率参数对应表 | 75 |
| 6. | RSSI 参数计算方法(C 语言描述) | 76 |
| 7. | 校验和计算方法(C 语言描述) | 77 |
| 8. | 出厂默认参数 | 78 |

1. 文档概述

1.1. 名称

UCChip UHF READER 串口协议。

1.2. 功能

本协议规定了御芯微超高频 RFID 读写器模组与上位机的通信格式。

1.3. 参考文档

无。

1.4. 使用范围

UCM60x 系列模组。

2. 数据包格式定义

2.1. 上位机发送数据包格式定义



图 1 上位机发送数据包格式

| 序号 | 字段名称 | 字节数 (Byte) | 说明 |
|----|---------|---------------|---|
| 1 | Head | 1 | 数据包头，每包数据均以 0xA0 开始 |
| 2 | Len | 1 | 数据包长，不包含包头及包长本身，若完整数据包长为 8 字节，则 Len 的值为 6 |
| 3 | Address | 1 | 读写器地址。供 RS-485 接口串联时使用。一般地址从 1~255(0xFF)，0 (0x00) 为公用地址。读写器接收自身地址和公用地址的命令 |

| | | | |
|---|----------|-----|--|
| 4 | Cmd | 1 | 命令码, RFID_COMMAND 中定义的枚举值 |
| 5 | Data | 0~N | 命令参数, 可能为 0 字节或多个字节 |
| 6 | Checksum | 1 | <p>LRC 校验, 除校验字段本身外所有字节的校验和与 0x100 求模, 再用 0x100 减去模 (或者校验和取反加 1)</p> <p>C 例程</p> <pre>static rt_uint8_t checksum(rt_uint8_t* buff, rt_uint32_t data_len) { rt_uint32_t index = 0; rt_uint16_t sum=0; for (index = 0; index < data_len; index++) { sum=sum+buff[index]; } return ((0x100-(sum%0x100))&0xff); }</pre> |

2.2. 读写器返回数据包格式定义



图 2 读写器返回数据包格式

| 序号 | 字段名称 | 字节数 (Byte) | 说明 |
|----|----------|------------|--|
| 1 | Head | 1 | 数据包头, 每包数据均以 0xA0 开始 |
| 2 | Len | 1 | 数据包长, 不包含包头及包长本身, 若完整数据包长为 8 字节, 则 Len 的值为 6 |
| 3 | Address | 1 | 读写器自身地址 |
| 4 | Cmd | 1 | 命令码, RFID_COMMAND 中定义的枚举值 |
| 5 | Data | 0~N | 读写器返回的数据, 可能为 0 字节或多个字节 |
| 6 | Checksum | 1 | <p>LRC 校验, 除校验字段本身外所有字节的校验和与 0x100 求模, 再用 0x100 减去模 (或者校验和取反加 1)</p> |

| | | | |
|--|--|--|---|
| | | | <p>C 例程</p> <pre>static rt_uint8_t checksum(rt_uint8_t* buff, rt_uint32_t data_len) { rt_uint32_t index = 0; rt_uint16_t sum=0; for (index = 0; index < data_len; index++) { sum=sum+buff[index]; } return ((0x100-(sum%0x100))&0xff); }</pre> |
|--|--|--|---|

备注：在未特殊说明的情况下，读写器返回数据包中的 Data 段为错误码，具体错误码请查阅第 4 章节 -- 错误码表。

2.3. 指令集定义

| 序号 | 命令码 | 名称 | 描述 |
|----|------|-------------------------------|----------------------------------|
| 1 | 0x3E | cmd_set_cw | 打开 cw 波 |
| 2 | 0x3F | cmd_get_cw | 获取 cw 波状态 |
| 3 | 0x42 | cmd_gb_seu_write | 复旦微 GB 加密通信写 |
| 4 | 0x43 | cmd_gb_seu_read | 复旦微 GB 加密通信读 |
| 5 | 0x45 | cmd_gb_read | 读标签 GB |
| 6 | 0x46 | cmd_gb_write | 写标签 GB |
| 7 | 0x47 | cmd_gb_lock | 锁定标签 GB |
| 8 | 0x49 | cmd_gb_kill | 灭活标签 GB |
| 9 | 0x4a | cmd_reader_para_save | 保存参数 |
| 10 | 0x4b | cmd_reader_para_reset | 恢复读写器默认配置 |
| 11 | 0x4c | cmd_reader_app_upgrade | 读写器应用软件升级 |
| 12 | 0x4d | cmd_baseband_firmware_upgrade | 基带固件升级 |
| 13 | 0x4E | cmd_set_keepalive | 设置心跳包和上报时间 |
| 14 | 0x4F | cmd_get_keepalive | 获取心跳包和上报时间 |
| 15 | 0x5A | cmd_get_session_target | 获取 session 和 target |
| 16 | 0x5B | cmd_set_session_target | 设置 session 和 target |
| 17 | 0x5C | cmd_get_tx_time | 获取读写器 tx_on 时间和 tx_off 时间 |
| 18 | 0x5D | cmd_set_tx_time | 设置读写器 tx_on 时间和 tx_off 时间,调整功耗使用 |
| 19 | 0x5E | cmd_set_8_ant_power | 设置的 8 天线读写器射频输出功率 |
| 20 | 0x5F | cmd_set_4_ant_power | 设置的 4 天线读写器射频输出功率 |

| | | | |
|----|------|---|-------------------------|
| 21 | 0x60 | cmd_read_gpio_value | 查询 GPIO 状态 |
| 22 | 0x61 | cmd_set_gpio_value | 设置 GPIO 状态 |
| 23 | 0x66 | cmd_set_temporary_output_power | 设置读写器临时射频输出功率 |
| 24 | 0x69 | cmd_set_rf_link_profile | 设置射频链路的通讯速率 |
| 25 | 0x6A | cmd_get_rf_link_profile | 查询射频链路的通讯速率 |
| 26 | 0x70 | cmd_reset | 复位读写器 |
| 27 | 0x71 | cmd_set_uart_baudrate | 设置串口通讯波特率 |
| 28 | 0x72 | cmd_get_firmware_version | 查询读写器固件版本 |
| 29 | 0x73 | cmd_set_reader_address | 设置读写器地址 |
| 30 | 0x74 | cmd_set_work_antenna | 设置读写器工作天线 |
| 31 | 0x75 | cmd_get_work_antenna | 查询当前工作天线 |
| 32 | 0x76 | cmd_set_output_power | 设置的单天线读写器射频输出功率 |
| 33 | 0x77 | cmd_get_output_power | 查询读写器当前输出功率 |
| 34 | 0x78 | cmd_set_frequency_region | 设置读写器工作频率范围 |
| 35 | 0x79 | cmd_get_frequency_region | 查询读写器工作频率范围 |
| 36 | 0x7B | cmd_get_reader_temperature | 查询当前设备的工作温度 |
| 37 | 0x80 | cmd_inventory | 盘存标签 |
| 38 | 0x81 | cmd_read | 读标签 |
| 39 | 0x82 | cmd_write | 写标签 |
| 40 | 0x83 | cmd_lock | 锁定标签 |
| 41 | 0x84 | cmd_kill | 灭活标签 |
| 42 | 0x85 | cmd_set_access_epc_match | 匹配 ACCESS 操作的 EPC 号 |
| 43 | 0x86 | cmd_get_access_epc_match | 查询匹配的 EPC 状态 |
| 44 | 0x87 | cmd_fast_switch_ant_inventory | 多天线轮询盘存 |
| 45 | 0x89 | cmd_real_time_inventory | 盘存标签(实时上传标签数据) |
| 46 | 0x8A | cmd_custom_inventory | 盘存标签(自定义读取数据) |
| 47 | 0x8B | cmd_customized_session_target_inventory | 自定义 session 和 target 盘存 |
| 48 | 0x8C | cmd_stop_inventory | 停止盘存 |
| 49 | 0x8D | cmd_set_select | 设置 select 参数 |
| 50 | 0x8E | cmd_get_select | 获取 select 参数 |
| 51 | 0x90 | cmd_get_inventory_buffer | 查询盘存缓存 |
| 52 | 0x91 | cmd_get_and_reset_inventory_buffer | 查询并重置盘存缓存 |
| 53 | 0x92 | cmd_get_inventory_buffer_tag_count | 查询盘存缓存标签数量 |
| 54 | 0x93 | cmd_reset_inventory_buffer | 重置盘存缓存 |
| 55 | 0x95 | cmd_sm7_write | SM7 加密写 |
| 56 | 0x96 | cmd_sm7_read | SM7 解密读 |
| 57 | 0x97 | cmd_sm7_pk_update | SM7 PK 秘钥更新 |
| 58 | 0x98 | cmd_gb_mul_Seu_Auth | 复旦微 GB 双向认证 |
| 59 | 0xE0 | cmd_check_ant | 获取天线连接状态 |
| 60 | 0xE1 | \ | 读写器温度过高告警 |
| 61 | 0xF3 | cmd_get_uuid | 读取 UUID |

3. 指令解析

3.1. 3EH——设置 CW 开关状态

命令: cmd_set_cw

上位机发送指令数据

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x3E | |
| 5 | Data | 1 | | 0x1 打开 CW 0x0 关闭 CW |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 3E 01 1D

Data 解析: 01: 打开 CW

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x3E | |
| 5 | Data | 1 | | 0x10 设置成功 0x11 失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 3E 10 0E

Data 解析: 10: 设置成功

3.2. 3FH——获取 CW 开关状态

命令: cmd_get_cw

上位机发送指令数据

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |

| | | | | |
|---|----------|---|------|--|
| 4 | Cmd | 1 | 0x3F | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 3F 1E

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x3F | |
| 5 | Data | 1 | | 0x1 CW 已打开 0x0 CW 已关闭 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 3F 01 1C

Data 解析: 01: CW 已打开

3.3. 42H——复旦微 GB 加密通信写

命令: cmd_gb_seu_write(0x42) 执行此命令操作前必须做一次双向认证!

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|-------------------|-------------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x42 | |
| 5.1 | Data | Password | 4 | 标签访问密码 |
| 5.2 | | MemBank 标签存储区域 | 1 | 0 标签信息区 1 编码区 2 安全区 3 用户区 |
| 5.3 | | WordAdd 数据首地址 | 2 | WORD(16 bits)地址。 写入 EPC 存储区域一般从 01 开始, 该区域前 2 个字节存放 PC。 |
| 5.4 | | WordCnt 写入数据长度 | 2 | WORD(16 bits)长度, 数值请参考标签规格 |
| 5.5 | | WordData | WordCnt * 2 | 写入的数据 (可变量), 是写入数据长度的 2 倍。 |

| | | | | |
|---|----------|---|--|--|
| 6 | Checksum | 1 | | |
|---|----------|---|--|--|

读写器返回数据包

返回如下数据包（多条），数量等于写入的标签数量(无重复数据)：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x42 | |
| 5.1 | Data | TagCount | 2 | 写入的数据长度。16 bits。 |
| 5.2 | | DataLen | 1 | 所操作标签的 EPC 长度。(PC +EPC)。单位是字节。 |
| 5.3 | | EPC | 不定长 | 所操作标签 EPC。 PC(2 字节) + EPC (根据标签规格) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果，即错误代码。 |
| 5.5 | | AntID | 1 | 天线号。 |
| 6 | Checksum | 1 | | |

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x42 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x33:写标签错误 0x58:双向认证失败 0x59:双向认证成功 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

3.4. 43H——复旦微 GB 加密通信读

命令: cmd_gb_seu_read(0x43) 执行此命令操作前必须做一次双向认证!

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------------------------|-----|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x0C | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x43 | |
| 5.1 | MemBank 标签存储区域 | 1 | | 0 标签信息区 1 编码区 2 安全区 3 用户区 |
| 5.2 | Data WordAdd 读取数据首地址 | 2 | | 取值范围请参考标签规格 |
| 5.3 | WordCnt 读取数据长度 | 2 | | 字长, WORD(16 bits)长度。 取值范围请参考标签规格书。 |
| 5.4 | Password | 4 | | 标签访问密码 |
| 6 | Checksum | 1 | | |

读写器返回数据包

返回如下数据包(多条), 数量等于读取的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|-------------|-----|------|------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x43 | |
| 5.1 | TagCount | 2 | | 成功操作的标签总数。16 bits。 |
| 5.2 | EPCLen | 1 | | 所操作标签的 EPC 长度。(PC +EPC)。 单位是字节。 |
| 5.3 | Data EPC | 不定长 | | 所操作标签的 EPC。(PC +EPC)。 |
| 5.4 | ReadLen | 2 | | Read 操作的数据长度。单位是字节。 |
| 5.5 | ReadData | 不定长 | | Read 操作的数据 |
| 5.6 | AntID | 1 | | 天线号。 |
| 6 | Checksum | 1 | | |

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x43 | |

| | | | | |
|---|----------|---|-----|--|
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x58:双向认证失败 0x59:双向认证成功 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

3.5. 45H——读标签 GB

命令: cmd_gb_read(0x45)

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------------------------|-----|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x0C | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x45 | |
| 5.1 | MemBank 标签存储区域 | 1 | | 0 标签信息区 1 编码区 2 安全区 3 用户区 |
| 5.2 | Data WordAdd 读取数据首地址 | 2 | | 取值范围请参考标签规格 |
| 5.3 | Data WordCnt 读取数据长度 | 2 | | 字长, WORD(16 bits)长度。 取值范围请参考标签规格书。 |
| 5.4 | Password | 4 | | 标签访问密码 |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x0C
 读写器地址 Address: 0x00
 命令码 Cmd: 0x45

数据包 Data: 0x00 0002 0001 00000000
 //00 标签信息区
 //0002 读取数据首地址
 //0001 读取数据长度
 //00000000 标签访问密码

校验码 Checksum: 0x0C

读写器返回数据包

返回如下数据包（多条），数量等于读取的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|--------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x45 | |
| 5.1 | Data | TagCount | 2 | 成功操作的标签总数。16 bits。 |
| 5.2 | | EPCLen | 1 | 所操作标签的 EPC 长度。(PC +EPC)。单位是字节。 |
| 5.3 | | EPC | 不定长 | 所操作标签的 EPC。(PC +EPC)。 |
| 5.4 | | ReadLen | 2 | Read 操作的数据长度。单位是字节。 |
| 5.5 | | ReadData | 不定长 | Read 操作的数据 |
| 5.6 | | AntID | 1 | 天线号。 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x0C
 读写器地址 Address: 0x00
 命令码 Cmd: 0x45
 数据包 Data: 0x0001 12 3000E280689400005016A9878056 0002 D578 01

//从高位到低 0001 为操作成功的总数量 1 个，
 //12 为数据操作的有效数据
 //3000 是读到数据的 PC，
 // E280689400005016A9878056 是标签 EPC
 //0002 Read 操作的数据长度
 //D578 read 操作数据
 //01 是天线号

校验码 Checksum: 0x0E

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x81 | |

| | | | | |
|---|----------|---|-----|--|
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x04
 读写器地址 Address: 0x00
 命令码 Cmd: 0x45
 数据包 Data: 0x36
 校验码 Checksum: 0xE1

3.6. 46H——写标签 GB

命令: cmd_gb_write(0x46)

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固 定 值 | 备注 |
|-----|---------|-------------------|-------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x46 | |
| 5.1 | Data | Password | 4 | 标签访问密码 |
| 5.2 | | MemBank 标签存储区域 | 1 | 0 标签信息区 1 编码区 2 安全区 3 用户区 |
| 5.3 | | WordAdd 数据首地址 | 2 | WORD(16 bits)地址。 写入 EPC 存储区域一般从 01 开始, 该区域前 2 个字节存放 PC。 |
| 5.4 | | WordCnt | 2 | WORD(16 bits)长度, 数值请参考标签 |

| | | | | |
|-----|----------|----------|------------|--------------------------|
| | | 写入数据长度 | | 规格 |
| 5.5 | | WordData | WordCnt *2 | 写入的数据（可变量），是写入数据长度的 2 倍。 |
| 6 | Checksum | | 1 | |

读写器返回数据包

返回如下数据包（多条），数量等于写入的标签数量(无重复数据)：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x46 | |
| 5.1 | Data | TagCount | 2 | 写入的数据长度。16 bits。 |
| 5.2 | | DataLen | 1 | 所操作标签的 EPC 长度。（PC +EPC）。单位是字节。 |
| 5.3 | | EPC | 不定长 | 所操作标签 EPC。 PC(2 字节) + EPC (根据标签规格) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果，即错误代码。 |
| 5.5 | | AntID | 1 | 天线号。 |
| 6 | Checksum | | 1 | |

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x46 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x33:写标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | | 1 | |

3.7. 47H——锁定标签 GB

命令: cmd_gb_lock

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------------------------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x09 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x47 | |
| 5.1 | Password | 4 | | 标签访问密码 |
| 5.2 | MemBank 标签存储区域 | 1 | | 0x00 标签信息区 0x01 编码区 0x02 安全区 0x03 用户区 |
| 5.3 | Data LockType 锁操作类型 | 1 | | 0x00 可读可写 0x01 可读不可写 0x02 不可读可写 0x03 不可读不可写 0x04 保留 0x05 不需要鉴别 0x06 需要鉴别不需要安全通信 0x07 需要鉴别需要安全通信 |
| 6 | Checksum | 1 | | |

读写器返回数据包:

返回如下数据包(多条), 数量等于写入的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|------------------|-----|------|--------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x47 | |
| 5.1 | TagCount | 2 | | 成功操作的标签总数。16 bits。 |
| 5.2 | DataLen | 1 | | 所操作标签的有效数据长度。 (PC+CRC+EPC)。单位是字节。 |
| 5.3 | Data LockData | 不定长 | | 所操作标签有效数据。 PC(2 字节) + EPC(根据标签规格) |
| 5.4 | ErrCode | 1 | | 所操作标签的操作结果, 即错误代码。 |
| 5.5 | AntID | 1 | | 高 6 位是第一次读取的频点参数, 低 2 位是天线号。 |
| 6 | Checksum | 1 | | |

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x47 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

3.8. 49H——灭活标签 GB

命令: cmd_gb_kill

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----|----------|--------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x49 | |
| 5.1 | Data | 4 | Password | 标签灭活密码 |
| 6 | Checksum | 1 | | |

读写器返回数据包

返回如下数据包（多条），数量等于销毁的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|-----|----------|--------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x49 | |
| 5.1 | Data | 2 | TagCount | 成功操作的标签总数。16 bits。 |
| 5.2 | Data | 1 | DataLen | 所操作标签的有效数据长度。（PC |

| | | | | |
|-----|----------|----------|-----|---------------------------------------|
| | | | | +EPC)。单位是字节。 |
| 5.3 | | KillData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果, 即错误代码。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数, 低 2 位是天线号。 |
| 6 | Checksum | | 1 | |

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x49 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

3.9. 4AH——保存参数

命令: cmd_reader_para_save

此命令会将配置的参数写进 FLASH 中, 数据掉电不丢失。在所有设置参数的指令中, 除设置通信带宽指令外, 其他参数配置后均需使用此命令, 以使参数永久保存。若需配置多个参数, 则所有参数配置完成后, 再发送一条保存参数指令, 所配置的参数会全部存入 FLASH 中。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |

| | | | | |
|---|----------|---|------|--|
| 4 | Cmd | 1 | 0x4a | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 4A 13

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4a | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 4A 10 02

Data 解析: 10: 操作成功

3.10. 4BH——恢复读写器默认配置

命令: cmd_reader_para_reset

此命令会将读写器除系统时间和 MAC 地址之外的所有参数恢复成默认值,包括 RFID 配置参数。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4b | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 4B 12

Data 解析: 无

读写器无返回数据包

3.11. 4CH——读写器应用软件升级

命令: cmd_reader_app_upgrade

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|--------------------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4c | |
| 5.1 | Data | PacketSerialNumber | 4 | 升级数据包序列号, 以 0x00000000 开始 0xFFFFFFFF 结束 |
| 5.2 | | Status | 1 | 0, 升级过程中; 1, 升级结束 (最后一包) |
| 5.3 | | upgradeDataLen | 1 | =n, upgradeData 的长度 |
| 5.4 | | upgradeData | 不定长 | 升级数据内容 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: 无

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|--------------------|------|--------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 8 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4c | |
| 5.1 | Data | PacketSerialNumber | 4 | 上位机发送的升级包序号 |
| 5.2 | | Status | 1 | 0 成功 1 失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: 无

Data 解析: 无

3.12. 4DH——基带固件升级

命令: cmd_baseband_firmware_upgrade

同[读写器应用软件升级](#)

参考示例:

串口数据: 无

Data 解析: 无

3.13. 4EH——设置心跳开关和周期

命令: cmd_set_keepalive

上位机发送指令数据

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 6 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4E | |
| 5.1 | Data | 开关 | 1 | 0x1 打开心跳包 0x0 关闭心跳包 |
| 5.2 | | 心跳周期 | 2 | 最小值为 100ms，最大 655350ms，最小修改单位 10ms，大端模式 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 06 00 4E 01 13 88 70

Data 解析: 01: 使能心跳包

13 88: 心跳周期 5000(大端模式)

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4E | |
| 5 | Data | 1 | 0x10 | 操作成功 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 4E 10 FE

Data 解析: 10: 操作成功

3.14. 4FH——获取心跳开关和心跳周期

命令: cmd_get_keepalive

上位机发送指令数据

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4F | |

| | | | | |
|---|----------|---|--|--|
| 6 | Checksum | 1 | | |
|---|----------|---|--|--|

参考示例:

串口数据: A0 03 00 4F 0E

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|------|------|--------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 6 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x4F | |
| 5.1 | Data | 开关 | 1 | 0x1 心跳包已打开 0x0 心跳包已关闭 |
| 5.2 | | 心跳周期 | 2 | (大端模式) |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 06 00 4F 00 27 10 D4

Data 解析: 00: 心跳关

27 10: 心跳周期 10000(大端模式)

3.15. 5AH——获取 Session 和 Target

命令: cmd_get_session_target

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5A | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 5A 03

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5A | |

| | | | | |
|-----|----------|---------|---|---|
| 5.1 | Data | Session | 1 | 指定盘存的 session 0 S0 1 S1 2 S2 3 S3 |
| 5.2 | | Target | 1 | 指定盘存的 Inventoried Flag 0 A 1 B |
| 6 | Checksum | | 1 | |

参考示例:

串口数据: A0 05 00 5A 02 01 FE

Data 解析: 02: session-s2

01: target-B

3.16. 5BH——设置 Session 和 Target

命令: cmd_set_session_target

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|---------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5B | |
| 5.1 | Data | Session | 1 | 指定盘存的 session 0 S0 1 S1 2 S2 3 S3 |
| 5.2 | | Target | 1 | 指定盘存的 Inventoried Flag 0 A 1 B |
| 6 | Checksum | | 1 | |

参考示例:

串口数据: A0 05 00 5B 02 01 FD

Data 解析: 02: session-s2

01: target-B

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | |
|---|----------|---|------|------|
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5B | |
| 5 | Data | 1 | 0x10 | 操作成功 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 5B 10 F1

Data 解析: 10: 操作成功

3.17. 5CH——获取 TX RF 占空比

命令: cmd_get_tx_time

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5C | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 5C 01

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|---------------|------|---------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5C | |
| 5.1 | Tx_RF | Tx_RF_ontime | 2 | 步进为 1ms |
| 5.2 | | Tx_RF_offtime | 2 | 步进为 1ms |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 07 00 5C 00 64 00 64 35

Data 解析: 00 64: Tx_RF_ontime(大端模式)

00 64: Tx_RF_offtime(大端模式)

3.18. 5DH——设置 TX RF 时间

命令: cmd_set_tx_time

该命令主要用来改变 TX RF 占空比，调整功耗。TX 发射时间主要分为 TX RF 使能时间和 TX RF 关闭时间。时间最小刻度 10ms，步进为 1ms。

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|---------------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5D | |
| 5.1 | Tx_RF | Tx_RF_ontime | 2 | 默认值 100ms,步进为 1ms, 可以两个字段同时设为 0, 表示关闭该功能 |
| 5.2 | | Tx_RF_offtime | 2 | |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 07 00 5D 00 64 00 64 34

Data 解析：00 64：Tx_RF_ontime(大端模式)

00 64：Tx_RF_offtime(大端模式)

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5D | |
| 5 | Data | 1 | | 0x10:设置成功 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 5D 10 EF

Data 解析：10：操作成功

3.19. 5FH——设置 4 天线读写器射频输出功率

命令：cmd_set_4_ant_power

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|-----------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5F | |
| 5 | Data | 4 | | 第 1 字节：天线 1 的 RF 输出功率 |

| | | | | |
|---|----------|---|--|---|
| | | | | 第2字节: 天线2的RF输出功率 第3字节: 天线3的RF输出功率 第4字节: 天线4的RF输出功率 取值范围0-33, 单位dBm |
| 6 | Checksum | 1 | | |

说明: 新的读写器配置立即生效。

参考示例:

串口数据: A0 07 00 5F 0A 21 21 21 8D

Data 解析:

0A: 天线1的RF输出功率(10dBm)

21: 天线2的RF输出功率(33dBm)

21: 天线3的RF输出功率(33dBm)

21: 天线4的RF输出功率(33dBm)

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x5F | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 0x25:参数超范围 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 5F 10 ED

Data 解析: 10: 操作成功

3.20. 60H——查询读写器 GPIO 状态

命令: cmd_read_gpio_value

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x60 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 60 FD

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|----------|------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x60 | |
| 5 | Data | 1 | GPIO1 状态 | 0: 低 |
| | | 1 | GPIO2 状态 | 1: 高 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 05 00 60 00 00 FB

Data 解析: 00: GPIO1 状态-低

00: GPIO2 状态-低

3.21. 61H——设置读写器 GPIO 状态

命令: cmd_set_gpio_value

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|----------|--------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x61 | |
| 5 | Data | 1 | GPIO 口选择 | 0x03:GPIO0 0x04:GPIO1 |
| | | 1 | GPIO 输出值 | 0: 低 1: 高 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 05 00 61 03 01 F6

Data 解析: 03: GPIO0

01: 设置为高

参考示例:

串口数据: A0 05 00 61 04 01 F5

Data 解析: 04: GPIO1

01: 设置为高

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x61 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 61 10 EB

Data 解析: 10: 操作成功

3.22. 66H——设置读写器临时射频输出功率

备注:4 此命令为调试命令, 设置 4 天线功率请使用 0x5F 命令

命令: cmd_set_temporary_output_power

操作成功后输出功率值将不会被保存在内部的 Flash 中, 重新启动或断电后输出功率将恢复至内部 Flash 中保存的输出功率值。此命令的操作速度非常快, 并且不写 Flash, 从而不影响 Flash 的使用寿命, 适合需要反复切换射频输出功率的应用。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x66 | |
| 5 | Data | 1 | | RF 临时输出功率, 取值范围 3-33, 单位 dBm (需设置 ≥ 3 的值, 最大值: 601 系列支持 20dbm, 606L 26dbm, 608 33dbm, 四天线, 八天线设置相同) |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 66 11 E5

Data 解析: 11: RF 临时输出功率

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | |
|---|----------|---|------|--------------------------------------|
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x66 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 0x25:参数超范围 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 66 10 EB

Data 解析: 10: 操作成功

3.23. 69H——设置射频链路的通讯速率

命令: cmd_set_rf_link_profile

操作成功后读写器会重新启动, 配置将直接保存在内部的 Flash 中, 断电后不丢失, 配置此参数之前若配置了其他参数, 则之前配置的参数也会被写进 Flash 中。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x69 | |
| 5 | Data | 1 | | ProfileID 0xD0 Miller 40KHz tari 25us 0xD1 FMO 200KHz tari 6.25us 0xD2 FMO 200KHz tari 12.5us 0xD3 FMO 200KHz tari 25us 0xD4 Miller4 200KHz tari 6.25us 0xD5 Miller4 200KHz tari 12.5us 0xD6 Miller4 200KHz tari 25us 0xD7 Miller_4 250KHz tari 6.25us 0xD8 FMO 640KHz tari 6.25us 0xD9 FMO 40KHz tari 25us 0xDA GB FMO 64KHz 0xDB GB Miller 128KHz 0xDC GB FMO 128KHz |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 69 D7 1C

Data 解析: D7: Miller_4 250KHz tari 6.25us

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x69 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 69 10 E3

Data 解析: 10: 操作成功

3.24. 6AH——查询射频链路的通讯速率

命令: cmd_get_rf_link_profile

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x6a | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 6A F3

Data 解析: 无

读写器返回数据包

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x6a | |
| 5 | Data | 1 | | 成功返回, ProfileID 0xD0 Miller 40KHz tari 25us 0xD1 FMO 200KHz tari 6.25us 0xD2 FMO 200KHz tari 12.5us 0xD3 FMO 200KHz tari 25us 0xD4 Miller4 200KHz tari 6.25us 0xD5 Miller4 200KHz tari 12.5us 0xD6 Miller4 200KHz tari 25us 0xD7 Miller_4 250KHz tari 6.25us 0xD8 FMO 640KHz tari 6.25us |

| | | | | |
|---|----------|---|--|--|
| | | | | 0xD9 FMO 40KHz tari 25us 0xDA GB FMO 64KHz 0xDB GB Miller 128KHz 0xDC GB FMO 128KHz |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 6A D6 1C

Data 解析: D6: Miller4 200KHz tari 25us

3.25. 70H——复位读写器

命令: cmd_reset

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x70 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 70 ED

Data 解析: 无

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x70 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 0x20:MCU 复位失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 70 10 DC

Data 解析: 10: 操作成功

3.26. 71H——设置串口通讯波特率

命令: cmd_set_uart_baudrate

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----|-----|-----|----|
|----|----|-----|-----|----|

| | | | | |
|---|----------|---|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x71 | |
| 5 | Data | 1 | | 0x00:1200 0x01:2400 0x02:4800 0x03:9600 0x04:14400 0x05:19200 0x06:38400 0x07:57600 0x08:76800 0x09:115200 0x0a:128000 0x0b:230400 0x0c:256000 0x0d:460800 |
| 6 | Checksum | 1 | | |

说明：读写器成功收到此命令帧后，命令立即生效，将读写器波特率更新。

参考示例：

串口数据：A0 04 00 71 09 E2

Data 解析：09：115200

读写器返回数据包

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x71 | |
| 5 | Data | 1 | 操作码 | 0x10:操作成功 0x11:操作失败 0x4a:波特率超范围 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 71 10 DB

Data 解析：10：操作成功

3.27. 72H——查询读写器固件版本

命令：cmd_get_firmware_version

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x72 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 72 EB

Data 解析: 无

读写器返回数据包

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 6 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x72 | |
| 5.1 | Data | Major | 1 | 固件主版本号 |
| 5.2 | | Minor | 1 | 固件次版本号 |
| 5.3 | Model | 1 | | 0x01:UCM601 0x02:UCM602 0x06:UCM606 0x08:UCM608 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 06 00 72 02 04 01 E1

Data 解析: 02: 主版本号

04: 次版本号

01: UCM601

3.28. 73H——设置读写器地址

命令: cmd_set_reader_address

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x73 | |
| 5 | Data | 1 | | 配置读写器地址, 配置范围 0-255 |
| 6 | Checksum | 1 | | |

说明: 新的读写器地址立即生效。

参考示例:

串口数据: A0 04 00 71 10 DB

Data 解析: 10: 操作成功

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x73 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x41:不支持该配置参数 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 73 10 D9

Data 解析: 10: 操作成功

3.29. 74H——设置读写器工作天线

命令: cmd_set_work_antenna

上位机发送指令数据--**单天线**:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x74 | |
| 5 | Data | 1 | | 天线号 AntennaID 天线 1, 0x01 |
| 6 | Checksum | 1 | | |

说明: 新的读写器配置立即生效。

参考示例:

串口数据: A0 04 00 74 01 E7

Data 解析: 01: 使能 1 号天线

上位机发送指令数据 -- **多天线** (仅 UCM608-4 和 UCM608-8 有效):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|-------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x74 | |
| 5 | Data | 1 | 0x00 | 固定值 |
| | | 8 | | 1~8 字节对应 1~8 号天线使能位, 打开 |

| | | | | |
|---|----------|---|--|---------------------------------------|
| | | | | 天线对应字节设置为 1,关闭设置为 0, 4 天线模组仅前面 4 字节有效 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 0C 00 74 00 01 01 00 00 00 00 00 00 DE

Data 解析: 00: 固定值

01: 使能 1 号天线

01: 使能 2 号天线

00: 除能 3 号天线

00: 除能 4 号天线

00: 除能 5 号天线

00: 除能 6 号天线

00: 除能 7 号天线

00: 除能 8 号天线

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x74 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x41:不支持该配置参数 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 74 10 D8

Data 解析: 10: 操作成功

3.30. 75H——查询当前工作天线

命令: cmd_get_work_antenna

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x75 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 75 E8

Data 解析: 无

读写器返回数据包

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x75 | |
| 5.1 | Data | 1 | | 天线号 AntennaID 天线 1, 0x01 天线 2, 0x02 天线 3, 0x03 天线 4, 0x04 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 75 01 E6

Data 解析: 01: 天线 1 工作

3.31. 76H——设置单天线读写器射频输出功率

命令: cmd_set_output_power

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x76 | |
| 5 | Data | 1 | | RF 输出功率, 取值范围 0-33, 单位 dBm (601 系列支持 20dbm, 606L 26dbm, 608 33dbm, 四天线, 八天线设置相同) |
| 6 | Checksum | 1 | | |

说明: 新的读写器配置立即生效。

参考示例:

串口数据: A0 04 00 76 0A DC

Data 解析: 0A: 射频输出功率(10dBm)

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x76 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 |

| | | | | |
|---|----------|---|--|-------------------------|
| | | | | 0x11:操作失败 0x25:参数超范围 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 76 10 D6

Data 解析: 10: 操作成功

3.32. 77H——查询读写器当前输出功率

命令: cmd_get_output_power

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x77 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 77 E6

Data 解析: 无

读写器返回数据包:

情况一: 单天线的功率

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x77 | |
| 5 | Data | 1 | | 天线 1 功率, 范围 0-33, 单位 dBm |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 77 14 D1

Data 解析: 14: 天线 1 功率

情况二: 4 天线的功率

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|--------|------|--------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x77 | |
| 5.1 | Data | Power1 | 1 | 天线 1 功率, 范围 0-33, 单位 dBm |
| 5.2 | | Power2 | 1 | 天线 2 功率, 范围 0-33, 单位 dBm |

| | | | |
|-----|----------|---|--------------------------|
| 5.3 | Power3 | 1 | 天线 3 功率, 范围 0-33, 单位 dBm |
| 5.4 | Power4 | 1 | 天线 4 功率, 范围 0-33, 单位 dBm |
| 6 | Checksum | 1 | |

参考示例:

串口数据: A0 07 00 77 0A 21 21 21 75

Data 解析: 0A: 天线 1 功率
 21: 天线 2 功率
 21: 天线 3 功率
 21: 天线 4 功率

3.33. 78H——设置读写器工作频率范围

命令: cmd_set_frequency_region

上位机发送指令数据:

射频频谱的定义有两种方法。

方法一: 使用系统默认的频点 (参见频率参数对应表)

CHN-1 是 840-845MHz

CHN-2 是 920-925Mhz

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------------------------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 6 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x78 | |
| 5.1 | Region 射频规范 | 1 | | 0x01 FCC 0x02 ETSI 0x03 CHN_1 0x05 CHN_2 使用系统默认频点时 (射频规范 1,2,3,5), 频率起始点和结束点采用默认配置 |
| 5.2 | Data StartFreq 频率起始点 | 1 | | 可以在射频规范的频率范围内再设置跳频的范围。参数所对应的频率请参见频率参数对应表。参数的设置规则为: 1. 起始频率与结束频率不能超过射频规范的范围。 2. 起始频率必须低于结束频率。 3. 起始频率等于结束频率则定频发射。 |
| 5.3 | EndFreq 频率结束点 | 1 | | |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 06 00 78 01 07 3B 9F

Data 解析: 01: 射频规范 FCC

07: FCC 规范频率起始点(902.250MHz)

3B: FCC 规范频率结束点(928.250MHz)

方法二: 用户自定义频谱

注: 使用自定义频谱时(射频规范 4), 根据频率起始点和结束点设置频点范围

| 序号 | 字段 | 字节数 | 固定值 | 备注 | |
|-----|----------|----------------------|------|----|--|
| 1 | Head | 1 | 0xA0 | | |
| 2 | Len | 1 | 10 | | |
| 3 | Address | 1 | | | |
| 4 | Cmd | 1 | 0x78 | | |
| 5.1 | Data | Region 射频规范 | 1 | 4 | 固定值 |
| 5.2 | | FreqSpace 频点间隔 | 2 | | 频点间隔 = 最小频点间隔为 25KHz |
| 5.3 | | FreqQuantity 频率数量 | 1 | | 包含起始频率的频点数量, 1 为以起始频率定频发射。此参数必须大于 0 |
| 5.4 | | StartFreq 频率起始点 | 3 | | 单位为 KHz。16 进制数高位在前。例如 915000KHz 则发送 0D F6 38 |
| 6 | Checksum | 1 | | | |

参考示例:

串口数据: A0 0A 00 78 04 00 19 01 0D BB A0 58

Data 解析: 04: 射频规范--自定义

00 19: 频段间隔(25KHz)

01: 频点数量

0D BB A0: 起始频点(900000KHz)

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x78 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 78 10 D4

Data 解析: 10: 操作成功

3.34. 79H——查询读写器工作频率范围

命令: cmd_get_frequency_region

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x79 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 79 E4

Data 解析: 无

读写器返回数据根据射频规范主要分两类:

系统默认频点返回数据

CHN-1 是 840-845MHz

CHN-2 是 920-925Mhz

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------------------------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 6 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x79 | |
| 5.1 | Region 射频规范 | 1 | | 0x01 FCC 0x02 ETSI 0x03 CHN_1 0x05 CHN_2 |
| 5.2 | Data StartFreq 频率起始点 | 1 | | 跳频频率范围的低点 |
| 5.3 | EndFreq 频率结束点 | 1 | | 跳频频率范围的高点 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 06 00 79 05 2B 35 7C

Data 解析: 05: 射频规范 CHN_2

07: CHN_2 规范频率起始点(920MHz)

3B: CHN_2 规范频率结束点(925MHz)

自定义频点返回数据

| 序号 | 字段 | 字节数 | 固定值 | 备注 | |
|-----|----------|----------------------|------|----|--|
| 1 | Head | 1 | 0xA0 | | |
| 2 | Len | 1 | 10 | | |
| 3 | Address | 1 | | | |
| 4 | Cmd | 1 | 0x79 | | |
| 5.1 | Data | Region 射频规范 | 1 | 4 | 固定值 |
| 5.2 | | FreqSpace 频点间隔 | 2 | | 频点间隔 = 最小频点间隔为 25KHz |
| 5.3 | | FreqQuantity 频率数量 | 1 | | 包含起始频率的频点数量, 1 为以起始频率定频发射。此参数必须大于 0 |
| 5.4 | | StartFreq 频率起始点 | 3 | | 单位为 KHz。16 进制数高位在前。例如 915000KHz 则发送 0D F6 38 |
| 6 | Checksum | 1 | | | |

参考示例:

串口数据: A0 0A 00 79 04 00 19 01 0D BB A0 57

Data 解析: 04: 射频规范--自定义

00 19: 频段间隔(25KHz)

01: 频点数量

0D BB A0: 起始频点(900000KHz)

3.35. 7AH——设置蜂鸣器开关

命令: cmd_get_reader_temperature

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x7a | |
| 5 | Data | 1 | | 0: 关闭蜂鸣器 1: 打开蜂鸣器 |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 7A 00 E2

Data 解析: 00:关闭蜂鸣器

读写器返回数据包:

成功返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |

| | | | | |
|-----|----------|---|------|----------------------|
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x7a | |
| 5.1 | Data | 1 | 错误码 | 10: 操作成功 11: 操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 7A 10 D2

Data 解析: 10: 操作成功

3.36. 7BH——查询当前设备的工作温度

命令: cmd_get_reader_temperature

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x7b | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 7B E2

Data 解析: 无

读写器返回数据包:

成功返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-------------|------|--------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x7b | |
| 5.1 | Data | PlusMinus | 1 | 0 零下 1 零上 |
| 5.2 | | Temperature | 1 | 摄氏度 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 05 00 7B 01 1D C2

Data 解析: 01: 零上

1D: 温度 29°

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | |
|---|----------|---|------|-----------|
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x7b | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 7B 11 D0

Data 解析: 11: 操作失败

3.37. 80H——盘存标签

命令: cmd_inventory

缓存模式: 读写器收到此命令后, 进行多标签识别操作。标签数据存入读写器缓存区, 使用提取缓存指令可获得标签数据, 详见: [缓存操作命令](#), 在收到停止命令前, 读写器将持续盘存。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x80 | |
| 5 | Data | 1 | | 单天线产品: 天线号固定为 0x01 多天线产品: 单天线模式: 天线号 0x01~0x04 多天线模式: 天线号固定为 0x00 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 80 02 DA

Data 解析: 02: 天线 2

读写器返回数据包

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x80 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 0x36:无可操作标签 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 80 10 CC

Data 解析: 10: 操作成功

无可操作标签时, 继续返回:

串口数据: A0 04 00 80 36 A6

Data 解析: 36: 无可操作标签

有可操作标签时, 继续返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x80 | |
| 5 | Data | 2 | | 识别标签的总数量, 根据 EPC 号来区分标签, 相同 EPC 号的标签将被视为同一张标签。若未清空缓存, 标签数量为多次盘存操作的数量累加。 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 05 00 80 00 02 D9

Data 解析: 00 02: 识别到的标签总数量

3.38. 81H——读标签 6C

命令: cmd_read

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------------------------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x0E | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x81 | |
| 5.1 | MemBank 标签存储区域 | 1 | | 0 RESERVED 1 EPC 2 TID 3 USER |
| 5.2 | Data WordAdd 读取数据首地址 | 4 | | 取值范围请参考标签规格 |
| 5.3 | WordCnt 读取数据长度 | 2 | | 字长, WORD(16 bits)长度。 取值范围请参考标签规格书。 |
| 5.4 | Password | 4 | | 标签访问密码 |

| | | | | |
|---|----------|---|--|--|
| 6 | Checksum | 1 | | |
|---|----------|---|--|--|

参考示例:

串口数据: A0 0E 00 81 01 00 00 00 02 00 06 00 00 00 00 C8

Data 解析: 01: 标签存储区域 EPC

00 00 00 02: 数据首地址

00 06: 读取数据长度(6 个字, 12 个字节)

00 00 00 00: 访问密码

读写器返回数据包

返回如下数据包, 每次返回一条数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x81 | |
| 5.1 | Data | TagCount | 2 | 成功操作的标签总数(大端模式) |
| 5.2 | | DataLen | 1 | 有效数据长度。(PC+CRC+EPC+读取的数据)。单位是字节。 |
| 5.3 | | ReadData | 不定长 | 所操作标签的有效数据。 PC (2 字节) + EPC (根据标签规格) + CRC (2 字节) + 读取的数据。(如果 epc 起始从 0 开始读取的话顺序是 crc+pc+epc) //(PC(2 字节) + EPC + CRC(2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ReadLen | 2 | Read 操作的数据长度。单位是字节。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数, 低 2 位是天线号。 |
| 5.6 | | ReadCount | 1 | 该标签被成功操作的次数。 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 26 00 81 00 01 1C 34 00 E2 00 00 02 02 03 04 05 06 07 08 09 E1 9F E2 00 00 02 02

03 04 05 06 07 08 09 00 0C 03 01 B8

Data 解析: 00 01: 成功操作的标签总数

1C: 有效数据长度。(PC+EPC+CRC+读取的数据)

34 00: PC

E2 00 00 02 02 03 04 05 06 07 08 09: EPC

E1 9F: CRC

E2 00 00 02 02 03 04 05 06 07 08 09: 读取的数据

00 0C: 读取的数据长度 12 字节

03: 天线号

01: 成功读取次数

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x81 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 81 00 40 9B

Data 解析：40：访问标签错误或密码错误

3.39. 82H——写标签 6C

命令：cmd_write(0x82)

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-------------------|---------------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x82 | |
| 5.1 | Data | Password | 4 | 标签访问密码 |
| 5.2 | | MemBank 标签存储区域 | 1 | 0 RESERVED 1 EPC 2 TID 3 USER |
| 5.3 | | WordAdd 数据首地址 | 4 | WORD(16 bits)地址。 写入 EPC 存储区域一般从 02 开始， 该区域前四个字节存放 PC+CRC。 |
| 5.4 | | WordCnt 写入数据长度 | 2 | WORD(16 bits)长度，数值请参考标 签规格 |
| 5.5 | | WordData | WordCnt *2 | 写入的数据（可变量），是写入数据 长度的 2 倍。 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 10 00 82 00 00 00 00 01 00 00 00 02 00 01 00 67 63

Data 解析: 01: 标签存储区域 EPC

00 00 00 02: 数据首地址

00 01: 写入长度(1 个字, 2 个字节)

00 67: 写入的数据

读写器返回数据包

返回如下数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x82 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 82 10 CA

Data 解析: 10: 操作成功

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x82 | |
| 5.1 | Data | TagCount | 2 | 保留 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。 (PC+EPC+CRC)。单位是字节。 |
| 5.3 | | WriteData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格) + CRC (2 字节) (PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果, 即错误代码。 |
| 5.5 | | AntID | 1 | 天线号。 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 18 00 82 00 00 10 30 00 E2 80 69 95 00 00 50 05 CF 23 28 62 46 81 10 01 7D

Data 解析: 00 00: 保留

10: 有效数据长度(PC+EPC+CRC)

30 00: PC

E2 80 69 95 00 00 50 05 CF 23 28 62: EPC

46 81: CRC

10: 操作结果: 成功

01: 天线号

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x82 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x33:写标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 82 41 99

Data 解析: 41: 无效的参数

3.40. 83H——锁定标签 6C

命令: cmd_lock

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|-------------------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 9 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x83 | |
| 5.1 | Password | 4 | | 标签访问密码 |
| 5.2 | MemBank 标签存储区域 | 1 | | 0x01 User Memory 0x02 TID Memory 0x03 EPC Memory 0x04 Access Password 0x05 Kill Password |
| 5.3 | LockType 锁操作类型 | 1 | | 0x00 开放 0x01 锁定 (标签锁定前提: 访问密码不为全 0, 需提前修改访问密码, 注: NXP U9 标签无访问密码所以无法执行锁定操作) 0x02 永久开放 |

| | | | | |
|---|----------|---|--|-----------|
| | | | | 0x03 永久锁定 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 09 00 83 00 00 00 00 01 00 D3

Data 解析: 00 00 00 00: 标签访问密码

01: 标签存储区域 user memory

00: 锁操作类型, 开放

读写器返回数据包:

返回如下数据包(多条), 数量等于写入的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x83 | |
| 5.1 | Data | TagCount | 2 | 保留 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。(PC+CRC+EPC)。单位是字节。 |
| 5.3 | | LockData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC(根据标签规格) + CRC(2 字节) (PC(2 字节) + EPC + CRC(2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果, 即错误代码。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数, 低 2 位是天线号。 |
| 5.6 | | LockCount | 1 | 该标签被操作的次数 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 18 00 83 00 00 10 34 00 E2 00 00 02 02 03 04 05 06 07 08 09 E1 9F 10 01 E0

Data 解析: 00 00: 保留

10: 有效数据长度(PC+EPC+CRC)

34 00: PC

E2 00 00 02 02 03 04 05 06 07 08 09: EPC

E1 9F: CRC

10: 操作成功

01: 该标签被操作的次数

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |

| | | | | |
|---|----------|---|------|--|
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x83 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 83 36 A3

Data 解析: 36: 无可操作标签

3.41. 84H——灭活标签 6C

命令: cmd_kill

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|--------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x84 | |
| 5.1 | Data | Password | 4 | 标签灭活密码 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 07 00 84 00 00 00 01 A3

Data 解析: 00 00 00 01: 标签灭活密码

读写器返回数据包

返回如下数据包(多条), 数量等于销毁的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|----------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x84 | |
| 5.1 | Data | TagCount | 2 | 保留 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。 (PC+CRC+EPC)。单位是字节。 |
| 5.3 | | KillData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC(根据标签规格) + CRC(2 字节) |

| | | | | |
|-----|----------|-----------|---|--|
| | | | | (PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果，即错误代码。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数，低 2 位是天线号。 |
| 5.6 | | KillCount | 1 | 销毁标签操作只能为 1 |
| 6 | Checksum | | 1 | |

参考示例：

串口数据：A0 19 00 84 00 00 10 30 00 E2 00 00 00 00 00 04 16 A9 87 50 56 22 8E 10 01 01

Data 解析：00 00：保留

10：有效数据长度(PC+EPC+CRC)

30 00：PC

E2 00 00 00 00 00 04 16 A9 87 50 56：EPC

22 8E：CRC

10：操作成功

01：天线号

01：销毁标签操作次数

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x84 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 84 36 A2

Data 解析：36：无可操作标签

3.42. 85H——匹配 ACCESS 操作的 EPC 号

命令：cmd_set_access_epc_match

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | |
|-----|----------|--------|------|-------------------------------------|
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x85 | |
| 5.1 | Data | Mode | 1 | 0 EPC 匹配一直有效，直到下一次刷新 1 清除 EPC 匹配 |
| 5.2 | | EpcLen | 1 | Epc 长度 |
| 5.3 | | Epc | 不定长 | EPC 数据，由 EpcLen 个字节组成 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 11 00 85 00 0C E2 80 69 95 00 00 40 02 D7 43 70 AA E8

Data 解析：00：EPC 匹配一直有效

0C：Epc 长度 12

E2 80 69 95 00 00 40 02 D7 43 70 AA：EPC 数据

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x85 | |
| 5 | Data | 1 | 错误码 | 0x10:设置成功 0x4c:epc 长度超长 0x4d:epc 长度错误 0x4e:模式错误 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 82 10 CA

Data 解析：10：操作成功

3.43. 86H——查询匹配的 EPC 状态

命令：cmd_get_access_epc_match

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x86 | |
| 5 | Checksum | 1 | | |

参考示例：

串口数据：A0 03 00 86 D7

Data 解析：无

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|--------|------|-----------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x86 | |
| 5.1 | Data | Status | 1 | 1 有匹配 0 无匹配 |
| 5.2 | | EpcLen | 1 | 匹配的 Epc 长度，无匹配时不返回此数据 |
| 5.3 | | Epc | 不定长 | 匹配的 EPC 号，无匹配时不返回此数据 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
数据包长 Len: 0x11
读写器地址 Address: 0x00
命令码 Cmd: 0x86
数据包 Data: 0x01 0C300833B2DDD9014000000000A6
//01 status 为 EPC 有匹配
//0C EPC 的长度
//300833B2DDD9014000000000EPC 号
校验码 Checksum: 0x06

3.44. 87H——多天线轮询盘存

命令：cmd_fast_switch_ant_inventory

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|---------|------|----------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x87 | |
| 5.1 | Data | Ant num | 1 | 轮询的天线数量 |
| 5.2 | | Ant1 | 1 | 首先轮询的天线，设置为 0 可跳过。 |
| 5.3 | | Stay1 | 1 | 天线重复轮询的次数。每个天线可单独配置。 |
| 5.4 | | Ant2 | 1 | 第二个轮询的天线，设置为 0 可跳过。 |

| | | | | |
|------|----------|---|--|--|
| 5.5 | Stay2 | 1 | | 天线重复轮询的次数。每个天线可单独配置。 |
| 5.6 | Antn | 1 | | 第 n 个轮询的天线，设置为 0 可跳过。 |
| 5.7 | Stayn | 1 | | 天线重复轮询的次数。每个天线可单独配置。 |
| 5.8 | Interval | 1 | | 天线间的休息时间。单位是 mS。休息时无射频输出，可降低功耗。 |
| 5.9 | Session | 1 | | 指定盘存的 session，00 为 S0，01 为 S1，02 为 S2，03 为 S3。 |
| 5.10 | Flag | 1 | | 指定盘存的 Inventoried Flag，00 为 A，01 为 B |
| 5.11 | Repeat | 1 | | 重复以上天线切换顺序次数。最大 0xFF |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 10 00 87 04 01 01 02 02 03 03 04 04 0A 00 00 01 A6

Data 解析：04：轮询天线数量

01：第 1 个轮询的天线号

01：天线重复轮询次数

02：第 2 个轮询的天线号

02：天线重复轮询次数

03：第 3 个轮询的天线号

03：天线重复轮询次数

04：第一个轮询的天线号

04：第一个轮询的天线号

0A：天线间隔时间 10ms

00：session s0

00：flag A

01：重复次数

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x87 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 87 10 C5

Data 解析：10：操作成功

如有标签应答，返回如下数据包（多条）：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | | |
|-----|----------|------|-----|------|-------------------|
| 2 | Len | | 1 | | |
| 3 | Address | | 1 | | |
| 4 | Cmd | | 1 | 0x87 | |
| 5.1 | Data | Ant | 1 | | 当前工作的天线号 |
| 5.2 | | PC | 2 | | 标签的 PC ,固定两个字节。 |
| 5.3 | | EPC | 不定长 | | 标签的 EPC 号, 长度可变化。 |
| 5.4 | | RSSI | 4 | | 标签的实时 RSSI |
| 5.5 | | Freq | 3 | | 读取标签的频点参数 |
| 6 | Checksum | | 1 | | |

参考示例:

串口数据: A0 19 00 87 01 30 00 E2 80 69 95 00 00 40 02 D7 43 70 AA E8 2B 1C 11 0D F6 38 3E

Data 解析: 01: 当前工作的天线号

30 00: PC

E2 80 69 95 00 00 40 02 D7 43 70 AA: EPC

E8 2B 1C 11: rssi

0D F6 38: 读取标签的频点

01: 重复次数

盘存自动结束:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x87 | |
| 5 | Data | 1 | | 0x13:盘存轮数到达, 自动结束 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 87 13 C2

Data 解析: 13: 盘存到达轮数

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x87 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x22:未连接天线 0x36:无可操作标签 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 87 22 B4

Data 解析: 22: 未连接天线

3.45. 89H——盘存标签(实时上传标签数据)

命令：cmd_real_time_inventory

实时模式（Auto）：读写器收到此命令后，进行多标签识别操作。标签数据实时上传，不存入读写器缓存区。此命令一轮盘存耗时较长，适用于大批量标签读取。**在收到停止命令前，读写器将持续盘存。**

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x89 | |
| 5 | Data | 1 | | 单天线产品：天线号固定为 0x01 多天线产品： 单天线模式：天线号 (0x01~0x04) 多天线模式：天线号固定 0x00 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 89 01 D2

Data 解析：01：1号天线

读写器返回数据包

如有标签应答，返回如下数据包（多条）：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|------|------|----------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x89 | |
| 5.1 | Data | Ant | 1 | 天线号 |
| 5.2 | | PC | 2 | 标签的 PC |
| 5.3 | | EPC | 不定长 | 标签的 EPC 号，长度可变 |
| 5.4 | | RSSI | 4 | 标签的实时 RSSI |
| 5.5 | | Freq | 3 | 读回标签的频率 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 19 00 89 01 34 00 E2 80 F3 36 20 00 F0 00 0B B7 7B 8D E8 14 30 47 0D F6 38 76

Data 解析：01：当前工作的天线号

34 00：PC

E2 80 F3 36 20 00 F0 00 0B B7 7B 8D：EPC

E8 14 30 47: rssi

0D F6 38: 读取标签的频点

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x89 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 0x22:未连接天线 0x36:无可操作标签 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 89 10 C3

Data 解析: 10: 操作成功

3.46. 8AH——盘存标签(自定义读取数据)

命令: cmd_custom_inventory

自定义读取模式(custom): 读写器收到此命令后, 进行多标签识别并进行自定义数据读取操作。标签数据实时上传, 不存入读写器缓存区。此命令可自定义盘存轮数盘, 并且可自定义读取标签内容。

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 7 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8A | |
| 5 | 盘存天线 | 1 | | 单天线产品: 天线号固定为 0x01 多天线产品: 单天线模式: 天线号(0x01~0x04) 多天线模式: 天线号固定 0x00 |
| 6 | 盘存次数 | 1 | | 选择读写器需要盘存的轮数,0xFF 表示一直盘存 |
| 7 | 匹配参数 | 1 | | 选择读写器需要匹配的 membank 0x00:不匹配 0x01:匹配 TID 0x02:匹配 EPC |

| | | | | |
|---|----------|---|--|--|
| | | | | 0x03:匹配 USERDATA 0x04:读 Magnus 温度标签 0x05:国标匹配标签信息区 0x06:国标匹配编码区 0x0a:匹配凯路威标签 |
| 8 | 匹配长度 | 1 | | 表示匹配 membank 的数据长度 |
| 9 | Checksum | 1 | | |

参考示例:

串口数据: A0 07 00 8A 00 01 02 06 C6

Data 解析: 00: 多天线产品多天线模式

01: 盘存次数

02: 匹配 EPC

06: 匹配长度

读写器返回数据包

如有标签应答, 返回如下数据包 (多条):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|--------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | | 根据选择的匹配参数返回命名头: 不匹配: 0x8A 匹配 TID: 0x53 匹配 EPC: 0x54 匹配 USERDATA: 0x55 读 Magnus 温度标签: 0x52 |
| 5.1 | Data | Ant | 1 | 当前盘存的天线号 |
| | | PC | 2 | 标签的 PC |
| | | EPC | | 标签的 EPC 号, 长度可变 |
| | | CRC | 2 | 标签的 CRC |
| | | 匹配数据长度 | 1 | 根据选择的匹配参数返回数据长度: 不匹配: 0 匹配 TID: epc_len 匹配 EPC: epc_len 匹配 USERDATA: 4 读 Magnus 温度标签: 4 |
| 5.2 | Data | 不定长 | | 根据匹配数据长度返回数据 |
| 5.3 | RSSI | 4 | | 标签的实时 RSSI |
| 5.4 | Freq | 3 | | 读回标签的频率 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 28 00 54 01 34 00 E2 80 F3 36 20 00 F0 00 0B B7 31 D9 A1 AD 0C E2 80 F3 36 20 00

F0 00 0B B7 31 D9 E8 09 AA 85 0D EE 68 04

Data 解析: 00: 地址

54: 匹配 EPC

01: 当前盘存的天线号

34 00: PC

E2 80 F3 36 20 00 F0 00 0B B7 31 D9: 标签的 EPC

A1 AD: CRC

0C : 匹配数据长度 12

E2 80 F3 36 20 00 F0 00 0B B7 31 D9: 匹配数据

E8 09 AA 85: rssi

0D EE 68: 读取标签的频点

盘存自动结束:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | | 根据读写器选择 membank 返回命名头: 不匹配: 0x8A 匹配 TID: 0x53 匹配 EPC: 0x54 匹配 USERDATA:0x55 读 Magnus 温度标签: 0x52 |
| 5 | Data | 1 | | 0x12:设置轮数到达, 自动退出盘存 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 54 12 F6

Data 解析: 54: 匹配 EPC

12: 设置轮数到达, 自动退出盘存

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|---------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8A | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x22:未连接天线 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 |

| | | | | |
|---|----------|---|--|--|
| 6 | Checksum | 1 | | |
|---|----------|---|--|--|

参考示例:

串口数据: A0 04 00 8A 10 C2

Data 解析: 10: 操作成功

3.47. 8BH——自定义 session 和 target 盘存

命令: cmd_customized_session_target_inventory

推荐使用的盘存指令

实时模式 (Session): 读写器收到此命令后, 按照指定的 session 和 inventoried flag 进行多标签识别操作。标签数据实时上传, 不存入读写器缓存区。普通盘存推荐使用此命令 S1 模式。在收到停止命令前, 读写器将持续盘存。

关于 S0~S1 模式, 详见: EPC RFID Protocols_Class1_Gen2_V1.1.0->6.3.2.2 Sessions and inventoried flags

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|-----------------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8b | |
| 5.1 | Data Session | 1 | | 指定盘存的 session 0 S0 1 S1 2 S2 3 S3 |
| 5.2 | Data Target | 1 | | 指定盘存的 Inventoried Flag 0 A 1 B |
| 5.3 | Data Antid | 1 | | 盘存天线 |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x06
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8B
 数据包 Data: 0x020001 //指定盘存的 session S2,指定盘存的 Inv flag A, 天线 1
 校验码 Checksum: 0xCE

读写器返回数据包:

如有标签应答, 返回如下数据包 (多条):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |

| | | | | |
|-----|----------|------|------|------------|
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x89 | |
| 5.1 | Data | Ant | 1 | 天线号 |
| 5.2 | | PC | 2 | 标签的 PC |
| 5.3 | | EPC | 不定长 | 标签的 EPC 号 |
| 5.4 | | RSSI | 4 | 标签的实时 RSSI |
| 5.5 | | Freq | 3 | 读回标签的频率 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x19
 读写器地址 Address: 0x00
 命令码 Cmd: 0x89
 数据包 Data: 0x01 3000 E280689400005016A9874C56 E60AD18D 0DBBA0
 //01 天线 1
 //3000 标签的 PC
 // E200000000004016A9875056 标签的 EPC 号
 //E621609A 标签的实时 RSSI
 //0DBBA0 标签的频点参数

校验码 Checksum: 0xD2

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x89 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x22:未连接天线 0x32:读标签错误 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x43:wordcnt 参数超过规定 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x04
 读写器地址 Address: 0x00
 命令码 Cmd: 0x89
 数据包 Data: 0x36

校验码 Checksum: 0x9D

3.48. 8CH——停止盘存

命令: cmd_stop_inventory

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8C | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 8C D1

读写器成功执行不返回命令

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8C | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 8C 11 D2

Data 解析: 11: 操作失败

3.49. 8DH——选择标签

命令: cmd_set_select

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|----------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8D | |
| 5.1 | Select | Enable | 1 | 0:不使能 select; 1:使能 select |
| 5.2 | | SelParam | 1 | [7:5]Target,[4:2]Action,[1:0]MemBank 即由 |

| | | | | |
|-----|----------|----------|-----|--|
| | | | | target(3bits)+Action(3bits)+MemBank(2bits)组合而成, 例如 target 为 S2(010),Action 是 000,MemBank 是 EPC(01)组合 0100 0001 即 41(Hex) |
| 5.3 | | Pointer | 4 | 掩码开始地址 bit |
| 5.4 | | MaskLen | 1 | 掩码长度 |
| 5.5 | | Truncate | 1 | 0:不使能 Truncate; 1:使能 Truncate |
| 5.6 | | Mask | 不定长 | 掩码 |
| 5 | Checksum | | 1 | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x17
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8D
 数据包 Data: 0x01 41 00000020 60 01 E200000000004016A9875056

//01 使能;
 //41 SelParam;
 //00000020 掩码开始地址
 //60 掩码长度
 //01 使能 Truncate
 //E200000000004016A9875056 EPC 码

校验码 Checksum: 0xEB

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8D | |
| 5 | Data | 1 | | 0x10:操作成功 |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x04
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8D
 数据包 Data: 0x10
 校验码 Checksum: 0xBF

3.50. 8EH——获取标签

命令: cmd_get_select

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8E | |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x03
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8E
 校验码 Checksum: 0xCF

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x8E | |
| 5.1 | Select | Enable | 1 | 0:不使能 select; 1:使能 select |
| 5.2 | | SelParam | 1 | [7:5]Target,[4:2]Action,[1:0]MemBank 即由 target(3bits)+Action(3bits)+MemBank(2bits) 组合而成, 例如 target 为 S2(010),Action 是 000,MemBank 是 EPC(01) 组合 0100 0001 即 41(Hex) |
| 5.3 | | Pointer | 4 | 掩码开始地址 bit |
| 5.4 | | MaskLen | 1 | 掩码长度 |
| 5.5 | | Truncate | 1 | 0:不使能 Truncate; 1:使能 Truncate |
| 5.6 | | Mask | | 不定长 |
| 5 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x17
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8E
 数据包 Data: 0x01 41 00000020 60 01 E200000000004016A9875056
 //01 使能;
 //41 SelParam;
 //00000020 掩码开始地址

//60 掩码长度
 //01 使能 Truncate
 //E200000000004016A9875056 EPC 码

校验码 Checksum: 0xEA

3.51. 90H——查询盘存缓存

命令: cmd_get_inventory_buffer

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x90 | |
| 5 | Checksum | 1 | | |

参考示例:

串口数据: A0 03 00 90 CD

读写器返回数据包:

成功返回: 此数据可能返回多条。数量等于缓存中的标签数量(无重复数据)。

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|------------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x90 | |
| 5.1 | Data | InvDataLen | 1 | 所操作标签的有效数据长度。 (PC+CRC+EPC)。单位是字节。 如果是国标的标签只返回(PC+EPC) |
| 5.2 | | InvData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格) + CRC (2 字节) 即 EPC 存储区域中的全部内容。 如果是国标的标签只返回(PC(2 字节) + EPC) |
| 5.3 | | RSSI | 4 | 第一次读到该标签时的信号强度。 |
| 5.4 | | Freq | 3 | 第一次读到该标签时的频率。 |
| 5.5 | | Ant | 1 | 第一次读取的天线号。 |
| 5.6 | | InvCount | 1 | 该标签成功读取的次数, 如果该值为 0xFF, 则说明成功读取次数 >= 255 次。 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 1D 00 90 10 30 00 E2 80 69 95 00 00 40 05 CF 27 A4 63 CE 11 E8 19 22 0A 0E 25 18 01 01 78

Data 解析: 10: 标签的有效数据长度。(PC+EPC+CRC)

30 00: PC

E2 80 69 95 00 00 40 05 CF 27 A4 63: 标签的 EPC

CE 11: CRC

E8 19 22 0A: rssi

0E 25 18: 读取标签的频点

01: 第一次读取的天线号

01: 成功读取的次数

注意:

★命令完成后, 缓存中的数据并不丢失, 可以多次提取。

★若再次运行 cmd_inventory 命令, 则盘存到的标签将累计存入缓存。

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|---------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x90 | |
| 5 | Data | 1 | 错误码 | 0x38:无 epc 数据 |
| 6 | Checksum | 1 | | |

参考示例:

串口数据: A0 04 00 90 38 94

Data 解析: 38: 无 epc 数据

3.52. 91H——查询并重置盘存缓存

命令: cmd_get_and_reset_inventory_buffer

数据格式请参考[查询盘存缓存](#)命令。

命令成功完成后, 缓存中的数据将被全部清空。

3.53. 92H——查询盘存缓存标签数量

命令: cmd_get_inventory_buffer_tag_count

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |

| | | | | |
|---|----------|---|------|--|
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x92 | |
| 5 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x03
 读写器地址 Address: 0x00
 命令码 Cmd: 0x92
 校验码 Checksum: 0xCB

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 5 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x92 | |
| 5 | Data | 2 | | 缓存中标签数据数量。无重复数据 |
| 6 | Checksum | 1 | | |

参考示例（仅供参考）：

数据帧头 Head: 0xA0
 数据包长 Len: 0x05
 读写器地址 Address: 0x00
 命令码 Cmd: 0x92
 数据包 Data: 0x0002
 校验码 Checksum: 0xC7

3.54. 93H——重置盘存缓存

命令：cmd_reset_inventory_buffer

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x93 | |
| 5 | Checksum | 1 | | |

参考示例：

串口数据：A0 03 00 93 CA

Data 解析：无

读写器返回数据包：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x93 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

参考示例：

串口数据：A0 04 00 93 10 B9

Data 解析：10：操作成功

3.55. 95H——SM7 加密写(可选项)

命令：cmd_sm7_write

上位机发送指令数据：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-------------------|---------------|---------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x95 | |
| 5.1 | Data | Password | 4 | 标签访问密码 |
| 5.2 | | WordCnt 写入数据长度 | 2 | WORD(16 bits)长度，数值请参考标签规格 |
| 5.3 | | WordData | WordCnt *2 | 写入的数据（可变长），是写入数据长度的 2 倍。 |
| 6 | Checksum | 1 | | |

读写器返回数据包

返回如下数据包（多条），数量等于写入的标签数量(无重复数据)：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|---------|-----------|------|--|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x95 | |
| 5.1 | Data | TagCount | 2 | 成功操作的标签总数。16 bits。 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。（PC+CRC+EPC）。单位是字节。 |
| 5.3 | | WriteData | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC(根据标签规格) + |

| | | | | |
|-----|----------|------------|---|--|
| | | | | CRC (2 字节) (PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果，即错误代码。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数，低 2 位是天线号。 |
| 5.6 | | WriteCount | 1 | 写入标签的数据长度 |
| 6 | Checksum | | 1 | |

失败返回：

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x95 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 |
| 6 | Checksum | 1 | | |

3.56. 96H——SM7 解密读(可选项)

命令: cmd_sm7_read

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x04 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x96 | |
| 5. | Data | 1 | | 读取的数据长度。单位是字节。 |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
 数据包长 Len: 0x04
 读写器地址 Address: 0x00
 命令码 Cmd: 0x8C
 数据包 Data: 0x11
 校验码 Checksum: 0xD2

读写器返回数据包

返回如下数据包 (多条)，数量等于读取的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----|-----|-----|----|
|----|----|-----|-----|----|

| | | | | |
|-----|----------|-----------|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x96 | |
| 5.1 | Data | TagCount | 2 | 成功操作的标签总数。16 bits。 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。 (PC+CRC+EPC+读取的标签数据)。单位是字节。 |
| 5.3 | | ReadData | 不定长 | 所操作标签的有效数据。 PC (2 字节) + EPC (根据标签规格) + CRC (2 字节) + 读取的数据。 (PC(2 字节) + EPC + CRC(2 字节) 即 EPC 存储区域中的全部内容。) |
| 5.4 | | ReadLen | 2 | Read 操作的数据长度。单位是字节。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数, 低 2 位是天线号。 |
| 5.6 | | ReadCount | 1 | 该标签被成功操作的次数。 |
| 6 | Checksum | 1 | | |

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|-----------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x96 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 |
| 6 | Checksum | 1 | | |

3.57. 97H——SM7 PK 秘钥更新(可选项)

命令: cmd_sm7_pk_update

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|-----|------|-----------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x13 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x97 | |
| 5.1 | Data | 16 | | SM7 加密密钥 128bit |
| 6 | Checksum | 1 | | |

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x97 | |
| 5 | Data | 1 | 错误码 | 0x10:操作成功 0x11:操作失败 |
| 6 | Checksum | 1 | | |

3.58. 98H——复旦微 GB 双向认证

命令: cmd_gb_mul_SeU_Auth

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|--------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x13 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x98 | |
| 5.1 | Data | Password | 16 | 双向认证密钥 |
| 6 | Checksum | 1 | | |

读写器返回数据包

返回如下数据包（多条），数量等于销毁的标签数量(无重复数据):

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|-----|----------|----------|------|---------------------------------------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x49 | |
| 5.1 | Data | TagCount | 2 | 成功操作的标签总数。16 bits。 |
| 5.2 | | DataLen | 1 | 所操作标签的有效数据长度。（PC +EPC）。单位是字节。 |
| 5.3 | | EPC | 不定长 | 所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格) |
| 5.4 | | ErrCode | 1 | 所操作标签的操作结果，即错误代码。 |
| 5.5 | | AntID | 1 | 高 6 位是第一次读取的频点参数，低 2 位是天线号。 |
| 6 | Checksum | 1 | | |

失败返回:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----|-----|-----|----|
|----|----|-----|-----|----|

| | | | | |
|---|----------|---|------|---|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 4 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0x49 | |
| 5 | Data | 1 | 错误码 | 0x11:操作失败 0x58:双向认证失败 0x59:双向认证成功 0x36:无可操作标签 0x37:成功盘存但访问失败 0x40:访问标签错误或密码错误 0x41:无效的参数 0x60:标签供电不足 0x61:标签权限不足 0x62:内存地址超出范围 0x63:内存被锁定 0x64:操作密码错误 0x65:标签认证读写器失败 0x66:未知错误 |
| 6 | Checksum | 1 | | |

3.59. E1H——读写器温度过高告警

读写器上报数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0xE1 | |
| 5 | Checksum | 1 | | |

3.60. F3H——读取 UUID

命令: cmd_read_uuid

上位机发送指令数据:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|----|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x3 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0xF3 | |
| 5 | Checksum | 1 | | |

读写器返回数据包:

| 序号 | 字段 | 字节数 | 固定值 | 备注 |
|----|----------|-----|------|--------|
| 1 | Head | 1 | 0xA0 | |
| 2 | Len | 1 | 0x13 | |
| 3 | Address | 1 | | |
| 4 | Cmd | 1 | 0xF3 | |
| 5 | Data | 16 | uuid | 128bit |
| 6 | Checksum | 1 | | |

参考示例 (仅供参考):

数据帧头 Head: 0xA0
数据包长 Len: 0x03
读写器地址 Address: 0x00
命令码 Cmd: 0xE1
校验码 Checksum: 0x7C

4. 错误码表

| 序号 | 值 | 名称 | 说明 |
|----|------|---|------------------|
| 1 | 0x10 | command_success | 命令成功完成 |
| 2 | 0x11 | command_fail | 命令执行失败 |
| 3 | 0x12 | Custom inventory complete | 自定义盘存结束 |
| 4 | 0x13 | Fast switch antenna inventory complete | 多天线轮询结束 |
| 5 | 0x20 | mcu_reset_error | CPU 复位错误 |
| 6 | 0x21 | cw_on_error | 打开 CW 错误 |
| 7 | 0x22 | antenna_missing_error | 天线未连接 |
| 8 | 0x23 | write_flash_error | 写 Flash 错误 |
| 9 | 0x24 | read_flash_error | 读 Flash 错误 |
| 10 | 0x25 | set_output_power_error | 设置发射功率错误 |
| 11 | 0x31 | tag_inventory_error | 盘存标签错误 |
| 12 | 0x32 | tag_read_error | 读标签错误 |
| 13 | 0x33 | tag_write_error | 写标签错误 |
| 14 | 0x34 | tag_lock_error | 锁定标签错误 |
| 15 | 0x35 | tag_kill_error | 灭活标签错误 |
| 16 | 0x36 | no_tag_error | 无可操作标签错误 |
| 17 | 0x37 | inventory_ok_but_access_fail | 成功盘存但访问失败 |
| 18 | 0x40 | access_or_password_error | 访问标签错误或访问密码错误 |
| 19 | 0x41 | parameter_invalid | 无效的参数 |
| 20 | 0x42 | parameter_invalid_wordCnt_too_long | wordCnt 参数超过规定长度 |
| 21 | 0x43 | parameter_invalid_membank_out_of_range | MemBank 参数超出范围 |
| 22 | 0x44 | parameter_invalid_lock_region_out_of_range | Lock 数据区参数超出范围 |
| 23 | 0x45 | parameter_invalid_lock_action_out_of_range | LockType 参数超出范围 |
| 24 | 0x46 | parameter_reader_address_invalid | 读写器地址无效 |
| 25 | 0x47 | parameter_invalid_antenna_id_out_of_range | Antenna_id 超出范围 |
| 26 | 0x48 | parameter_invalid_output_power_out_of_range | 输出功率参数超出范围 |
| 27 | 0x49 | parameter_invalid_frequency_region_out_of_range | 射频规范区域参数超出范围 |
| 28 | 0x4A | parameter_invalid_baudrate_out_of_range | 波特率参数超出范围 |
| 29 | 0x4C | parameter_epc_match_len_too_long | EPC 匹配长度越界 |

| | | | |
|----|------|--------------------------------------|--------------|
| 30 | 0x4D | parameter_epc_match_len_error | EPC 匹配长度错误 |
| 31 | 0x4E | parameter_invalid_epc_match_mode | EPC 匹配参数超出范围 |
| 32 | 0x4F | parameter_invalid_frequency_range | 频率范围设置参数错误 |
| 33 | 0x50 | fail_to_get_RN16_from_tag | 无法接收标签的 RN16 |
| 34 | 0x53 | rf_chip_fail_to_response | 射频芯片无响应 |
| 35 | 0x54 | fail_to_achieve_desired_output_power | 输出达不到指定的输出功率 |
| 36 | 0x55 | copyright_authentication_fail | 版权认证未通过 |
| 37 | 0x56 | spectrum_regulation_error | 频谱规范设置错误 |
| 38 | 0x57 | output_power_too_low | 输出功率过低 |
| | 0x58 | GB sm7 double identify failed | 国标双向认证失败 |
| | 0x59 | GB sm7 double identify success | 国标双向认证成功 |
| | 0x60 | GB tag be short of power | 标签功率不足 |
| | 0x61 | GB tag permission error | 权限不足 |
| | 0x62 | GB tag memory over limit | 内存区超限 |
| | 0x63 | GB tag memory has be locked | 内存被锁定 |
| | 0x64 | GB tag password error | 密码错误 |
| | 0x65 | GB identify error | 认证失败 |
| | 0x66 | GB unknown error | 未知错误 |

5. 频率参数对应表

| 频率参数 | 对应频点 | 频率参数 | 对应频点 |
|-----------|------------|-----------|------------|
| 0 (0x00) | 865.00 MHz | 30 (0x1E) | 913.50 MHz |
| 1 (0x01) | 865.50 MHz | 31 (0x1F) | 914.00 MHz |
| 2 (0x02) | 866.00 MHz | 32 (0x20) | 914.50 MHz |
| 3 (0x03) | 866.50 MHz | 33 (0x21) | 915.00 MHz |
| 4 (0x04) | 867.00 MHz | 34 (0x22) | 915.50 MHz |
| 5 (0x05) | 867.50 MHz | 35 (0x23) | 916.00 MHz |
| 6 (0x06) | 868.00 MHz | 36 (0x24) | 916.50 MHz |
| 7 (0x07) | 902.00 MHz | 37 (0x25) | 917.00 MHz |
| 8 (0x08) | 902.50 MHz | 38 (0x26) | 917.50 MHz |
| 9 (0x09) | 903.00 MHz | 39 (0x27) | 918.00 MHz |
| 10 (0x0A) | 903.50 MHz | 40 (0x28) | 918.50 MHz |
| 11 (0x0B) | 904.00 MHz | 41 (0x29) | 919.00 MHz |
| 12 (0x0C) | 904.50 MHz | 42 (0x2A) | 919.50 MHz |
| 13 (0x0D) | 905.00 MHz | 43 (0x2B) | 920.00 MHz |
| 14 (0x0E) | 905.50 MHz | 44 (0x2C) | 920.50 MHz |
| 15 (0x0F) | 906.00 MHz | 45 (0x2D) | 921.00 MHz |
| 16 (0x10) | 906.50 MHz | 46 (0x2E) | 921.50 MHz |
| 17 (0x11) | 907.00 MHz | 47 (0x2F) | 922.00 MHz |
| 18 (0x12) | 907.50 MHz | 48 (0x30) | 922.50 MHz |
| 19 (0x13) | 908.00 MHz | 49 (0x31) | 923.00 MHz |
| 20 (0x14) | 908.50 MHz | 50 (0x32) | 923.50 MHz |
| 21 (0x15) | 909.00 MHz | 51 (0x33) | 924.00 MHz |
| 22 (0x16) | 909.50 MHz | 52 (0x34) | 924.50 MHz |
| 23 (0x17) | 910.00 MHz | 53 (0x35) | 925.00 MHz |
| 24 (0x18) | 910.50 MHz | 54 (0x36) | 925.50 MHz |
| 25 (0x19) | 911.00 MHz | 55 (0x37) | 926.00 MHz |
| 26 (0x1A) | 911.50 MHz | 56 (0x38) | 926.50 MHz |
| 27 (0x1B) | 912.00 MHz | 57 (0x39) | 927.00 MHz |
| 28 (0x1C) | 912.50 MHz | 58 (0x3A) | 927.50 MHz |
| 29 (0x1D) | 913.00 MHz | 59 (0x3B) | 928.00 MHz |

6. RSSI 参数计算方法(C 语言描述)

```
const uint8_t para_B[5][8] = {
    {43,43,45,49,43,43,45,49},
    {43,43,45,49,43,43,45,49},
    {43,43,45,49,43,43,45,49},
    {53,53,48,43,49,45,43,43},
    {47,47,47,47,46,43,43,43}
};

const int para_C[5][8] = {
    {43,43,45,49,43,43,45,49},
    {43,43,45,49,43,43,45,49},
    {43,43,45,49,43,43,45,49},
    {-283,-283,-283,-283,-283,-283,-283,-283},
    {-303,-283,-253,-238,-304,-313,-280,-266}
};

int Calculate_Rssi(char data[],uint8_t epc_len){
    uint8_t rssi_mode = 0, hardware_mode = 0;
    int B=0,C=0,D=0, RssiVal = 0;
    float A = 1.0f, rssi_temp = 0.0f;
    union{
        uint32_t u32;
        uint8_t chr[4];
    }UNION;
    if(epc_len == 0)epc_len = 1;
    rssi_mode      = (data[0]&0xE0) >> 5;
    hardware_mode  = (data[0]&0x1E) >> 1;
    UNION.chr[3] = data[0] & 0x01;
    UNION.chr[2] = data[1];
    UNION.chr[1] = data[2];
    UNION.chr[0] = data[3];
    B = para_B[hardware_mode][rssi_mode];
    C = para_C[hardware_mode][rssi_mode];
    rssi_temp = (UNION.u32/epc_len)*A;
    RssiVal = (B * log10(rssi_temp)) + C + D;
    if(RssiVal > 0)RssiVal = 0;
    else if(RssiVal < -90)RssiVal = -90;
    return RssiVal;
}
```

7. 校验和计算方法(C 语言描述)

```
uint8_t CheckSum(uint8_t *uBuff, uint8_t uBuffLen)
{
    unsigned char i, uSum = 0;
    for (i = 0; i < uBuffLen; i++)
    {
        uSum = uSum + uBuff[i];
    }
    uSum = (~uSum) + 1;
    return uSum;
}
```

8. 出厂默认参数

| 参数名称 | 参数值(单位) | 备注 |
|-------|---|---------------------------------------|
| 波特率 | 115200 (bps) | 指令通信串口的波特率可配，打印日志的串口波特率不可配（115200bps） |
| 读写器地址 | 0x00 | 出厂时默认使用公共地址 |
| 发射功率 | ucm601:20 (dBm) ucm606:30 (dBm) ucm608:33 (dBm) | |
| 工作天线 | 1 | |
| 工作模式 | 跳频 | |
| 工作频段 | FCC | 902MHz-928MHz |
| 频点个数 | 53 | 跳频时使用此参数 |
| 频点间隔 | 500 (KHz) | 跳频时使用此参数 |
| 起始频点 | 902 (MHz) | |
| 通信带宽 | Miller 200KHz 25us | |
| 蜂鸣器 | 1 (打开) | |



注意

本手册公开了 UCM60x 系列串口协议。

免责声明

UCchip 提供技术指导文档和规格参数（包括接口定义、电气性能和结构尺寸等信息），对本手册不做任何明示或暗示的声明或保证，并且不对特定目的适销性及适用性或者任何间接的、特殊的或连带的损失承担任何责任。

本手册不提供任何形式的保证，并可能随时对本手册或手册描述的产品进行修改或改进，恕不另行通知且不承担任何责任。

在适用的法律允许的范围内，UCchip 均不对用户因使用本手册内容和本手册中描述的产品而引起的任何特殊的、间接的、附带的或后果性的损坏、利润损失、数据丢失、声誉和预期的节省而负责。包括因使用本手册中的信息而导致的侵犯任务专有权利的责任

商标声明

 是北京御芯微科技有限公司商标。

本手册和本手册描述的产品中出现的其他商标、产品名称、服务名称和公司名称，均为其各自所有者的财产。

进出口法规

出口、转口或进口本手册中描述的产品（包括但不限于产品软件和技术数据），用户应遵守相关进出口法律和法规。

版权所有

©北京御芯微科技有限公司保留所有权利。未经北京御芯微科技有限公司书面许可，任何单位和个人不得擅自修改、摘抄、复制、翻译、分发、存储本手册内容的部分或全部，并以任何形式传播。