

# Inventory: inventory

- request

| 字段名称         | 比特数         |
|--------------|-------------|
| SOF          |             |
| Flags        | 8 bits      |
| Inventory    | 8 bits      |
| Optional AFI | 8 bits      |
| Mask length  | 8 bits      |
| Mask value   | 0 – 64 bits |
| CRC16        | 16 bits     |
| EOF          |             |

- responses

| 字段名称  | 比特数     |
|-------|---------|
| SOF   |         |
| Flags | 8 bits  |
| DSFID | 8 bits  |
| UID   | 64 bits |
| CRC16 | 16 bits |
| EOF   |         |

- 函数

```
1 int inventory(  
2     uint8 u8Flag,  
3     uint8 u8AFI,  
4     uint8 u8MaskLen, // bitlen  
5     uint8 au8MaskValue[8], // .  
6     uint8 *pu8RxFlags,  
7     uint8 *pu8RxDSFID,  
8     uint8 au8RxUID[8]  
9 );
```

返回:

- 1 表示清晰收到一个 VICC 的应答;
- 0 表示没有 VICC 应答;
- <0 表示有多个 VICC 应答;

如果返回  $< 0$ ，或者：VCD 发送 EOF，继续点名，或者：修改 masklen & maskvalue，重新发送 inventory()

# Stay quiet:

- request

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Stay quiet | 8 bits  |
| UID        | 64 bits |
| CRC16      | 16 bits |
| EOF        |         |

- response

无

- 函数

```
1 int stayQuiet(  
2     uint8 u8Flags,  
3     uint8 au8UID[8]  
4 );
```

# Read Single Block

- request

| 字段名称              | 比特数     |
|-------------------|---------|
| SOF               |         |
| Flags             | 8 bits  |
| Read single block | 8 bits  |
| UID               | 64 bits |
| Block number      | 8 bits  |
| CRC16             | 16 bits |
| EOF               |         |

request 中，是否存在 UID，由 flags 决定 (address mode or select mode) 。

- response

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Error code | 8 bits  |
| CRC16      | 16 bits |
| EOF        |         |

| 字段名称                  | 比特数     |
|-----------------------|---------|
| SOF                   |         |
| Flags                 | 8 bits  |
| Block security status | 8 bits  |
| Data                  |         |
| Block length          |         |
| CRC16                 | 16 bits |
| EOF                   |         |

- 函数

```

1 int readSingleBlk(
2     uint8 u8Flags,
3     uint8 au8UID[8],
4     uint8 u8BlkNo,
5     uint8 *pu8RxFlags,
6     uint8 *pu8ErrCode
7     uint8 *pu8BlkSecurityStatus,
8     uint8 au8Dat[]
9 );

```

返回值: flags.

是否返回: pu8BlkSecurityStatus, 要看 u8Flags 中 optional\_flag.

或

```

1 // 返回: 0, 正确, 否则 Flags.
2 uint8 readSingleBlk(
3     uint8 u8VisitMode, // 1 for address mode, 0 for select mode
4     uint8 u8Optionalflag; // 1 return security status;
5     uint8 au8UID[8],
6     uint8 u8BlkNo,
7     uint8 *pu8BlkSecurityStatus, // optionalflag = 1 时, 有效;
8     uint8 au8Dat[]
9 );

```

# Write Single Block

- request

| 字段名称               | 比特数     |
|--------------------|---------|
| SOF                |         |
| Flags              | 8 bits  |
| Write single block | 8 bits  |
| UID                | 64 bits |
| Block number       | 8 bits  |
| Data               |         |
| Block length       |         |
| CRC16              | 16 bits |
| EOF                |         |

- response

- 有 Error\_flag

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Error code | 8 bits  |
| CRC16      | 16 bits |
| EOF        |         |

- 无 Error\_flag

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Error code | 8 bits  |
| CRC16      | 16 bits |
| EOF        |         |

- 函数

```

1 int writeSingleBlk(
2     uint8 u8Flags,
3     uint8 au8UID[8],
4     uint8 u8BlkNo,
5     uint8 au8Dat[],
6     uint8 *pu8RxFlags,
7     uint8 *pu8ErrCode
8 );

```

## Read multiple Blocks

- request

| 字段名称                | 比特数     |
|---------------------|---------|
| SOF                 |         |
| Flags               | 8 bits  |
| Read multiple block | 8 bits  |
| UID                 | 64 bits |
| First block number  | 8 bits  |
| Number of blocks    | 8 bits  |
| CRC16               | 16 bits |
| EOF                 |         |

- response
  - 有 Error\_flag

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Error code | 8 bits  |
| CRC16      | 16 bits |
| EOF        |         |

- 无 Error\_flag

| 字段名称  | 比特数    |
|-------|--------|
| SOF   |        |
| Flags | 8 bits |

| 字段名称                  | 比特数     |
|-----------------------|---------|
| Block security status | 8 bits  |
| Data                  |         |
| Block length          |         |
| CRC16                 | 16 bits |
| EOF                   |         |

注: Data 和 Block length 的内容根据需要重复。

- 函数

```

1  int readMultipleBlks(
2      uint8 u8Flags,
3      uint8 au8UID[8],
4      uint8 u8BlkFirstNo,
5      uint8 u8BlkNums,
6      uint8 *pu8RxFlags,
7      uint8 *pu8ErrCode
8      uint8 au8BlkSecurityStatus[],
9      uint8 au8Dat[]
10 );

```

## Write multiple Blocks

- request

| 字段名称                 | 比特数     |
|----------------------|---------|
| SOF                  |         |
| Flags                | 8 bits  |
| Write multiple block | 8 bits  |
| UID                  | 64 bits |
| First block number   | 8 bits  |
| Number of blocks     | 8 bits  |
| Data                 |         |
| Block length         |         |
| CRC16                | 16 bits |
| EOF                  |         |

注: Data 和 Block length 的内容根据需要重复。

- response

- 有 Error\_flag

| 字段名称       | 比特数     |
|------------|---------|
| SOF        |         |
| Flags      | 8 bits  |
| Error code | 8 bits  |
| CRC16      | 16 bits |
| EOF        |         |

- 无 Error\_flag

| 字段名称  | 比特数     |
|-------|---------|
| SOF   |         |
| Flags | 8 bits  |
| CRC16 | 16 bits |
| EOF   |         |

- 函数

```
1 int writeMultipleBlks (  
2     uint8 u8Flags,  
3     uint8 au8UID[8],  
4     uint8 u8BlkFirstNo,  
5     uint8 u8BlkNums,  
6     uint8 au8Dat[],  
7     uint8 *pu8RxFlags,  
8     uint8 *pu8ErrCode  
9 );
```