

Reader 通讯协议

读写器通讯协议

适用于所有读无源标签读头

文档版本: 02

发布日期: 2008-05-20

部件编码:

1. 通信帧格式介绍

1.1. 命令帧格式定义

数据流通方向：主机———》读写器。

命令帧是主机操作读写器的数据帧，格式如下表所示：

Packet Type	Length	Command Code	Command Data	...	Command Data	Command Data	Checksum
0xA0	n+2	1 byte	Byte 1		Byte n-1	Byte n	cc

- Packet Type 是包类型域，命令帧包类型固定为 0xA0。
- Length 是包长域，表示 Length 域后帧中字节数。
- Command Code 是命令码域。
- Command Data 是命令帧中的参数域。
- Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为止所有字节的校验和。读写器接收到命令帧后需要计算校验和来检错。

1.2. 读写器命令完成响应帧格式定义

数据流通方向：读写器———》主机。

读写器命令完成响应帧是一种固定长度的数据帧，其格式如下表所示：

Packet Type	Length	Command Code	Status	Checksum
0xE4	0x03	1 byte	1 Byte	cc

- Packet Type 是包类型域，命令帧包类型固定为 0xE4。
- Length 是包长域，表示 Length 域后帧中字节数,固定为 0x03。
- Command Code 是命令码域。
- Status 是状态域。
- Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为止所有字节的校验和。读写器接收到命令帧后需要计算校验和来检错。

状态域表明读写器完成 PC 机命令后读写器状态或执行命令后结果，其规定如下表所示：

序号	值	名称	描述

1	0x00	ERR_NONE	命令成功完成
2	0x02	CRC_ERROR	CRC 校验错误
3	0x10	DRF_COMMAND_ERROR	非法命令
4	0x01	OTHER_ERROR	其他错误

1.3. 读写器发送的信息帧格式定义

数据流通方向：读写器———》主机。

信息帧是返回给主机的数据帧，比如，用于发送标签给主机，其帧格式定义如下表所示：

Packet Type	Length	Response Code	Response Data	...	Response Data	Response Data	Checksum
0xE0	n+2	1 byte	Byte 1		Byte n-1	Byte n	cc

- Packet Type 是包类型域，响应帧包类型固定为 0xE0。
- Length 是包长域，表示 Length 域后帧中字节数。
- Response Code 是信息码域，取值表示信息的类型。
- Response Data 是信息帧中的参数域。
- Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为止所有字节的校验和。PC 机接收到命令帧后需要计算校验和来检错。

2. 通信帧详细介绍

1 SO18000-6B 标签识别

主机发：

应答	数据长度	命令	卡类型	校验和
Data0	Data1	Data2	Data3	Data4
A0	03	82	01	checksum

TEST Code: A0,03,82,01,DA;

从机回：(E0 0B 82 01) 头, (E0 04 00 00 C0 B1 CD 01) ID, checksum。

1.1 EPC 标签识别

主机发：

应答	数据长度	命令	卡类型	校验和
Data0	Data1	Data2	Data3	Data4
A0	03	82	04	checksum

TEST Code: A0,03,82,04,D7;

从机回：(E0 0F 82) 头, (04) 天线号, (12 34 33 B2 DD D9 04 80 35 05 00 00)ID,checksum.

2 ISO18000-6B 标签读取

主机发：

应答	数据长度	命令	卡类型	地址	读取长度	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data0
A0	05	80	01	addr	Length	checksum

TEST CODE: A0 05 80 01 00 08 D2; 从 0x00 地址开始起, 读取 8 个 BYTE 的数据。

从机回：(E0 15 80 01 00 08) 头, (E0 04 00 00 C0 B1 CD 01) ID, (E0 04 00 00 C0 B1 CD 01) 读出的数据, checksum

2.1 EPC 标签读取

主机发：

应答	数据长度	命令	卡类型	内存位置	地址	读取长度 (字)	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7
A0	06	80	04	MemBank	addr	Length	checksum

TEST Code: A0 06 80 04 01 02 01 D2;从 0x02 地址开始起, 读取 1 个字的数据。

注：1 个字=两个 BYTE;

MemBank:

- 00₂ Reserved 保留区
- 01₂ EPC EPC
- 10₂ TID TID
- 11₂ User 用户区

从机回：E0 09 80 04 01 02 01, (12 34) ID 49 , (49) checksum.

3 ISO18000-6B 标签写入

主机发：

应答	数据长度	命令	卡类型	地址	写入长度	写入数据	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Datan
A0	05+ Length	81	01	addr	Length	D1	checksum

TEST CODE: A0 06 81 01 16 01 00 C1; 向 0x16 写入 0x00;

从机回: (E0 0E 81 01 16 01 E0 04 00 00 C0 B1 CD 01) 头, Status, checksum.

Status = 00: 写入成功;

Status = 其它值: 写入失败;

Addr 说明: 从 19 - 255 有效;

3.1 EPC 标签单个字写入

主机发:

应答	数据长度	命令	卡类型	写入方式	内存位置	地址	写入长度 (字)	写入数据	写入数据	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Datan
A0	09	81	04	WriteMode	MemBank	addr	01	D1	D2	checksum

TEST Code1: A0,09,81,04,00,01,02,01,12,34,88;

TEST Code1: A0,09,81,04,00,01,02,01,88,88,BE;

从机回: (E0 0E 81 02 02 01 08 55 60 20 00 12 34 45) 头, Status, checksum.

注: 1 个字=两个 BYTE;

MemBank:

00 ₂	Reserved	保留区
01 ₂	EPC	EPC区
10 ₂	TID	TID区
11 ₂	User	用户区

WriteMode:

00 单个字写入

Status = 00: 写入成功;

Status = 其它值: 写入失败;

Addr 说明: EPC 区从 0x02 - 0x07 有效;

3.2 EPC 标签多个字写入 (块写)

主机发:

应答	数据长度	命令	卡类型	写入方式	内存位置	地址	写入长度 (字)	写入数据	写入数据	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Datan-1	Datan
A0	07+ (Length *2)	81	04	WriteMode	MemBank	addr	Length	D1	D (Length)	checksum

TEST Code1: A0,0B,81,04,01,01,02,02,55,55,AA,AA,CC;

从机回: (E0 0E 81 03 02 02 08 55 60 20 00 12 34 45) 头, Status, checksum.

注: 1 个字=两个 BYTE;

MemBank:

00 ₂	Reserved	保留区
01 ₂	EPC	EPC
10 ₂	TID	TID
11 ₂	User	用户区

WriteMode:

01 块写

Status = 00: 写入成功;

Status = 其它值: 写入失败;

注: 大多数标签不支持。

4 ISO18000-6B 标签 LOCK

应答	数据长度	命令	卡类型	地址	校验和
Data0	Data1	Data2	Data3	Data4	Data5
A0	04	87	01	addr	checksum

Addr 说明:

0x13 以上。

从机回: E4, 03, 87, status, checksum

Status = 00: 写入成功;

Status = 其它值: 写入失败;

4.1 EPC 标签 LOCK

应答	数据长度	命令	卡类型	LOCK 类型	校验和
Data0	Data1	Data2	Data3	Data4	Data5
A0	04	87	04	LOCK 类型值	checksum

LOCK 类型值说明:

00: LOCK USER

01: LOCK TID

02: LOCK EPC

03: LOCK ACCESS

04: LOCK KILL

05: LOCK ALL

其它值: 不锁定

从机回: E4, 03, 87, status, checksum

Status = 00: 写入成功;

Status = 其它值: 写入失败;

5 EPC 标签 KILL

应答	数据长度	命令	卡类型	RFU	密码 1	密码 2	密码 3	密码 4	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9
A0	08	86	04	00	MM1	MM2	MM3	MM4	checksum

从机回: E4 03 86 status checksum

Status = 00: 写入成功;

Status = 其它值: 写入失败;

6 初始化 EPC 标签 CODE

应答	数据长度	命令	卡类型	校验和
Data0	Data1	Data2	Data3	Data4
A0	03	99	04	checksum

从机回: E4 03 99 status checksum

Status = 00: 写入成功;

Status = 其它值: 写入失败;

7 读取读头软件版本号 CODE

应答	数据长度	命令	校验和
Data0	Data1	Data2	Data4
A0	02	6A	checksum

从机回: (E0 04 6A) 头, (01 29) 版本号, (88) checksum

8 复位读写器命令帧 CODE

Packet Type	Length	Command Code	Checksum
0xA0	2	0x65	Checksum

读写器收到此命令帧后, 先返回命令完成帧, 然后读写器复位。

从机回: (E4 03 65) 头, (00) 状态位, (B4) Checksum

状态位 00: 成功 ; 其它值: 失败

9 停止读取标签 CODE

应答	数据长度	命令	校验和
Data0	Data1	Data2	Data4
A0	02	FE	checksum

主机发: A0, 02, FE, 60。

从机回: E0, 04, 88, 88, 88, 84

注: EPC 标签的操作是以“字”为单位; ISO18000-6B 标签是以“字节”为单位。

10. 重新识别标签 CODE (多标签模式下有效)

应答	数据长度	命令	校验和
Data0	Data1	Data2	Data4
A0	02	FC	checksum

成功回: E0, 04, 88, 88, 88, CheckSum;

11. 重新获取数据 CODE (多标签模式下有效)

应答	数据长度	命令	校验和
----	------	----	-----

Data0	Data1	Data2	Data4
A0	02	FF	checksum

成功回：E0, 04, N (ID 帧数), 88, 88, CheckSum; 紧接着回 N 个 ID 数据帧。

12. 设置时间 CODE (注: 有时钟功能设备才有效)

应答	数据长度	命令	操作功能	年高字节	年低字节	月	日	星期	时	分	秒	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	Data11	Data12
A0	0B	FB	FU	Year-H	Year-L	Mon	Day	Mow	Hour	Min	Sec	checksum

FU : 00 初始化日期和时钟

成功回：E0, 03, FB, Status, CheckSum;

Status = 00: 写入成功;

Status = 其它值: 写入失败;

FU : 01 读取日期和时钟;

主机发：A0, 03, FB, 01, CheckSum;

成功回：E0, 0A, FB, Year-H, Year-L, Mon, Day, Mow, Hour, Min, Sec, checksum

错误回：E0, 03, FB, 01, CheckSum;

Reader 参数设置通讯协议

1. 通信波特率设置

应答	数据长度	命令	波特率参数	校验和
Data0	Data1	Data2	Data3	Data4
A0	03	64	波特率参数	checksum

波特率参数说明:

00 : 9600

01 : 19200

02 : 38400

03 : 57600

04 : 115200

其它参数: 9600

主机发: A0 03 64 01 F8

从机回: (E4 03 64) 头, (00) 状态位, (B5) Checksum

状态位 00: 成功 ; 其它值: 失败

注: Reader 上电默认为 9600

2. 停止工作设置

应答	数据长度	命令	校验和
Data0	Data1	Data2	Data3
0xA0	0x02	0x50	Checksum(0x0E)

从机回: (E4 03 50)头, (00) 状态位, (C9) Checksum

状态位 00: 成功 ; 其它值: 失败

3. 同时查询读写器的多个设置参数

应答	数据长度	命令	查询参数的个数	查询参数指定的高位地址	查询参数指定的低位地址	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data6
0xA0	0x05	0x63	Length	Parameter address(MSB)	Parameter address(LSB)	Checksum

如：主机发：A0 05 63 05 00 20 D3 （产品标识查询）

从机回：（E0 0A 63 05 00 20）头，（ FF FF 5E FF FF）参数值，（34）Checksum

4. 查询读写器的单个设置参数

应答	数据长度	命令	查询参数指定的高位地址	查询参数指定的低位地址	校验和
Data0	Data1	Data2	Data3	Data4	Data5
0xA0	0x04	0x61	Parameter address(MSB)	Parameter address(LSB)	Checksum

如：主机发：A0 04 61 00 65 96 （查询功率）

从机回：（E0 05 61 00 65）头，（ 00 ）参数值，（55）Checksum。

5. 同时设置多个读写器参数

应答	数据长度	命令	设置参数的个数	查询参数指定的高位地址	查询参数指定的低位地址	Command Data	Command Data	Command Data	Command Data	校验和
Data0	Data1	Data2	Data3	Data4	Data5	Data5	Data5	Data5	Data (N-1)	Data N
0xA0	0x05+ Length	0x62	Length	Parameter address(MSB)	Parameter address(LSB)	Parameter value	Parameter value	Parameter value	Parameter value	Checksum

如：主机发：A0 0C 62 07 00 92 01 04 10 40 00 01 02 01 (频率设置)

从机回：（E4 03 62）头，（00）状态位，（B7）Checksum。

状态位 00: 成功；其它值：失败

6. 设置单个读写器参数

Packet Type	Length	Command Code	Command Data	Command Data	Command Data	Checksum
0xA0	5	0x60	Parameter address(MSB)	Parameter address(LSB)	Parameter value	Checksum

Parameter address(MSB)为参数在 EEPROM 中地址的高字节。

Parameter address(LSB)为参数在 EEPROM 中地址的低字节。

Parameter value 为需要设置的参数值。

读写器接收到此命令帧后，将需要设置的参数写入 EEPROM 中，并返回命令完成帧。

如：主机发：A0 05 60 00 65 96 00 （设置功率）

从机回：(E4 03 60) 头， (00) 状态位， (B9)

状态位 00: 设置成功 ; 其它值: 设置失败

附表 1 :

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x64	用户标致码	0 - 255	用户设置标致值	
0x65	发射功率	0 - 150	功率模拟量	
0x70	读写器读卡操作发生模式	1, 2, 3	1: 主从工作模式 2: 定时工作模式 3: 触发工作模式	注意: 工作在模式 2, 3 时, 主从模式仍然有效

附表 2 :

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x71	读卡时间间隔	N	数值单位为: (N*10)毫秒 其中 N 为 10 - 100;	读写器读卡操作发生模式为 2, 3 时有效
0x72	读卡器读到数据后主动发送数据的链路选择	1, 2, 3	1: RS485 链路 2: wiegand 链路 3: RS232 链路	

附表 3:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x73	Wiegand 协议选择	1, 2, 3	1: wiegand26 2: wiegand34 3: wiegand32	对 wiegand 方式有效
0x74	Wiegand 输出数据脉冲宽度	1 - 255	读卡器内部转换为时间, 时间=这个值*10(微秒)。	
0x75	Wiegand 输出数据脉冲周期	1 - 255	读卡器内部转换为时间, 时间=这个值*100(微秒)。	

0x76	Wiegand 输出重复次数	1, 2, 3	暂不支持	
0x77	Wiegand 重复输出的间隔时间	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	读卡器内部转换为时间, 时间=这个值*10(毫秒)。(暂不支持)	

附表 5:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x7C	485 链路发送数据确认	0, 1	0: 无确认 (暂不支持) 1: 有确认 (暂不支持)	对 485 方式有效
0x7D	485 链路发送方式	0, 1	0: 被动发送 (暂不支持) 1: 主动发送	

附表 6:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x80	触发管脚启用触发工作模式设置	数值低四位设置为 0 或者 1, 表示工作不触发或者触发	Bit0: 对应触发管脚 0 Bit1: 对应触发管脚 1 (暂不支持) Bit2: 对应触发管脚 2 (暂不支持) Bit3: 对应触发管脚 3 (暂不支持)	读写器读卡操作发生模式为 3 时有效
0x81	触发管脚触发模式	数值低四位设置为 0 或者 1, 表示低电平触发或者高电平触发	Bit0: 对应触发管脚 0 (支持高电平触发) Bit1: 对应触发管脚 1 (暂不支持) Bit2: 对应触发管脚 2 (暂不支持) Bit3: 对应触发管脚 3 (暂不支持)	
0x84	延迟关闭时间	0 - 240	读卡器内部转换为时间, 时间=这个值*100(毫秒)。	

附表 7:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值(十进制)	数值含义解释	其他
0x90	跳频设置	0 - 50	0: 跳频工作方式。 1--50: 固定频率工作方式, 频率值由此数值决定	
0x92~0x98	跳频-频率参数	比特设置为 0 或者 1, 表示不选定该频率或者选定该频率	从 0x92 的 BIT0 (第 1 个频率) - BIT7(第 7 个频率), 依此类推, 可设置 50 个频率循环工作	

附表 8:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值(十进制)	数值含义解释	其他
0xA0	前向链路速率	0, 1, 2	目前固定, 不受控制	
0xA1	反向链路速率	0, 1, 2	目前固定, 不受控制	

附表 9:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值(十进制)	数值含义解释	其他
0x87	单标签和多标签标别	0, 1, 2, 3	0: EPC 单标签识别 1: EPC 多标签识别 2: 18000_6B 单标签识别 3: 18000_6B 多标签识别 暂不支持	

附表 10:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值(十进制)	数值含义解释	其他
0x89	天线工作方式	1, 4	1: 单一天线工作 4: 多天线循环工作	
0x8A	选择工作天线	数值低四比特设置为 0 或者 1, 表示不选择或者选择相应天线工作。	0: 不选任何天线工作 1: 天线 1 工作 2: 天线 2 工作 4: 天线 3 工作 8: 天线 4 工作 15: 全部天线都工作	

--	--	--	--	--

附表 11:

参数在 EEPROM 中地址 (十六进制)	项目含义	设置操作的有效值 (十进制)	数值含义解释	其他
0x7B	ID 相邻判别	1, 2	1: ID 相邻判别启动 2: 不启动(实时发数据有效)	
0x7A	ID 相邻判别时间	1 - 255	读卡器内部转换为时间, 时间=这个值*1 (秒)	注: ID 相邻判别启动时, 时间值不能为 0, 否则自动转为不启动。

上述地址在命令中使用两个字节, 由于上述字节仅一个字节范围, 所以实际使用时, 填写命令中的高位字节填写为 0。
例如: 读写器读卡操作发生模式为定时工作模式, 实际填写为:

Parameter address(MSB)	Parameter address(LSB)
0x00	0x70

说明:

- 在上述命令完成后, 需要使得读写器使用新的参数工作, 使用下面两种方法之一即可:
 - 人工复位读写器, 这样操作者需要接近读写器 (重新上电);
 - 上位机远程操作, 在上位机的软件通过使用 **Reset Reader** 命令控制读写器

3. 检验和计算方法 (c语言)

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

```
return uSum;  
}
```

4. 读头设置例程

联机数据组合帧如下：

```
【Write 数据:】 A0 03 64 01 F8  
【Read 数据:】 E4 03 64 00 B5  
【Write 数据:】 A0 02 50 0E  
【Read 数据:】 E4 03 50 00 C9  
【Write 数据:】 A0 02 6A F4  
【Read 数据:】 E0 04 6A 01 29 88
```

其本工作参数设置组合帧如下：

```
【Write 数据:】 A0 05 60 00 65 87 0F  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 7E 00 7D  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 91 1E 4C  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 0C 62 07 00 92 01 04 10 40 00 01 02 01  
【Read 数据:】 E4 03 62 00 B7  
【Write 数据:】 A0 05 60 00 90 00 6B  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 8A 01 70  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 89 01 71  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 87 00 74  
【Read 数据:】 E4 03 60 00 B9
```

主从工作方式设置组合帧如下：

```
【Write 数据:】 A0 05 60 00 70 00 8B  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 05 60 00 72 00 89  
【Read 数据:】 E4 03 60 00 B9  
【Write 数据:】 A0 04 61 00 24 D7  
【Read 数据:】 E0 05 61 00 24 FF 97  
【Write 数据:】 A0 04 61 00 25 D6  
【Read 数据:】 E0 05 61 00 25 FF 96
```

定时工作方式设置组合帧如下：

【Write 数据:】 A0 05 60 00 70 02 89
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 71 0A 80
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 72 01 88
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7B 00 80
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 79 FF 83
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 64 00 97
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7A 00 81
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7D 00 7E
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7C 00 7F
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 04 61 00 24 D7
【Read 数据:】 E0 05 61 00 24 FF 97
【Write 数据:】 A0 04 61 00 25 D6
【Read 数据:】 E0 05 61 00 25 FF 96

触发工作方式设置组合帧如下：

【Write 数据:】 A0 05 60 00 70 03 88
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 71 0A 80
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 72 01 88
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7B 00 80
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 79 FF 83
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 64 00 97
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7A 00 81
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7D 01 7D
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 7C 00 7F
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 81 01 79
【Read 数据:】 E4 03 60 00 B9

【Write 数据:】 A0 05 60 00 80 01 7A
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 05 60 00 84 0A 6D
【Read 数据:】 E4 03 60 00 B9
【Write 数据:】 A0 04 61 00 24 D7
【Read 数据:】 E0 05 61 00 24 FF 97
【Write 数据:】 A0 04 61 00 25 D6
【Read 数据:】 E0 05 61 00 25 FF 96

查询工作方式组合帧如下:

【Write 数据:】 A0 04 61 00 70 8B
【Read 数据:】 E0 05 61 00 70 03 47
【Write 数据:】 A0 04 61 00 71 8A
【Read 数据:】 E0 05 61 00 71 0A 3F
【Write 数据:】 A0 04 61 00 72 89
【Read 数据:】 E0 05 61 00 72 01 47
【Write 数据:】 A0 04 61 00 7B 80
【Read 数据:】 E0 05 61 00 7B 00 3F
【Write 数据:】 A0 04 61 00 72 89
【Read 数据:】 E0 05 61 00 72 01 47
【Write 数据:】 A0 04 61 00 79 82
【Read 数据:】 E0 05 61 00 79 FF 42
【Write 数据:】 A0 04 61 00 64 97
【Read 数据:】 E0 05 61 00 64 00 56
【Write 数据:】 A0 04 61 00 7A 81
【Read 数据:】 E0 05 61 00 7A 00 40
【Write 数据:】 A0 04 61 00 7D 7E
【Read 数据:】 E0 05 61 00 7D 01 3C
【Write 数据:】 A0 04 61 00 7C 7F
【Read 数据:】 E0 05 61 00 7C 00 3E
【Write 数据:】 A0 04 61 00 81 7A
【Read 数据:】 E0 05 61 00 81 01 38
【Write 数据:】 A0 04 61 00 80 7B
【Read 数据:】 E0 05 61 00 80 01 39
【Write 数据:】 A0 04 61 00 84 77
【Read 数据:】 E0 05 61 00 84 0A 2C

查询工作参数组合帧如下:

【Write 数据:】 A0 04 61 00 65 96
【Read 数据:】 E0 05 61 00 65 87 CE
【Write 数据:】 A0 04 61 00 7E 7D
【Read 数据:】 E0 05 61 00 7E 00 3C
【Write 数据:】 A0 04 61 00 91 6A
【Read 数据:】 E0 05 61 00 91 1E 0B
【Write 数据:】 A0 04 61 00 90 6B
【Read 数据:】 E0 05 61 00 90 00 2A

【Write 数据:】 A0 05 63 07 00 92 5F
 【Read 数据:】 E0 0C 63 07 00 92 01 04 10 40 00 01 02 C0
 【Write 数据:】 A0 04 61 00 87 74
 【Read 数据:】 E0 05 61 00 87 00 33
 【Write 数据:】 A0 04 61 00 8A 71
 【Read 数据:】 E0 05 61 00 8A 01 2F

更新工作参数组合帧如下:

【Write 数据:】 A0 03 64 00 F9
 【Read 数据:】 E4 03 64 00 B5
 【Write 数据:】 A0 02 65 F9
 【Read 数据:】 E4 03 65 00 B4

5. 自动识别数据输出格式例程

5.1. ISO18000-6B 标签输出格式如下:

数据帧采用信息帧格式发送, 其中 Response Code 域为 0x58, Response Data 域共 10 字节。

下面是一个示例:

一个标签数据共十四字节: (以下数值均是 十六进制)

E0 0C 58 00 01 E0 04 00 00 41 C2 30 01 A3

其中:

E0: 头标志, 这个是固定的

0C: 长度, 这个也是固定的

58: 消息标志, 这个也是固定的

00: 用户码标志

01: 天线编号, 本次识别来自哪个天线 注: (一体化天线是固定的)

E0 04 00 00 41 C2 30 01: ID 号

A3 : 校验和, 计算: 从首个字节至倒数第二个字节, 共 13 字节。

每次读写器返回一个标签数据。

5.2. EPC G2 标签输出格式如下:

下面是一个示例:

一个标签数据共十七字节: (以下数值均是 十六进制)

00 FF E3 00 60 19 D2 6D 1C E9 AA BB CC DD 01 52 FF

其中:

00: 头标志, 这个是固定的

FF: 用户码标志

E3 00 60 19 D2 6D 1C E9 AA BB CC DD: ID 号

01: 天线编号, 本次识别来自哪个天线 注: (一体化天线是固定的)

52: 校验和, 计算: 从首个字节至倒数第三个字节, 共 15 字节。

FF: 标志, 这个是固定的

每次读写器返回一个标签数据。

5.3. EPC G2 标签有时钟功能输出格式如下:

下面是一个示例:

一个标签数据共二十字节: (以下数值均是 十六进制)

开始: **E4 03 00 00 19 E4 04 05 00 01 12**

ID

FF FF 12 34 56 78 9A BC DE F0 11 22 33 44 06 03 0C 00 0A 01

其中:

FF FF: 头标志, 这个是固定的

FF: 用户码标志

12 34 56 78 9A BC DE F0 11 22 33 44: ID 号

06 03 0C 00 0A: 日期 (月, 日), 时间 (时, 分, 秒)

01: 校验和, 计算: 从首个字节至倒数第二个字节, 共 19 字节。

5.4. EPC G2 标签读 TID 输出格式如下:

下面是一个示例:

一个标签数据共 25 字节: (以下数值均是 十六进制)

00 FF E3 00 60 19 D2 6D 1C E9 AA BB CC DD 01 E3 00 60 19 D2 6D 1C E9 ** FF

其中:

00: 头标志, 这个是固定的

FF: 用户码标志

E3 00 60 19 D2 6D 1C E9 AA BB CC DD: EPC-ID 号

E3 00 60 19 D2 6D 1C E9: TID-ID 号

01: 天线编号, 本次识别来自哪个天线 注: (一体化天线是固定的)

******: 校验和, 计算: 从首个字节至倒数第三个字节, 共 23 字节。

FF: 标志, 这个是固定的

每次读写器返回一个标签数据。