# 40505W-M 通讯协议使用说明

# V2.0

# 版本记录

| Version | Day | Description |
|---|---|---|
| V1.1 | 2013 年 10 月 15 日 | 40505W-M 固件指令使用说明 |
| V1.2 | 2013 年 12 月 19 日 | 添加 IO 控制指令 |
| V2.0 | 2014 年 5 月 20 日 | 修改 Read, Write, Lock, Kill 等指令，不带有 PC 和 EPC 数据段；<br>增加 NXP 标签特有指令如 Change EAS，EAS_Alarm 等 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# 1. 目录

1. 目录 ..... 3

2. 40505W-M 芯片内部 MCU 简介 ..... 6

3. 固件指令简介 ..... 8

3.1. 指令帧格式 ..... 8

3.2. 指令帧类型 ..... 8

4. 固件指令定义 ..... 9

4.1. 获取读写器模块信息 ..... 9

4.2. 单次轮询指令 ..... 10

4.3. 多次轮询指令 ..... 11

4.4. 停止多次轮询指令 ..... 12

4.5. 设置 Select 参数指令 ..... 13

4.6. 设置 Select 模式 ..... 14

4.7. 读标签数据存储区 ..... 15

4.8. 写标签数据存储区 ..... 17

4.9. 锁定 Lock 标签数据存储区 ..... 19

4.10. 灭活 Kill 标签 ..... 22

4.11. 获取 Query 参数 ..... 24

4.12. 设置 Query 参数 ..... 25

4.13. 设置工作地区 ..... 26

4.14. 设置工作信道 ..... 27

4.15. 获取工作信道 ..... 28

4.16. 设置自动跳频 ..... 29

4.17. 获取发射功率 ..... 30

4.18. 设置发射功率 ..... 31

4.19. 设置发射连续载波 ..... 32

4.20. 获取接收解调器参数 ..... 33

4.21. 设置接收解调器参数 ..... 34

4.22. 测试射频输入端阻塞信号 ..... 36

4.23. 测试信道 RSSI ..... 37

4.24. 控制 IO 端口 ..... 38

4.25. NXP ReadProtect/Reset ReadProtect 指令 ..... 40

4.26. NXP Change EAS 指令 ..... 42

4.27. NXP EAS_Alarm 指令 ..... 44

5. 指令总结 ..... 45

6. 命令帧执行失败的响应帧总结 ..... 46

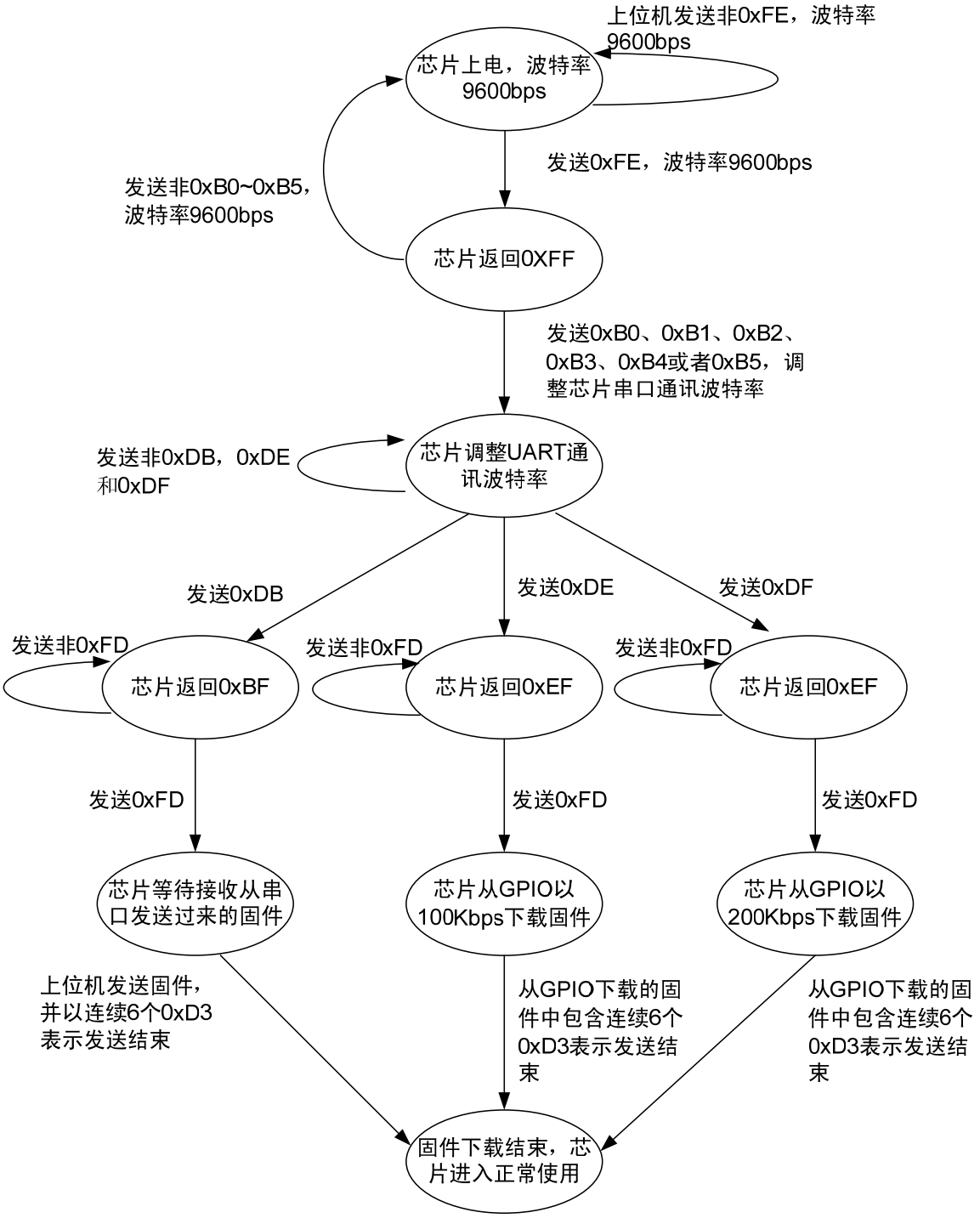
# 2. 40505W-M 内部 MCU 简介

40505W-M 芯片内置有 8 位 8051 MCU，256Byte 内部存储器和 16Kbyte 程序存储器和 3 个定时器(Timer2 用于波特率发生器，Timer0 用于跳频时序控制，Timer1 可以供用户使用)。同时，内置 8Kbyte 的数据 RAM，由 8051MCU 和数字解调电路共用。当正在接收标签返回数据时，该数据 RAM 不能被 MCU 访问。

MCU 固件可以通过 40505W-M 芯片的 UART 串口或者 GPIO(P1.0 和 P1.1)从外部 I2C EEPROM 下载。

UART 串口数据位为 8 位，1 位停止位，无校验位。

40505W-M 芯片通过一系列上电握手协议完成固件下载配置，握手协议如下



波特率设置

| Type | Baud Rate(bps) |
|---|---|
| 0xB0 | 9600 |
| 0xB1 | 19200 |
| 0xB2 | 28800 |
| 0xB3 | 38400 |
| 0xB4 | 57600 |
| 0xB5 | 115200 |

# 3. 固件指令简介

## 3.1. 指令帧格式

固件指令由帧头、帧类型、指令代码、指令数据长度、指令参数、校验码和帧尾组成，均为十六进制表示。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | 07 | 00 | 01 | 01 | 09 | 7E |

帧头 Header: 0xBB
帧类型 Type: 0x00
指令代码 Command: 0x07
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x01
校验位 Checksum: 0x09
帧尾 End: 0x7E

校验位 Checksum 为从帧类型 Type 到最后一个指令参数 Parameter 累加和，并只取累加和最低一个字节(LSB)。

## 3.2. 指令帧类型

| Type | Description |
|---|---|
| 0x00 | 命令帧：由上位机发送给 40505W-M 芯片 |
| 0x01 | 响应帧：由 40505W-M 芯片发回给上位机 |
| 0x02 | 通知帧：由 40505W-M 芯片发回给上位机 |

每一条指令帧都有对应的响应帧。响应帧表示指令是否已经被执行了。

单次轮询指令和多次轮询指令还有相应的通知帧。发送通知帧的个数是由 MCU 根据读取的情况，自主的发给上位机。当读写器读到一个标签就发一个通知帧，而当读写器读到多个标签就发多个通知帧。

# 4. 固件指令定义

## 4.1. 获取读写器模块信息

### 4.1.1. 命令帧定义

获取模块信息如硬件版本、软件版本和制造商信息。

帧类型: 0x00
命令码: 0x03
参数:
硬件版本: 0x00
软件版本: 0x01
制造商: 0x02

例：获取读写器硬件版本

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | 03 | 00 | 01 | 00 | 04 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x03
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00(获取硬件版本)
校验位 Checksum: 0x04

### 4.1.2. 响应帧定义

帧类型: 0x01
指令代码 Command: 0x03
数据: 变量(ASCII 码表示)
例：硬件版本
响应数据 0 为模块信息类型:
硬件版本: 0x00
软件版本: 0x01
制造商: 0x02
之后的数据为模块信息的 ASCII 码。
获取模块硬件版本的响应如下:

| Header | Type | Command | PL(MSB) | PL(LSB) | Info Type | Info | |
|---|---|---|---|---|---|---|---|
| BB | 01 | 03 | 00 | 0B | 00 | 4D (‘M’) | 31 (‘1’) |
| | | | | | | | |
| 30 (‘0’) | 30 (‘0’) | 20 (‘ ‘) | 56 (‘V’) | 31 (‘1’) | 2E (‘.’) | 30 (‘0’) | 30 (‘0’) |
| **Checksum** | **End** | | | | | | |
| 22 | 7E | | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0x03
指令参数长度 PL: 0x000B
模块信息类型 Info Type: 0x00 (硬件版本)
版本信息 Info: 4D 31 30 30 20 56 31 2E 30 30(“M100 V1.00”的 ASCII 码)
校验位 Checksum: 0x22

## 4.2. 单次轮询指令

### 4.2.1. 命令帧定义

完成一次 EPC Class1 Gen2 协议中轮询 Inventory 操作。该指令中不包含 Select 操作。每次轮询指令执行前后都会自动打开和关闭功放。单次轮询 Inventory 指令中，Query 操作参数由另外一条指令来配置，固件中已经有初始值。单次轮询 Inventory 指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | 22 | 00 | 00 | 22 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x22
指令参数长度 PL: 0x0000
校验位 Checksum: 0x22

### 4.2.2. 通知帧定义

芯片接收到单次轮询指令后，如果能够读到 CRC 校验正确的标签，芯片 MCU 将返回包含 RSSI、PC、EPC 和 CRC 的数据。读到一个标签 EPC 就返回一条指令响应，读到多个标签则返回多条指令响应。如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | RSSI | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 02 | 22 | 00 | 11 | C9 | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **CRC(MSB)** | **CRC(LSB)** | **Checksum** | **End** |
| E3 | D5 | 0D | 70 | 3A | 76 | EF | 7E |

帧类型 Type: 0x02
指令代码 Command: 0x22
指令参数长度 PL: 0x0011
RSSI: 0xC9
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
CRC : 0x3A76
校验位 Checksum: 0xEF

RSSI 值反映的是芯片输入端信号大小，不包含天线增益和定向耦合器衰减等。RSSI 为芯片输入端信号强度，十六进制有符号数，单位为 dBm。上面的例子中 RSSI 为 0xC9，代表芯片输入端信号强度为-55dBm。

### 4.2.3. 响应帧定义

如果没有收到标签返回或者返回数据 CRC 校验错误，将返回错误代码 0x15，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 15 | 16 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x01
指令参数 Parameter: 0x15
校验位 Checksum: 0x16

## 4.3. 多次轮询指令

### 4.3.1. 命令帧定义

该指令要求芯片 MCU 进行多次轮询 Inventory 操作，轮询次数限制为 0-65535 次。如果轮询次数为 10000 次，则指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Reserved | CNT(MSB) | CNT(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 00 | 27 | 00 | 03 | 22 | 27 | 10 |
| **Checksum** | **End** | | | | | | |
| 83 | 7E | | | | | | |

帧类型 Type: 0x00
指令代码 Command: 0x27
指令参数长度 PL: 0x0003
保留位 Reserved: 0x22
轮询次数 CNT: 0x2710
校验位 Checksum: 0x83

### 4.3.2. 通知帧定义

多次轮询 Inventory 指令响应帧与单词轮询 Inventory 响应帧格式一样，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | RSSI | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 02 | 22 | 00 | 11 | C9 | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **CRC(MSB)** | **CRC(LSB)** | **Checksum** | **End** |
| E3 | D5 | 0D | 70 | 3A | 76 | EF | 7E |

帧类型 Type: 0x02
指令代码 Command: 0x27
指令参数长度 PL: 0x0011
RSSI: 0xC9
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
CRC : 0x3A76
校验位 Checksum: 0xEF

### 4.3.3. 响应帧定义

如果没有收到标签返回或者返回数据 CRC 校验错误，将返回错误代码 0x15，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 15 | 16 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x01
指令参数 Parameter: 0x15
校验位 Checksum: 0x16

## 4.4. 停止多次轮询指令

### 4.4.1. 命令帧定义

在芯片内部 MCU 进行多次轮询 Inventory 操作的过程中，可以立即停止多次轮询操作，非暂停多次轮询操作，指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | 28 | 00 | 00 | 28 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x28
指令参数长度 PL: 0x0000
校验位 Checksum: 0x28

### 4.4.2. 响应帧定义

如果停止多次轮询指令成功执行，固件则返回响应如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | 28 | 00 | 01 | 00 | 2A | 7E |

帧类型 Type: 0x01
指令代码 Command: 0x28
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0x2A

## 4.5. 设置 Select 参数指令

### 4.5.1. 命令帧定义

设置 Select 参数，并且同时设置 Select 模式为 0x02。在对标签除轮询操作之前，先发送 Select 指令。在多标签的情况下，可以根据 Select 参数只对特定标签进行轮询和读写等操作。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | SelParam | Ptr(MSB) | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 0C | 00 | 13 | 01 | 00 | 00 |
| | Ptr(LSB) | MaskLen | Truncate | Mask(MSB) | | | |
| 00 | 20 | 60 | 00 | 30 | 75 | 1F | EB |
| | | | | | | | Mask(LSB) |
| 70 | 5C | 59 | 04 | E3 | D5 | 0D | 70 |
| Checksum | End | | | | | | |
| AD | 7E | | | | | | |

帧类型 Type: 0x00
指令代码 Command: 0x0C
指令参数长度 PL 0x0013
SelParam: 0x01 (Target: 3’b000, Action: 3’b000, MemBank: 2’b01)
Ptr: 0x00000020(以 bit 为单位，非 word) 从 EPC 存储位开始
Mask 长度 MaskLen: 0x60(6 个 word，96bits)
是否 Truncate: 0x00(0x00 是 Disable truncation，0x80 是 Enable truncation)
Mask: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0xAD

SelParam 共 1 个 Byte，其中 Target 占最高 3 个 bits，Action 占中间 3 个 bits，MemBank 占最后 2 个 bits。MemBank 含义如下：

2’b00: 标签 RFU 数据存储区
2’b00: 标签 EPC 数据存储区
2’b00: 标签 TID 数据存储区
2’b00: 标签 User 数据存储区

Target 和 Action 详细含义请参见 EPC Gen2 协议。
当 Select Mask 长度大于 80 bits(5 words)，发送 Select 指令会先把场区内所有标签设置成 Inventoried Flag 为 A，SL Flag 为~SL 的状态，然后再根据所选的 Action 进行操作。当 Select Mask 长度小于 80 bits(5 words)的时候，不会预先将标签状态通过 Select 指令设置成 Inventoried Flag 为 A，SL Flag 为~SL 的状态。

### 4.5.2. 响应帧定义

当成功设置了 Select 参数后，固件返回如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Data | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | 0C | 00 | 01 | 00 | 0E | 7E |

帧类型 Type: 0x01
指令代码 Command: 0x0C
指令参数长度 PL: 0x0001
返回数据 Data: 0x00
校验位 Checksum: 0x0E

## 4.6. 设置 Select 模式

### 4.6.1. 命令帧定义

如果已经设置好了 Select 参数，执行该条指令，可以设置 Select 模式。例如，如果要取消 Select 指令：

| Header | Type | Command | PL(MSB) | PL(LSB) | Mode | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | 12 | 00 | 01 | 01 | 14 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x12
指令参数长度 PL: 0x0001
指令参数，Select 模式: 0x01
校验位 Checksum: 0x14

Select 模式 Mode 含义：

0x00: 在对标签的所有操作之前都预先发送 Select 指令选取特定的标签。
0x01: 在对标签操作之前不发送 Select 指令。
0x02: 仅对除轮询 Inventory 之外的标签操作之前发送 Select 指令，如在 Read，Write，Lock，Kill 之前先通过 Select 选取特定的标签。

### 4.6.2. 响应帧定义

当成功设置了取消或者发送 Select 指令后，固件返回如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Data | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | 0C | 00 | 01 | 00 | 0E | 7E |

帧类型 Type: 0x01
指令代码 Command: 0x0C
指令参数长度 PL: 0x0001
返回数据 Data: 0x00(执行成功)
校验位 Checksum: 0x0E

## 4.7. 读标签数据存储区

### 4.7.1. 命令帧定义

对单个标签，读取标签数据存储区 Memory Bank 中指定地址和长度的数据。读标签数据区地址偏移 SA 和读取标签数据存储区长度 DL，他们的单位为 Word，即 2 个 Byte/16 个 Bit。这条指令之前应先设置 Select 参数，以便选择指定的标签进行写标签数据区操作。如果 Access Password 全为零，则不发送 Access 指令。

| Header | Type | Command | PL(MSB) | PL(LSB) | AP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 39 | 00 | 09 | 00 | 00 | FF |
| **AP(LSB)** | **MemBank** | **SA(MSB)** | **SA(LSB)** | **DL(MSB)** | **DL(LSB)** | **Checksum** | **End** |
| FF | 03 | 00 | 00 | 00 | 02 | 45 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x39
指令参数长度 PL 0x0009
Access Password: 0x0000FFFF
标签数据存储区 MemBank: 0x03(User 区)
读标签数据区地址偏移 SA: 0x0000
读标签数据区地址长度 DL: 0x0002
校验位 Checksum: 0x45

### 4.7.2. 响应帧定义

读到指定标签存储区数据后，并且 CRC 校验正确，会返回如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | 39 | 00 | 13 | 0E | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Data(MSB)** | | | **Data(LSB)** |
| E3 | D5 | 0D | 70 | 12 | 34 | 56 | 78 |
| **Checksum** | **End** | | | | | | |
| B0 | 7E | | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0x39
指令参数长度 PL: 0x0013
操作的标签 PC+EPC 长度 UL: 0x0E
操作的标签 PC: 0x3400
操作的标签 EPC: 0x30751FEB705C5904E3D50D70
返回数据 Data: 0x12345678
校验位 Checksum: 0xB0

如果该标签没有在场区或者指定的 EPC 代码不对，会返回错误代码 0x09，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 09 | 0A | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Error Code: 0x09

校验位 Checksum: 0x0A

如果 Access Password 不正确，则返回错误代码 0x16，并会返回所操作的标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | 16 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 75 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0010
指令参数 Error Code: 0x16
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x75

如果操作标签返回了 EPC Gen2 协议规定的错误代码(error codes)，因为 EPC Gen2 规定的 error codes 只有低 4 位有效，响应帧会将标签返回的错误代码或上 0xA0 之后再返回。
比如如果发送指令参数中地址偏移或者数据长度不正确，读取数据长度超过标签数据存储区长度，按照 EPC Gen2 协议，标签会返回 error code 0x03(存储区超出，Memory Overrun)。响应帧则返回错误代码 0xA3，并返回所操作标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | A3 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 02 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0010
指令参数 Error Code: 0xA3
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x02

## 4.8. 写标签数据存储区

### 4.8.1. 命令帧定义

对单个标签，写入标签数据存储区 Memory Bank 中指定地址和长度的数据。标签数据区地址偏移 SA 和要写入的标签数据长度 DL，他们的单位为 Word，即 2 个 Byte/16 个 Bit。这条指令之前应先设置 Select 参数，以便选择指定的标签进行写标签数据区操作。如果 Access Password 全为零，则不发送 Access 指令。

写入标签数据存储区的数据长度 DT 应不超过 32 个 word，即 64Byte 字节/512Bit 位。

| Header | Type | Command | PL(MSB) | PL(LSB) | AP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 49 | 00 | 0D | 00 | 00 | FF |
| **AP(LSB)** | **MemBank** | **SA(MSB)** | **SA(LSB)** | **DL(MSB)** | **DL(LSB)** | **DT(MSB)** | |
| FF | 03 | 00 | 00 | 00 | 02 | 12 | 34 |
| | **DT(LSB)** | **Checksum** | **End** | | | | |
| 56 | 78 | 6D | 7E | | | | |

帧类型 Type: 0x00
指令代码 Command: 0x39
指令参数长度 PL: 0x000D
Access Password: 0x0000FFFF
标签数据存储区 MemBank: 0x03
标签数据区地址偏移 SA: 0x0000
数据长度 DL: 0x0002
写入数据 DT: 0x12345678
校验位 Checksum: 0x6D

### 4.8.2. 响应帧定义

将数据写入标签数据存储区后，如果读写器芯片接收到标签返回值正确，则响应帧如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | 49 | 00 | 10 | 0E | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | A9 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0x49
指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0xA9

如果该标签没有在场区或者指定的 EPC 代码不对，会返回错误代码 0x10，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 10 | 0A | 7E |

帧类型 Type: 0x01

指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x10
校验位 Checksum: 0x0A

如果 Access Password 不正确，则返回错误代码 0x16，并会返回所操作的标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | 16 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 75 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0016
指令参数 Error Code: 0x16
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x75

如果操作标签返回了 EPC Gen2 协议规定的错误代码(error codes)，响应帧会将标签返回的错误代码或上 0xB0 之后再返回。
比如如果发送指令参数中地址偏移或者数据长度不正确，写入数据长度超过标签数据存储区长度，按照 EPC Gen2 协议，标签会返回 error code 0x03(存储区超出，Memory Overrun)。则响应帧返回错误代码 0xB3，并返回所操作标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | B3 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 12 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0010
指令参数 Error Code: 0xB3
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x12

## 4. 9. 锁定 Lock 标签数据存储区

### 4. 9. 1. 命令帧定义

对单个标签，锁定 Lock 或者解锁 Unlock 该标签的数据存储区。这条指令之前应先设置 Select 参数，以便选择指定的标签进行锁定 Lock 操作。例如要锁定 Access Password，则指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | AP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 82 | 00 | 07 | 00 | 00 | FF |
| **AP(LSB)** | **LD(MSB)** | | **LD(LSB)** | **Checksum** | **End** | | |
| FF | 02 | 00 | 80 | 09 | 7E | | |

帧类型 Type: 0x00
指令代码 Command: 0x82
指令参数长度 PL: 0x0007
Access Password: 0x0000FFFF
Lock 操作数 LD: 0x020080
校验位 Checksum: 0x09

Lock 操作参数 LD 的高 4 位是保留位，剩下的 20 为是 Lock 操作 Payload，包括 Mask 和 Action，从高到低依次各 10 位。详细含义请参见 EPC Gen2 协议 1.2.0 版 6.3.2.11.3.5 节。
Mask 是一个掩膜，只有 Mask 位为 1 的 Action 才有效。每个数据区的 Action 有 2 bits，00~11，依次对应为开放，永久开放，锁定，永久锁定。
比如 Kill Mask 为 2bits 00，则不管 Kill Action 是什么，Kill Action 都不会生效。当 Kill Mask 为 2bits 10，Kill Action 为 2bits 10，代表 Kill Password 被 Lock(非 Perma Lock)住了，只有通过有效的 Access Password 才能被读写。
Mask 和 Action 每一位的含义如下表表示。

**Lock-Command Payload**

| 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Kill Mask | | Access Mask | | EPC Mask | | TID Mask | | User Mask | | Kill Action | | Access Action | | EPC Action | | TID Action | | User Action | |

**Masks and Associated Action Fields**

| | Kill pwd | | Access pwd | | EPC memory | | TID memory | | User memory | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| *Mask* | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write |
| | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| *Action* | pwd read/ write | perma lock | pwd read/ write | perma lock | pwd write | perma lock | pwd write | perma lock | pwd write | perma lock |

| pwd-write | permalock | Description |
|---|---|---|
| 0 | 0 | Associated memory bank is writeable from either the **open** or **secured** states. |
| 0 | 1 | Associated memory bank is permanently writeable from either the **open** or **secured** states and may never be locked. |
| 1 | 0 | Associated memory bank is writeable from the **secured** state but not from the **open** state. |
| 1 | 1 | Associated memory bank is not writeable from any state. |
| **pwd-read/write** | **permalock** | **Description** |
| 0 | 0 | Associated password location is readable and writeable from either the **open** or **secured** states. |
| 0 | 1 | Associated password location is permanently readable and writeable from either the **open** or **secured** states and may never be locked. |
| 1 | 0 | Associated password location is readable and writeable from the **secured** state but not from the **open** state. |
| 1 | 1 | Associated password location is not readable or writeable from any state. |

### 4.9.2. 响应帧定义

如果 Lock 指令执行正确，标签的返回有效，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | 82 | 00 | 10 | 0E | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | E2 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0x82
指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0xE2

如果该标签没有在场区或者指定的 EPC 代码不对，会返回错误代码 0x13，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 13 | 14 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x13
校验位 Checksum: 0x14

如果 Access Password 不正确，则返回错误代码 0x16，并会返回所操作的标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | 16 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 75 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0016
指令参数 Error Code: 0x16
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x75

如果操作标签返回了 EPC Gen2 协议规定的错误代码(error codes)，响应帧会将标签返回的错误代码或上 0xC0 之后再返回。
比如如果标签 TID 区已经被永久锁定了，然后通过 Lock 指令设置 TID 区为开放状态，按照 EPC Gen2 协议，标签会返回 error code 0x04(存储区锁定，Memory Locked)。则响应帧返回错误代码 0xC4，并返回所操作标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | C4 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 23 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0010
指令参数 Error Code: 0xC4
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x23

## 4.10. 灭活 Kill 标签

### 4.10.1. 命令帧定义

这条指令之前应先设置 Select 参数，以便选择指定的标签进行灭活 Kill 操作。对单标签的灭活操作。

| Header | Type | Command | PL(MSB) | PL(LSB) | KP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 65 | 00 | 04 | 00 | 00 | FF |
| **KP(LSB)** | **Checksum** | **End** | | | | | |
| FF | 67 | 7E | | | | | |

帧类型 Type: 0x00
指令代码 Command: 0x65
指令参数长度 PL: 0x0012
Kill Password: 0x0000FFFF
校验位 Checksum: 0x67

### 4.10.2. 响应帧定义

如果 Kill 指令执行正确，标签的返回 CRC 正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | 65 | 00 | 10 | 0E | 34 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | C5 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0x65
指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0xC5

如果该标签没有在场区或者指定的 EPC 代码不对，会返回错误代码 0x12，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 12 | 13 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x12
校验位 Checksum: 0x13

如果操作标签返回了 EPC Gen2 协议规定的错误代码(error codes)，响应帧会将标签返回的错误代码或上 0xD0 之后再返回。
注意：标签如果没有设置过 Kill Password 密码，即 Kill Password 密码全为 0，按照 Gen2 协议，标签不会被 Kill。这时返回错误代码 0xD0，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | D0 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 2F | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0010
指令参数 Error Code: 0xD0
PC+EPC 长度 UL: 0x0E
PC: 0x3400
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x2F

## 4.11. 获取 Query 参数

### 4.11.1. 命令帧定义

获取固件中 Query 命令相关参数。指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End | |
|---|---|---|---|---|---|---|---|
| BB | 00 | 0D | 00 | 00 | 0D | 7E | |

帧类型 Type: 0x00
指令代码 Command: 0x0D
指令参数长度 PL: 0x0000
校验位 Checksum: 0x0D

### 4.11.2. 响应帧定义

如果设置 Query 参数指令执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Para(MSB) | Para(LSB) | Checksum |
|---|---|---|---|---|---|---|---|
| BB | 01 | 0D | 00 | 02 | 10 | 20 | 40 |
| **End** | | | | | | | |
| 7E | | | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0x0D
指令参数长度 PL: 0x0002
Query Parameter: 0x1020
校验位 Checksum: 0x40

参数为 2 字节，有下面的具体参数按位拼接而成。上述响应帧对应的 Query 参数为：
DR=8, M=1, TRext=Use pilot tone, Sel=00, Session=00, Target=A, Q=4

其中:
DR(1 bit): DR=8(1’b0), DR=64/3(1’b1). 只支持 DR=8 的模式
M(2 bit): M=1(2’b00), M=2(2’b01), M=4(2’b10), M=8(2’b11). 只支持 M=1 的模式
TRext(1 bit): No pilot tone(1’b0), Use pilot tone(1’b1). 只支持 Use pilot tone(1’b1)模式
Sel(2 bit): ALL(2’b00/2’b01), ~SL(2’b10), SL(2’b11)
Session(2 bit): S0(2’b00), S1(2’b01), S2(2’b10), S3(2’b11)
Target(1 bit): A(1’b0), B(1’b1)
Q(4 bit): 4’b0000-4’b1111

.

## 4. 12. 设置 Query 参数

### 4. 12. 1. 命令帧定义

设置 Query 命令中的相关参数。参数为 2 字节，有下面的具体参数按位拼接而成：

DR(1 bit): DR=8(1’b0), DR=64/3(1’b1). 只支持 DR=8 的模式

M(2 bit): M=1(2’b00), M=2(2’b01), M=4(2’b10), M=8(2’b11). 只支持 M=1 的模式

TRext(1 bit): No pilot tone(1’b0), Use pilot tone(1’b1). 只支持 Use pilot tone(1’b1)模式

Sel(2 bit): ALL(2’b00/2’b01), ~SL(2’b10), SL(2’b11)

Session(2 bit): S0(2’b00), S1(2’b01), S2(2’b10), S3(2’b11)

Target(1 bit): A(1’b0), B(1’b1)

Q(4 bit): 4’b0000-4’b1111

如果 DR=8, M=1, TRext=Use pilot tone, Sel=00, Session=00, Target=A, Q=4，则指令如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Para(MSB) | Para(LSB) | Checksum |
|---|---|---|---|---|---|---|---|
| BB | 00 | 0E | 00 | 02 | 10 | 20 | 40 |
| **End** | | | | | | | |
| 7E | | | | | | | |

帧类型 Type: 0x00

指令代码 Command: 0x0E

指令参数长度 PL: 0x0002

Query 参数 Parameter: 0x1020

校验位 Checksum: 0xC6

### 4. 12. 2. 响应帧定义

如果设置 Query 参数指令执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | 0E | 00 | 01 | 00 | 10 | 7E |

帧类型 Type: 0x01

指令代码 Command: 0x0E

指令参数长度 PL: 0x0001

指令参数 Parameter: 0x00

校验位 Checksum: 0x10

# 4.13. 设置工作地区

## 4.13.1. 命令帧定义

设置读写器工作地区，如果是中国 900MHz 频段，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Region | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | 07 | 00 | 01 | 01 | 09 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0x07
指令参数长度 PL: 0x0001
地区 Region: 0x01
校验位 Checksum: 0x09

不同国家地区代码如下表：

| Region | Parameter |
|---|---|
| 中国 900MHz | 01 |
| 中国 800MHz | 04 |
| 美国 | 02 |
| 欧洲 | 03 |
| 韩国 | 06 |

## 4.13.2. 响应帧定义

如果地区设置执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | 07 | 00 | 01 | 00 | 09 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0x07
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0x09

## 4.14. 设置工作信道

### 4.14.1. 命令帧定义

如果是中国 900MHz 频段，设置读写器工作信道 920.125MHz，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | CH Index | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | AB | 00 | 01 | 01 | AC | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xAB
指令参数长度 PL: 0x0001
信道代号 Channel Index: 0x01
校验位 Checksum: 0xAC

中国 900MHz 信道参数计算公式，Freq_CH 为信道频率：
CH_Index = (Freq_CH-920.125M)/0.25M

中国 800MHz 信道参数计算公式，Freq_CH 为信道频率：
CH_Index = (Freq_CH-840.125M)/0.25M

美国信道参数计算公式，Freq_CH 为信道频率：
CH_Index = (Freq_CH-902.25M)/0.5M

欧洲信道参数计算公式，Freq_CH 为信道频率：
CH_Index = (Freq_CH-865.1M)/0.2M

韩国信道参数计算公式，Freq_CH 为信道频率：
CH_Index = (Freq_CH-917.1M)/0.2M

### 4.14.2. 响应帧定义

如果信道设置执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | AB | 00 | 01 | 00 | AD | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xAB
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0xAD

## 4.15. 获取工作信道

### 4.15.1. 命令帧定义

在当前的读写器工作地区，获取读写器工作信道，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | AA | 00 | 00 | AA | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xAA
指令参数长度 PL: 0x0000
校验位 Checksum: 0xAA

### 4.15.2. 响应帧定义

如果获取信道执行正确，则命令帧响应为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | AA | 00 | 01 | 00 | AC | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xAA
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00(Channel_Index 为 0x00)
校验位 Checksum: 0xAC

中国 900MHz 信道参数计算公式，Freq_CH 为信道频率：
Freq_CH = CH_Index * 0.25M + 920.125M

中国 800MHz 信道参数计算公式，Freq_CH 为信道频率：
Freq_CH = CH_Index * 0.25M + 840.125M

美国信道参数计算公式，Freq_CH 为信道频率：
Freq_CH = CH_Index * 0.5M + 902.25M

欧洲信道参数计算公式，Freq_CH 为信道频率：
Freq_CH = CH_Index * 0.2M + 865.1M

韩国信道参数计算公式，Freq_CH 为信道频率：
Freq_CH = CH_Index * 0.2M + 917.1M

## 4.16. 设置自动跳频

### 4.16.1. 命令帧定义

设置为自动跳频模式或者取消自动跳频模式，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | AD | 00 | 01 | FF | AD | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xAD
指令参数长度 PL: 0x0001
指令参数 Parameter: 0xFF(0xFF 为设置自动跳频，0x00 为取消自动跳频)
校验位 Checksum: 0xAD

### 4.16.2. 响应帧定义

如果设置为自动跳频行或者取消自动跳频正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | AD | 00 | 01 | 00 | AF | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xAD
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0xAF

# 4.17. 获取发射功率

## 4.17.1. 命令帧定义

获取当前读写器发射功率，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | B7 | 00 | 00 | B7 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xB7
指令参数长度 PL: 0x0000
校验位 Checksum: 0xB7

## 4.17.2. 响应帧定义

如果获取信道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Pow(MSB) | Pow(LSB) | Checksum |
|---|---|---|---|---|---|---|---|
| BB | 01 | B7 | 00 | 02 | 07 | D0 | 91 |
| **End** | | | | | | | |
| 7E | | | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0xB7
指令参数长度 PL: 0x0002
功率参数 Pow: 0x07D0(当前功率为十进制 2000，即 20dBm)
校验位 Checksum: 0x91

## 4.18. 设置发射功率

### 4.18.1. 命令帧定义

设置当前读写器发射功率，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Pow(MSB) | Pow(LSB) | Checksum |
|---|---|---|---|---|---|---|---|
| BB | 00 | B6 | 00 | 02 | 07 | D0 | 8F |
| **End** | | | | | | | |
| 7E | | | | | | | |

帧类型 Type: 0x00
指令代码 Command: 0xB6
指令参数长度 PL: 0x0002
功率参数 Pow: 0x07D0(当前功率为十进制 2000，即 20dBm)
校验位 Checksum: 0x8F

### 4.18.2. 响应帧定义

如果获取信道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | B6 | 00 | 01 | 00 | B8 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xB6
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0xB8

## 4.19. 设置发射连续载波

### 4.19.1. 命令帧定义

设置发射连续载波或者关闭连续载波，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 00 | B0 | 00 | 01 | FF | B0 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xB0
指令参数长度 PL: 0x0001
指令参数 Parameter: 0xFF(0xFF 为打开连续波，0x00 为关闭连续波)
校验位 Checksum: 0xB0

### 4.19.2. 响应帧定义

如果设置执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | B0 | 00 | 01 | 00 | B2 | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xB0
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0xB2

## 4.20. 获取接收解调器参数

### 4.20.1. 命令帧定义

获取当前读写器接收解调器参数。解调器参数有 Mixer 增益，中频放大器 IF AMP 增益和信号解调阈值。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | F1 | 00 | 00 | F1 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xF1
指令参数长度 PL: 0x0000
校验位 Checksum: 0xF1

### 4.20.2. 响应帧定义

如果获取信道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Mixer_G | IF_G | Thrd(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | F1 | 00 | 04 | 03 | 06 | 01 |
| **Thrd(LSB)** | **Checksum** | **End** | | | | | |
| B0 | B0 | 7E | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0xF1
指令参数长度 PL: 0x0004
混频器增益 Mixer_G: 0x03(混频器 Mixer 增益为 9dB)
中频放大器增益 IF_G: 0x06(中频放大器 IF AMP 增益为 36dB)
信号解调阈值 Thrd: 0x01B0(信号解调阈值越小能解调的标签返回 RSSI 越低，但越不稳定，低于一定值完全不能解调；相反阈值越大能解调的标签返回信号 RSSI 越大，距离越近，越稳定。0x01B0 是推荐的最小值)
校验位 Checksum: 0xB0

**混频器 Mixer 增益表**

| Type | Mixer_G(dB) |
|---|---|
| 0x00 | 0 |
| 0x01 | 3 |
| 0x02 | 6 |
| 0x03 | 9 |
| 0x04 | 12 |
| 0x05 | 15 |
| 0x06 | 16 |

**中频放大器 IF AMP 增益表**

| Type | IF_G(dB) |
|---|---|
| 0x00 | 12 |
| 0x01 | 18 |
| 0x02 | 21 |
| 0x03 | 24 |
| 0x04 | 27 |
| 0x05 | 30 |
| 0x06 | 36 |
| 0x07 | 40 |

## 4.21. 设置接收解调器参数

### 4.21.1. 命令帧定义

设置当前读写器接收解调器参数。解调器参数有 Mixer 增益，中频放大器 IF AMP 增益和信号解调阈值。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | Mixer_G | IF_G | Thrd(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 00 | F0 | 00 | 04 | 03 | 06 | 01 |
| **Thrd(LSB)** | **Checksum** | **End** | | | | | |
| B0 | AE | 7E | | | | | |

帧类型 Type: 0x00

指令代码 Command: 0xF0

指令参数长度 PL: 0x0004

混频器增益 Mixer_G: 0x03(混频器 Mixer 增益为 9dB)

中频放大器增益 IF_G: 0x06(中频放大器 IF AMP 增益为 36dB)

信号解调阈值 Thrd: 0x01B0(信号解调阈值越小能解调的标签返回 RSSI 越低，但越不稳定，低于一定值完全不能解调；相反阈值越大能解调的标签返回信号 RSSI 越大，距离越近，越稳定。0x01B0 是推荐的最小值)

校验位 Checksum: 0xAE

**混频器 Mixer 增益表**

| Type | Mixer_G(dB) |
|---|---|
| 0x00 | 0 |
| 0x01 | 3 |
| 0x02 | 6 |
| 0x03 | 9 |
| 0x04 | 12 |
| 0x05 | 15 |
| 0x06 | 16 |

**中频放大器 IF AMP 增益表**

| Type | IF_G(dB) |
|---|---|
| 0x00 | 12 |
| 0x01 | 18 |
| 0x02 | 21 |
| 0x03 | 24 |
| 0x04 | 27 |
| 0x05 | 30 |
| 0x06 | 36 |
| 0x07 | 40 |

### 4.21.2. 响应帧定义

如果获取信道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | F0 | 00 | 01 | 00 | F2 | 7E |

帧类型 Type: 0x01

指令代码 Command: 0xF0

指令参数长度 PL: 0x0001

| | |
|---|---|
| 指令参数 Parameter: | 0x00 |
| 校验位 Checksum: | 0xF1 |

## 4.22. 测试射频输入端阻塞信号

### 4.22.1. 命令帧定义

测试射频输入端阻塞信号 Scan Jammer，用于检测读写器天线在当前地区每个信道的阻塞信号大小。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | F2 | 00 | 00 | F2 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xF2
指令参数长度 PL: 0x0000
校验位 Checksum: 0xF2

### 4.22.2. 响应帧定义

如果在中国 900MHz 频段下，一共 20 个信道，测试射频输入端阻塞信号 Scan Jammer 道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | CH_L | CH_H | JMR(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | F2 | 00 | 16 | 00 | 13 | F2 |
| | | | | | | | |
| F1 | F0 | EF | EC | EA | E8 | EA | EC |
| | | | | | | | |
| EE | F0 | F1 | F5 | F5 | F5 | F6 | F5 |
| | | JMR(LSB) | Checksum | End | | | |
| F5 | F5 | F5 | DD | 7E | | | |

帧类型 Type: 0x01
指令代码 Command: 0xF2
指令参数长度 PL: 0x0016
测试起始信道 CH_L: 0x00(测试起始信道 Index 为 0)
测试结束信道 CH_H: 0x13(测试结束信道 Index 为 19)
信道阻塞信号 JMR: 0xF2F1F0EFECEAE8EAECEEF0F1F5F5F5F6F5F5F5F5(其中 0xF2 为 -14dBm)
校验位 Checksum: 0xDD

## 4.23. 测试信道 RSSI

### 4.23.1. 命令帧定义

测试射频输入端 RSSI 信号大小，用于检测当前环境下有无读写器在工作。例如：

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | F3 | 00 | 00 | F3 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xF3
指令参数长度 PL: 0x0000
校验位 Checksum: 0xF3

### 4.23.2. 响应帧定义

如果在中国 900MHz 频段下，一共 20 个信道，检测每个信道 RSSI 道执行正确，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | CH_L | CH_H | RSSI(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | F3 | 00 | 16 | 00 | 13 | BA |
| | | | | | | | |
| BA | BA | BA | BA | BA | BA | BA | BA |
| | | | | | | | |
| BA | BA | BA | BA | BA | BA | BA | BA |
| | | RSSI(LSB) | Checksum | End | | | |
| BA | BA | BA | A5 | 7E | | | |

帧类型 Type: 0x01
指令代码 Command: 0xF2
指令参数长度 PL: 0x0016
测试起始信道 CH_L: 0x00(测试起始信道 Index 为 0)
测试结束信道 CH_H: 0x13(测试结束信道 Index 为 19)
信道阻塞信号 JMR: 0xBABABABABABABABABABABABABABABABABABABABABABA(其中 0xBA 为 -70dBm，检测 RSSI 为最小值)
校验位 Checksum: 0xDD

## 4.24. 控制 IO 端口

### 4.24.1. 命令帧定义

设置 IO 端口的方向，读取 IO 电平以及设置 IO 电平，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter0 | Parameter1 | Parameter2 |
|---|---|---|---|---|---|---|---|
| BB | 01 | 1A | 00 | 03 | 00 | 04 | 01 |
| **Checksum** | **End** | | | | | | |
| 22 | 7E | | | | | | |

帧类型 Type: 0x00
指令代码 Command: 0x1A
指令参数长度 PL: 0x0003
指令参数 Parameter: 0x00 0x04 0x01
校验位 Checksum: 0x22

参数说明:

| 编号 | 描述 | 长度 | 说明 |
|---|---|---|---|
| 0 | 参数 0 | 1 byte | 操作类型选择:<br>0x00：设置 IO 方向；<br>0x01：设置 IO 电平；<br>0x02：读取 IO 电平。<br>要操作的管脚在参数 1 中指定 |
| 1 | 参数 1 | 1 byte | 参数值范围为 0x01~0x04，分别对应要操作的端口 IO1~IO4 |
| 2 | 参数 2 | 1 byte | 参数值为 0x00 或 0x01。<br>Parameter0 \| Parameter2 \| 描述<br>0x00 \| 0x00 \| IO 配置为输入模式<br>0x00 \| 0x01 \| IO 配置为输出模式<br>0x01 \| 0x00 \| 设置 IO 输出为低电平<br>0x01 \| 0x01 \| 设置 IO 输出为高电平<br>当参数 0 为 0x02 时，此参数无意义。 |

### 4.24.2. 响应帧定义

响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter0 | Parameter1 | Parameter2 |
|---|---|---|---|---|---|---|---|
| BB | 01 | 1A | 00 | 03 | 00 | 04 | 01 |
| **Checksum** | **End** | | | | | | |
| 23 | 7E | | | | | | |

帧类型 Type: 0x01
指令代码 Command: 0x1A
指令参数长度 PL: 0x0003
指令参数 Parameter: 0x00 0x04 0x01
校验位 Checksum: 0x23

<table>
<tr><th>编号</th><th>描述</th><th>长度</th><th>说明</th></tr>
<tr><td>0</td><td>参数 0</td><td>1 byte</td><td>操作类型选择:<br>0x00: 设置 IO 方向;<br>0x01: 设置 IO 电平;<br>0x02: 读取 IO 电平。<br>要操作的管脚在参数 1 中指定</td></tr>
<tr><td>1</td><td>参数 1</td><td>1 byte</td><td>参数值范围为 0x01~0x04，分别对应要操作的端口 IO1~IO4</td></tr>
<tr><td>2</td><td>参数 2</td><td>1 byte</td><td>参数值为 0x00 或 0x01。
<table>
<tr><th>Parameter0</th><th>Parameter2</th><th>描述</th></tr>
<tr><td>0x00</td><td>0x00</td><td>表示 IO 配置失败</td></tr>
<tr><td>0x00</td><td>0x01</td><td>表示 IO 配置成功</td></tr>
<tr><td>0x01</td><td>0x00</td><td>表示设置 IO 输出失败</td></tr>
<tr><td>0x01</td><td>0x01</td><td>表示设置 IO 输出成功</td></tr>
<tr><td>0x02</td><td>0x00</td><td>表示对应端口为低电平</td></tr>
<tr><td>0x02</td><td>0x01</td><td>表示对应端口为高电平</td></tr>
</table></td></tr>
</table>

## 4.25. NXP ReadProtect/Reset ReadProtect 指令

NXP G2X 标签支持 ReadProtect/Reset ReadProtect 指令。当标签执行 ReadProtect 指令成功，标签的 ProtectEPC and ProtectTID 位将会被设置为'1'，标签会进入到数据保护的状态。如果让标签从数据保护状态回到正常状态，需要执行 Reset ReadProtect 指令。这条指令之前应先设置 Select 参数，以便选择指定的标签进行操作。

### 4.25.1. 命令帧定义

ReadProtect/Reset ReadProtect 指令帧定义如下:

| Header | Type | Command | PL(MSB) | PL(LSB) | AP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | E1 | 00 | 05 | 00 | 00 | FF |
| **AP(LSB)** | **Reset** | **Checksum** | **End** | | | | |
| FF | 00 | E4 | 7E | | | | |

帧类型 Type: 0x00
指令代码 Command: 0xE1
指令参数长度 PL: 0x0005
Kill Password: 0x0000FFFF
ReadProtect/Reset ReadProtect: 0x00(0x00 代表执行 ReadProtect，0x01 代表执行 Reset ReadProtect)
校验位 Checksum: 0x0B

### 4.25.2. 响应帧定义

如果 ReadProtect 指令执行正确，则响应帧为:

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | E1 | 00 | 10 | 0E | 30 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | 3D | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xE1
指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3000
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0x3D

如果 Reset ReadProtect 指令执行正确，则响应帧为:

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | E2 | 00 | 10 | 0E | 30 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | 3E | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xE2

指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3000
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0x3E

如果在执行 ReadProtect(Set/Reset 参数为 0x00)指令的时候，该标签没有在场区，指定的 EPC 代码不对或者标签没有响应，会返回错误代码 0x2A，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 2A | 2B | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x2A
校验位 Checksum: 0x2B

如果在执行 Reset ReadProtect(Set/Reset 参数为 0x01)指令的时候，该标签没有在场区或者指定的 EPC 代码不对，会返回错误代码 0x2B，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 2B | 2C | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x2B
校验位 Checksum: 0x2C

如果 Access Password 不正确，则返回错误代码 0x16，并会返回所操作的标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | 16 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 71 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0016
指令参数 Error Code: 0x16
PC+EPC 长度 UL: 0x0E
PC: 0x3000
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x71

## 4.26. NXP Change EAS 指令

NXP G2X 标签支持 Change EAS 指令。当标签执行 Change EAS 指令成功，标签的 PSF 位将会相应的变成'1'或者'0'。当标签的 PSF 位置为'1'的时候，标签将响应 EAS_Alarm 指令，否则标签不响应 EAS_Alarm 指令。这条指令之前应先设置 Select 参数，以便选择指定的标签进行操作。

### 4.26.1. 命令帧定义

Change EAS 指令帧定义如下:

| Header | Type | Command | PL(MSB) | PL(LSB) | AP(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 00 | E3 | 00 | 05 | 00 | 00 | FF |
| **AP(LSB)** | **PSF** | **Checksum** | **End** | | | | |
| FF | 01 | E7 | 7E | | | | |

帧类型 Type: 0x00
指令代码 Command: 0xE3
指令参数长度 PL: 0x0005
Kill Password: 0x0000FFFF
Set/Reset: 0x01(0x01 代表设置 PSF 位为'1'，0x00 代表设置 PSF 位为'0')
校验位 Checksum: 0xE7

### 4.26.2. 响应帧定义

如果 Change EAS 指令执行正确，则响应帧为:

| Header | Type | Command | PL(MSB) | PL(LSB) | UL | PC(MSB) | PC(LSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | E3 | 00 | 10 | 0E | 30 | 00 |
| **EPC(MSB)** | | | | | | | |
| 30 | 75 | 1F | EB | 70 | 5C | 59 | 04 |
| | | | **EPC(LSB)** | **Parameter** | **Checksum** | **End** | |
| E3 | D5 | 0D | 70 | 00 | 3F | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xE3
指令参数长度 PL: 0x0010
PC+EPC 长度 UL: 0x0E
PC: 0x3000
EPC: 0x30751FEB705C5904E3D50D70
指令参数 Parameter: 0x00(执行成功)
校验位 Checksum: 0x3F

如果在执行 Change EAS 指令的时候，该标签没有在场区，指定的 EPC 代码不对或者标签没有响应，会返回错误代码 0x1B，如下:

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 1B | 1C | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x1B
校验位 Checksum: 0x1C

如果 Access Password 不正确，则返回错误代码 0x16，并会返回所操作的标签的 PC+EPC，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Error Code | UL | PC(MSB) |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 10 | 16 | 0E | 34 |
| **PC(LSB)** | **EPC(MSB)** | | | | | | |
| 00 | 30 | 75 | 1F | EB | 70 | 5C | 59 |
| | | | | **EPC(LSB)** | **Checksum** | **End** | |
| 04 | E3 | D5 | 0D | 70 | 71 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0016
指令参数 Error Code: 0x16
PC+EPC 长度 UL: 0x0E
PC: 0x3000
EPC: 0x30751FEB705C5904E3D50D70
校验位 Checksum: 0x71

## 4.27. NXP EAS_Alarm 指令

NXP G2X 标签支持 EAS_Alarm 指令。当标签接收到 EAS_Alarm 指令后，标签会立刻返回 64bits EAS-Alarm code。注意只有当标签的 PSF 位置为'1'的时候，标签才响应 EAS_Alarm 指令，否则标签不响应 EAS_Alarm 指令。该指令适合于电子商品防窃（盗）系统。

### 4.27.1. 命令帧定义

EAS_Alarm 指令。

| Header | Type | Command | PL(MSB) | PL(LSB) | Checksum | End |
|---|---|---|---|---|---|---|
| BB | 00 | E4 | 00 | 00 | E4 | 7E |

帧类型 Type: 0x00
指令代码 Command: 0xE4
指令参数长度 PL: 0x0000
校验位 Checksum: 0xE4

### 4.27.2. 响应帧定义

如果 EAS_Alarm 指令执行成功，有标签响应并返回了正确的 64bits EAS-Alarm code，则响应帧为：

| Header | Type | Command | PL(MSB) | PL(LSB) | EAS-Alarm code(MSB) | | |
|---|---|---|---|---|---|---|---|
| BB | 01 | E4 | 00 | 08 | 69 | 0A | EC |
| | | | | EAS-Alarm code(LSB) | Checksum | End | |
| 7C | D2 | 15 | D8 | F9 | 80 | 7E | |

帧类型 Type: 0x01
指令代码 Command: 0xE3
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x00
校验位 Checksum: 0xE5

如果在执行 EAS_Alarm 指令的时候，没有标签响应，会返回错误代码 0x1D，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 1D | 1E | 7E |

帧类型 Type: 0x01
指令代码 Command: 0xFF
指令参数长度 PL: 0x0001
指令参数 Parameter: 0x1D
校验位 Checksum: 0x1E

# 5. 指令总结

| Code | Description |
|---|---|
| 0x03 | 获取读写器模块信息 |
| 0x22 | 单次轮询指令 |
| 0x27 | 多次轮询指令 |
| 0x28 | 停止多次轮询指令 |
| 0x0C | 设置 Select 参数指令 |
| 0x12 | 设置发送 Select 指令 |
| 0x39 | 读标签数据存储区 |
| 0x49 | 写标签数据存储区 |
| 0x82 | 锁定 Lock 标签数据存储区 |
| 0x65 | 灭活 Kill 标签 |
| 0x0D | 获取 Query 参数 |
| 0x0E | 设置 Query 参数 |
| 0x07 | 设置工作地区 |
| 0xAB | 设置工作信道 |
| 0xAA | 获取工作信道 |
| 0xAD | 设置自动跳频 |
| 0xB7 | 获取发射功率 |
| 0xB6 | 设置发射功率 |
| 0xB0 | 设置发射连续载波 |
| 0xF1 | 获取接收解调器参数 |
| 0xF0 | 设置接收解调器参数 |
| 0xF2 | 测试射频输入端阻塞信号 |
| 0xF3 | 测试信道 RSSI |
| 0x1A | 控制 IO 端口 |
| 0xE1 | NXP ReadProtec/Reset ReadProtect 指令 |
| 0xE3 | NXP Change EAS 指令 |
| 0xE4 | NXP EAS-Alarm 指令 |

# 6. 命令帧执行失败的响应帧总结

如果命令帧执行失败，则 40505W-M 芯片向上位机发送执行失败的响应帧。执行失败的响应帧共用指令代码 0xFF。如果在执行失败之前没有得到标签的 EPC，则指令参数固定为 1 个 byte 的错误代码。如果在执行失败前得到了标签的 EPC，则响应帧参数为 1 个 byte 的错误代码再加上标签的 PC+EPC 数据。

例如，如果轮询命令帧执行失败，没有收到标签返回或者返回数据 CRC 校验错误，将返回错误代码 0x15，如下：

| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
|---|---|---|---|---|---|---|---|
| BB | 01 | FF | 00 | 01 | 15 | 16 | 7E |

帧类型 Type: 0x01

指令代码 Command: 0xFF(0xFF 代表命令帧执行失败)

指令参数长度 PL: 0x01

指令参数 Parameter: 0x15(为执行失败后返回的错误代码)

校验位 Checksum: 0x16

错误代码总结如下:

| Type | Code | Description |
|---|---|---|
| Command Error | 0x17 | 命令帧中指令代码错误。 |
| FHSS Fail | 0x20 | 跳频搜索信道超时。所有信道在这段时间内都被占用。 |
| Inventory Fail | 0x15 | 轮询操作失败。没有标签返回或者返回数据 CRC 校验错误。 |
| Access Fail | 0x16 | 访问标签失败，有可能是访问密码 password 不对。 |
| Read Fail | 0x09 | 读标签数据存数区失败。标签没有返回或者返回数据 CRC 校验错误 |
| Read Error | 0xA0 \| Error code | 读标签数据存储区错误。返回的代码由 0xA0 位或 Error Code 得到。Error code 信息详见下表。 |
| Write Fail | 0x10 | 写标签数据存数区失败。标签没有返回或者返回数据 CRC 校验错误。 |
| Write Error | 0xB0 \| Error code | 写标签数据存储区错误。返回的代码由 0xB0 位或 Error Code 得到。Error code 信息详见下表。 |
| Lock Fail | 0x13 | 锁定标签数据存数区失败。标签没有返回或者返回数据 CRC 校验错误。 |
| Lock Error | 0xC0 \| Error code | 锁定标签数据存储区错误。返回的代码由 0xC0 位或 Error Code 得到。Error code 信息详见下表。 |
| Kill Fail | 0x12 | 灭活标签失败。标签没有返回或者返回数据 CRC 校验错误。 |
| Kill Error | 0xD0 \| Error code | 灭活标签错误。返回的代码由 0xC0 位或 Error Code 得到。Error code 信息详见下表。 |

NXP G2X 标签特有指令错误代码：

| | | |
|---|---|---|
| ReadProtect Fail | 0x2A | ReadProtect 指令失败，标签没有返回数据或者返回数据 CRC 校验错误。 |
| Reset ReadProtect Fail | 0x2B | Reset ReadProtect 指令失败，标签没有返回数据或者返回数据 CRC 校验错误。 |
| Change EAS Fail | 0x1B | Change EAS 指令失败，标签没有返回数据或者返回数据 CRC 校验错误。 |
| NXP 特有指令标签返回的错误代码 | 0xE0 \| Error code | NXP 特有指令标签返回的错误代码，错误代码由 0xE0 或上标签返回的 Error Code 得到。 |

EPC Gen2 协议中标签返回错误代码:

**Tag error-code**

| Error-code Support | Error Code | Error code Name | Error Description |
|---|---|---|---|
| Error-specific | $00000000_2$ | Other error | 本表中没有声明的其他所有错误 |
| | $00000011_2$ | Memory overrun | 指定的标签数据存储区不存在；或者该标签不支持指定长度的 EPC，比如 XPC。 |
| | $00000100_2$ | Memory locked | 指定的标签数据存储区被锁定并且/或者是永久锁定，而且锁定状态为不可写或不可读 |
| | $00001011_2$ | Insufficient power | 标签没有收到足够的能量来进行写操作 |
| Non-specific | $00001111_2$ | Non-specific error | 标签不支持 Error-code 返回 |