

# RFID 读写设备 开发指南 Python

编写人：万通

**V0.0.4**

## 目录

1. 前言 .....	4
1.1. 概述 .....	4
1.2. 适用设备 .....	4
1.3. 版权说明 .....	4
1.4. 读写基本流程 .....	5
2. 快速上手 .....	5
3. 连接说明 .....	6
3.1. RS232 连接 .....	6
3.2. RS485 连接 .....	7
3.3. TCP 客户端连接 .....	7
3.4. TCP 服务器监听 .....	7
3.5. USB-HID 连接 .....	7
3.6. 关闭连接 .....	8
4. 事件说明 .....	8
4.1. ISO18000-6C 标签上报事件 .....	8
4.2. ISO18000-6C 标签上报结束事件 .....	9
4.3. ISO18000-6B 标签上报事件 .....	9
4.4. ISO18000-6B 标签上报结束事件 .....	10
4.5. 国标标签上报事件 .....	10
4.6. 国标标签上报结束事件 .....	11
4.7. 国军标标签上报事件 .....	11
4.8. 国军标标签上报结束事件 .....	12
4.9. GPI 触发上报事件 .....	13
4.10. GPI 触发上报结束事件 .....	13
4.11. TCP 连接断开事件 .....	14
4.12. TCP 连接事件 .....	14
5. 消息配置与查询说明 .....	15
5.1. 发送同步消息 .....	15
6. 消息说明 .....	16
6.1. 读写器配置与管理 .....	16
6.1.1. 重启读写器 .....	16
6.1.2. 配置与查询串口参数 .....	16
6.1.3. 配置 GPO 状态参数 .....	16
6.1.4. 查询 GPI 状态参数 .....	17
6.1.5. 配置与查询 GPI 触发参数 .....	17
6.1.6. 查询软件基带版本 .....	18
6.1.7. 查询读写器 RFID 能力 .....	18
6.1.8. 查询读写器信息 .....	18
6.1.9. 蜂鸣器控制 .....	19
6.2. RFID 配置与操作 .....	19
6.2.1. 停止指令 .....	19
6.2.2. 配置与查询读写器功率 .....	20

6.2.3. 配置与查询读写器工作频段 .....	20
6.2.4. 配置与查询 EPC 基带参数 .....	20
6.2.5. 配置与查询 GB 基带参数 .....	21
6.2.6. 配置与查询标签上传参数 .....	21
6.2.7. 读 EPC 标签 .....	22
6.2.8. 写 EPC 标签 .....	22
6.2.9. 锁 EPC 标签 .....	23
6.2.10. 灭活 EPC 标签 .....	23
6.2.11. 读 6B 标签 .....	24
6.2.12. 写 6B 标签 .....	24
6.2.13. 6B 标签锁定 .....	25
6.2.14. 6B 标签锁定查询 .....	25
6.2.15. 读 GB 标签 .....	25
6.2.16. 写 GB 标签 .....	26
6.2.17. 锁 GB 标签 .....	26
6.2.18. 灭活 GB 标签 .....	27
6.2.19. 读 GJB 标签 .....	27
6.2.20. 写 GJB 标签 .....	28
6.2.21. 锁 GJB 标签 .....	28
6.2.22. 灭活 GJB 标签 .....	29
7. 参数说明 .....	29
7.1.1. 6C 标签选择参数 .....	29
7.1.2. 6C 标签读取 TID 参数 .....	30
7.1.3. 6C 标签读取用户数据区参数 .....	30
7.1.4. 6B 标签读用户数据区参数 .....	30
7.1.5. GB 标签选择参数 .....	31
7.1.6. GB 标签读用户数据区参数 .....	31
7.1.7. GJB 标签选择参数 .....	31
7.1.8. GJB 标签读取用户数据区参数 .....	31
8. 附录 1 .....	32
读写器所支持的频段列表 .....	32

# 1. 前言

## 1.1. 概述

为了方便进行二次开发，我们提供 Python 进行运行的函数库。该库采用 Python 语言编写并封装成标准的 tar.gz 包。

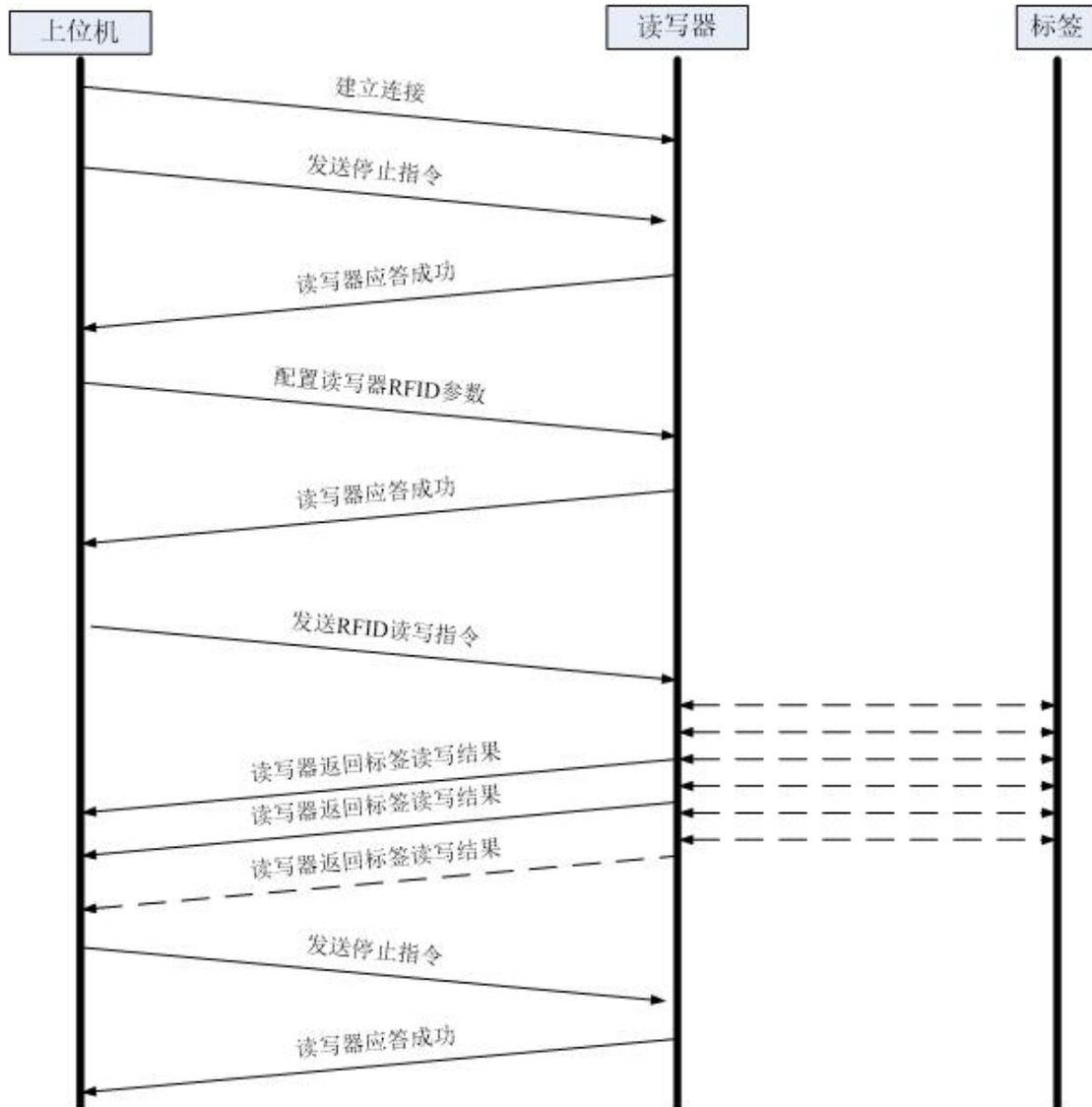
## 1.2. 适用设备

功能模块	适用设备类型
读写器配置与管理	
RFID 配置与操作	所有设备

## 1.3. 版权说明

文档的所有内容，包括文字、图片均为原创。对未经许可，擅自使用在商业用者，本公司保留追究其法律责任的权利。未经授权，使用者不得擅自添加、修改、删除本文档内容，不得以网络、光盘等方式进行传播。如若违反，后果自负。

## 1.4. 读写基本流程



## 2. 快速上手

```
from uhf.reader import *

def receivedEpc(epcInfo: LogBaseEpcInfo):
    if epcInfo.result == 0:
        print(epcInfo.epc)

def uploadEpcOver(epcOver: LogBaseEpcOver):
```

```

print("LogBaseEpcOver")

if __name__ == '__main__':

    client = GClient()
    # 串口连接 True 即成功连接
    if client.openSerial(("COM7", 115200)): # 不要重复连接, 可调用 close 释放资源再次连接, 未 close
    的 client 可复用
        # 订阅盘点回调
        client.callEpcInfo = receivedEpc
        client.callEpcOver = uploadEpcOver

        # 盘点 6c 1 号天线 循环盘点
        msg = MsgBaseInventoryEpc(EnumG.AntennaNo_1.value,
                                   EnumG.InventoryMode_Inventory.value)

        if client.sendSynMsg(msg) == 0:
            print(msg.rtMsg)

        sleep(5) # 盘点 5s

        stop = MsgBaseStop() # 停止盘点
        if client.sendSynMsg(stop) == 0:
            print(stop.rtMsg)

        client.close() # 断开连接 释放资源

```

## 3. 连接说明

### 3.1. RS232 连接

类	GClient
函数	openSerial(readName: (), kwargs) --> bool
说明	readName-->("COM7", 115200)  串口号, 波特率   kwargs-->timeout  默认即可

## 3.2. RS485 连接

类	GClient
函数	<code>openSerial485(readName: (), **kwargs) --&gt; bool</code>
说明	<code>readName--&gt;("COM7",115200,1)</code> 串口号, 波特率, 地址   <code>kwargs--&gt;timeout</code> 默认即可

## 3.3. TCP 客户端连接

类	GClient
函数	<code>openTcp(readName: (), **kwargs) --&gt; bool</code>
说明	<code>readName--&gt;("192.168.1.168",8160)</code> ip, 端口号   <code>kwargs--&gt;timeout</code> 默认即可

## 3.4. TCP 服务器监听

类	GServer
方法	<code>openServer(port: int)</code>
停止监听	<code>close()</code>
说明	<code>port</code> : 上位机监听的本地接口。 使用该方法监听, 需要将 UHF 读写设备配置成“客户端模式”。 “客户端模式”配置方法详见《RFID 演示软件操作手册》。

## 3.5. USB-HID 连接

类	GClient
函数	<code>openUsbHid(param: bytes) --&gt; bool</code>
说明	<code>param--&gt;</code> 可通过函数 <code>getUsbHidPathList()</code> 或者 <code>getUsbHidPathDict()</code> 获取

### 3.6. 关闭连接

函数	<code>client.close()</code>
说明	断开连接 释放资源 [对于已经失效的连接对象，需要主动调用此方法释放资源]

## 4. 事件说明

### 4.1. ISO18000-6C 标签上报事件

类	<code>GClient</code>
示例说明	<pre>client.callEpcInfo = receiveEpc def receiveEpc(epcInfo: LogBaseEpcInfo):     Pass</pre> <p>6C 标签主去上报事件：当读写器为读卡状态时，标签信息会通过此事件上报。 示例详见“快速上手”。</p> <p><code>LogBaseEpcInfo</code>：详见“上报对象”</p>

上报对象

类	<code>LogBaseEpcInfo</code>
---	-----------------------------

<p><b>属性</b></p>	<p><b>Epc:</b> 16 进制 EPC 字符串  <b>Pc:</b> PC 值  <b>AntId:</b> 天线编号  <b>Rssi:</b> 信号强度  <b>Result:</b> 标签读取结果, 0 为读取成功, 非 0 为失败  <b>Tid:</b> 16 进制 TID 字符串  <b>Userdata:</b> 16 进制 Userdata 字符串  <b>Reserved:</b> 16 进制保留区字符串</p>
<p><b>说明</b></p>	<p>6C 标签主动上报参数。</p>

## 4.2. ISO18000-6C 标签上报结束事件

<p><b>类</b></p>	<p>GClient</p>
<p><b>示例说明</b></p>	<pre>client.callEpcOver = receiveEpcOver def receiveEpcOver(epcOver: LogBaseEpcOver):     Pass</pre> <p>6C 标签主动上报结束参数, 以确保异步消息得到同步状态</p>

## 4.3. ISO18000-6B 标签上报事件

<p><b>类</b></p>	<p>GClient</p>
<p><b>示例说明</b></p>	<pre>client.call6bInfo = receive6b def receive6b(6bInfo: LogBase6bInfo):     pass</pre> <p>6B 标签主动上报事件: 当读写器为读卡状态时, 标签信息会通过此事件上报。          示例详见“快速上手”。</p> <p><b>LogBase6bInfo:</b> 详见“上报对象”</p>

## 上报对象

类	LogBase6bInfo
属性	<p>Antid: 天线编号</p> <p>Rssi: 信号强度</p> <p>Result: 标签读取结果, 0 为读取成功, 非 0 为失败</p> <p>Tid: 16 进制 TID 字符串</p> <p>Userdata: 16 进制 Userdata 字符串</p>
说明	6b 标签主动上报参数。

## 4.4. ISO18000-6B 标签上报结束事件

类	GClient
示例说明	<pre>client.call6bOver = receive6bOver</pre> <pre>def receive6bOver(6bOver: LogBase6bOver):</pre> <pre>    Pass</pre> <p>6B 标签主动上报结束参数, 以确保异步消息得到同步状态。</p>

## 4.5. 国标标签上报事件

类	GClient
示例说明	<pre>client.callGbInfo = receiveGbInfo</pre> <pre>def receiveGbInfo(gbInfo: LogBaseGbInfo):</pre> <pre>    pass</pre> <p>GB 标签主动上报事件: 当读写器为读卡状态时, 标签信息会通过此事件上报。 示例详见“快速上手”。</p> <p>LogBaseGbInfo: 详见“上报对象”</p>

## 上报对象

类	LogBaseGbInfo
属性	<p>Epc: 16 进制 EPC 字符串(标签编码区) Pc: PC 值 AntId: 天线编号 Rssi: 信号强度 Result: 标签读取结果, 0 为读取成功, 非 0 为失败 Tid: 16 进制 TID 字符串(标签信息区) Userdata: 16 进制 Userdata 字符串 BUser: Userdata 字节数组</p>
说明	Gb 标签主动上报参数。

## 4.6. 国标标签上报结束事件

类	GClient
示例说明	<pre>client.callGbOver = receiveGbOver def receiveGbOver(gbOver: LogBaseGbOver):     pass</pre> <p>GB 标签主动上报结束参数, 以确保异步消息得到同步状态。</p>

## 4.7. 国军标标签上报事件

类	GClient
---	---------

<b>示例说明</b>	<pre>client.callGJbInfo = receiveGJbInfo def receiveGJbInfo(gjbInfo: LogBaseGJbInfo):     pass</pre> <p>GJb 标签主动上报事件：当读写器为读卡状态时，标签信息会通过此事件上报。 示例详见“快速上手”。</p> <p>LogBaseGJbInfo：详见“上报对象”</p>
-------------	--

上报对象

<b>类</b>	LogBaseGJbInfo
<b>属性</b>	<p>Epc: 16 进制 EPC 字符串(标签编码区)</p> <p>Pc: PC 值</p> <p>Antid: 天线编号</p> <p>Rssi: 信号强度</p> <p>Result: 标签读取结果, 0 为读取成功, 非 0 为失败</p> <p>Tid: 16 进制 TID 字符串(标签信息区)</p> <p>Userdata: 16 进制 Userdata 字符串</p> <p>BUser: Userdata 字节数组</p>
<b>说明</b>	Gb 标签主动上报参数。

## 4.8. 国军标标签上报结束事件

<b>类</b>	GClient
<b>示例说明</b>	<pre>client.callGJbOver = receiveGJbOver def receiveGJbOver(gjbOver: LogBaseGJbOver):     pass</pre> <p>GJb 标签主动上报结束参数，以确保异步消息得到同步状态。</p>

## 4.9. GPI 触发上报事件

类	GClient
示例说明	<pre>client.callGpiStart = gpiStart def gpiStart(start: LogGpiStart):     pass</pre> <p>当触发起始条件满足时，读写器会主动上传一条通知消息，通知上位机触发操作已开始 LogGpiStart: 详见“上报对象”</p>

上报对象

类	LogAppGpiStart
属性	<p>gpiPort: 触发 GPI 端口号 gpiPortLevel: GPI 端口电平 systemTime: 当前系统时间</p>
说明	GPI 触发上报参数。

## 4.10. GPI 触发上报结束事件

类	GClient
示例说明	<pre>client.callGpiOver = gpiOver def gpiOver(over: LogGpiOver):     pass</pre> <p>当触发停止条件满足时，读写器会主动上传一条通知消息，通知上位机触发操作已开始 LogGpiOver: 详见“上报对象”</p>

上报对象

类	LogAppGpiOver
属性	<p>gpiPort: 触发 GPI 端口号</p> <p>gpiPortLevel: GPI 端口电平</p> <p>systemTime: 当前系统时间</p>
说明	GPI 触发停止上报参数。

## 4.11. TCP 连接断开事件

类	GClient
示例	<pre>client.callTcpDisconnect = tcpDisconnect def tcpDisconnect(readerName: ()):     pass</pre>
说明	<p>说明:</p> <ul style="list-style-type: none"> <li>➢ 连接处于 TCP，当远程连接主动断开或者物理层异常时，此事件上报。</li> <li>➢ 此事件上报后，需由上位机（调用者）释放连接对象，否则事件会循环上报，直至连接对象被释放。</li> <li>➢ 为满足不同需求，是否重连该远程设备，由上位机（调用者）自主控制。</li> </ul> <p>readerName: 连接对象名称, ('192.168.1.168', 8160)。</p>

## 4.12. TCP 连接事件

类	GServer
示例	<pre>g_server.callGClientConnected = clientConnected def clientConnected(readerSerialNumber: str, gClient: GClient):     pass</pre>
说明	<p>TCP 处于监听状态，远程读写设备主动连接上位机时，会触发此事件上报。</p> <p>readerSerialNumber: 唯一序列号</p> <p>gClient: 详见“上报对象”。</p>

上报对象

类	GClient
说明	说明：此连接对象与其他主动连接的对象相同，用法完全一致。

## 5. 消息配置与查询说明

### 5.1. 发送同步消息

类	GClient
方法	<code>def sendSynMsg(msg: Message, timeout=3)</code> 同步消息 等待返回值
方法 1	<code>def senUnSynMsg(msg: Message)</code> 非同步消息 不等待返回值
返回值	<code>msg.rtCode</code> : 消息返回码 0 为操作成功, 非 0 操作失败。 <code>msg.rtMsg</code> : 通信成功下返回操作信息, 通信失败下返回 <code>None</code>
说明	<u>“读写器配置与管理”、“RFID 配置与操作”等消息，均通过此方法发送。</u>

#### 代码示例 1

```
// 停止指令，空闲态
stop = MsgBaseStop()
if client.sendSynMsg(stop) == 0: #快捷判断写法
    print(stop.rtMsg)
```

#### 代码示例 2

```
// 功率配置，将 4 个天线功率都设置为 30dBm.
dicPower = {"1": 30, "2": 30, "3": 30, "4": 30}
msg = MsgBaseSetPower(**dicPower)
if client.sendSynMsg(msg) == 0:
    print(msg.rtMsg)
```

## 6. 消息说明

### 6.1. 读写器配置与管理

#### 6.1.1. 重启读写器

类	<a href="#">MsgAppReset</a>
属性	无
说明	设备重启消息，一般在修改需要重启生效的配置后执行。

#### 6.1.2. 配置与查询串口参数

配置类	<a href="#">MsgAppSetSerialParam(baudRateIndex: int)</a>
查询类	<a href="#">MsgAppGetSerialParam</a>
属性	<a href="#">baudRateIndex</a> : 波特率索引（0, 9600 bps; 1, 19200 bps; 2, 115200 bps; 3, 230400 bps; 4, 460800bps）
说明	持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验配置设备串口参数。

#### 6.1.3. 配置 GPO 状态参数

配置类	<a href="#">MsgAppSetGpo(**kwargs)</a>
属性	<a href="#">gpo1</a> : 0（低，继电器断开） 1（高，继电器闭合） <a href="#">gpo2</a> : 0（低，继电器断开） 1（高，继电器闭合） <a href="#">gpo3</a> : 0（低，继电器断开） 1（高，继电器闭合） <a href="#">gpo4</a> : 0（低，继电器断开） 1（高，继电器闭合）

说明	<p>(持久化配置, 断电保存)配置设备 GPO 参数。</p> <p>注: 对于不需要控制状态的 GPO, 无须赋值。</p>
----	--

#### 6.1.4. 查询 GPI 状态参数

查询类	MsgAppGetGpiState
属性	dicGpi: 对应 GPI 的电平状态( 类型: dict , key: GPI 索引号, value: 电平状态(0 低, 1 高))
说明	<p>查询设备 GPI 状态。</p> <p>注: 索引号从 1 开始。</p>

#### 6.1.5. 配置与查询 GPI 触发参数

配置类	MsgAppSetGpiTrigger
查询类	MsgAppGetGpiTrigger
属性	<p>gpiPort: GPI 端口号, 索引从 0 开始</p> <p>triggerStart: 触发开始 (0 触发关闭, 1 低电平触发, 2 高电平触发, 3 上升沿触发, 4 下降沿触发, 5 任意边沿触发)</p> <p>hexTriggerCommand:触发绑定命令 (Hex, 可为空)</p> <p>triggerOver:触发停止 (0 不停止, 1 低电平触发, 2 高电平触发, 3 上升沿触发, 4 下降沿触发, 5 任意边沿触发, 6 延时停止)</p> <p>overDelayTime:延时停止时间 (仅当停止条件为“延时停止”生效)</p> <p>levelUploadSwitch:触发不停止时 IO 电平变化上传开关 (0 不上传, 1 上传)</p>
说明	<p>持久化配置, 断电保存, 请勿做重复配置操作, 需要频繁配置时, 请先查询校验配置设备 GPI 触发参数。</p> <p>注: 此配置需在设备空闲时修改 (即循环读卡状态无法更改配置)。</p>

### 6.1.6. 查询软件基带版本

查询类	MsgAppGetBaseVersion
属性	baseVersions: 基带软件版本
说明	用于获取基带软件的版本号。

### 6.1.7. 查询读写器 RFID 能力

查询类	MsgBaseGetCapabilities
属性	<p>MaxPower: 最大支持功率</p> <p>MinPower: 最小支持功率</p> <p>AntennaCount: 天线数量</p> <p>FrequencyArray: 支持的频段列表，</p> <ul style="list-style-type: none"><li>0, 国标 920~925MHz</li><li>1, 国标 840~845MHz</li><li>2, 国标 840~845MHz 和 920~925MHz</li><li>3, FCC, 902~928MHz</li><li>4, ETSI, 866~868MHz</li></ul> <p>ProtocolArray: 支持的协议列表，</p> <ul style="list-style-type: none"><li>0, ISO18000-6C/EPC C1G2</li><li>1, ISO18000-6B</li><li>2, 国标 GB/T 29768-2013</li><li>3, 国军标 GJB 7383.1-2011</li></ul>
说明	无。

### 6.1.8. 查询读写器信息

查询类	MsgAppGetReaderInfo
-----	---------------------

属性	<p>readerSerialNumber: 读写器流水号</p> <p>powerOnTime: 上电时间</p> <p>baseCompileTime: 基带编译时间</p> <p>appVersions: 应用软件版本（如：“0.1.0.0”）</p> <p>appCompileTime: 应用编译时间</p> <p>systemVersions: 操作系统版本</p>
说明	无。

### 6.1.9. 蜂鸣器控制

查询类	MsgAppSetBeep
属性	<p>beepStatus: 0-停止 1-响</p> <p>beepMode: 0-响一次 1-常响</p>
说明	无。

## 6.2. RFID 配置与操作

### 6.2.1. 停止指令

类	MsgBaseStop
属性	无
说明	<p>停止读写器所有的 RFID 操作，并使读写器进入到空闲状态。</p> <p><u>提示：当读写器处于读卡状态时，所有配置消息将无法发送，必须发送停止指令。</u></p>

## 6.2.2. 配置与查询读写器功率

配置类	MsgBaseSetPower
查询类	MsgBaseGetPower
属性	dicPower: 读写器对应天线功率( 类型: dict, key: 天线索引号, value: 天线功率值)
说明	持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验用于对读写器当前的功率进行配置。
配置示例	<pre>dicPower = {"1": 30, "2": 30, "3": 30, "4": 30} msg = MsgBaseSetPower(**dicPower) if client.sendSynMsg(msg) == 0:     print(msg.rtMsg)</pre>

## 6.2.3. 配置与查询读写器工作频段

配置类	MsgBaseSetFreqRange
查询类	MsgBaseGetFreqRange
属性	freqRangeIndex: 频段索引，具体对应关系，详见附录 1。
说明	持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验用于对读写器当前的工作频段进行配置。

## 6.2.4. 配置与查询 EPC 基带参数

配置类	MsgBaseSetBaseband
查询类	MsgBaseGetBaseband

属性	<p><b>baseSpeed</b>: EPC 基带速率 (可选)。</p> <p><b>qValue</b>: 默认 Q 值 (可选) (0~15)。</p> <p><b>session</b>: (可选) (0,Session0; 1,Session1; 2,Session2; 3,Session3)。</p> <p><b>inventoryFlag</b>: 盘存标志参数(可选)(0,仅用 Flag A 盘存;1,仅用 Flag B 盘存;2,轮流使用 Flag A 和 Flag B)。</p>
说明	持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验用于配置读写器使用的基带参数。

### 6.2.5. 配置与查询 GB 基带参数

配置类	MsgBaseSetGbBaseband
查询类	MsgBaseGetGbBaseband
属性	<p><b>speed_tc</b>: 0, Tc=6.25us; 1, Tc=12.5us</p> <p><b>speed_trext</b>:0, Trext=0, 无前导音; 1, Trext=1, 有前导音</p> <p><b>speed_k</b>:反向链路因子 K, 0, K=1/5; 1, K=3/7; 2, K=6/11; 3, K=1; 4, K=2/5; 5, K=6/7; 6, K=12/11; 7, K=2; 8~15, 保留</p> <p><b>speed_miller</b>:反向链路编码, 0, FM0; 1, Miller=2; 2, Miller=4; 3, Miller=8</p> <p><b>cin</b>:防碰撞参数, 连续空闲阈值, CIN, 默认值为 4</p> <p><b>ccn</b>:防碰撞参数, 连续碰撞阈值, CCN, 默认值为 3</p> <p><b>session</b>: (0,Session0; 1,Session1; 2,Session2; 3,Session3)。(可选)</p> <p><b>inventoryFlag</b>: 盘存标志参数(0,仅用 Flag A 盘存;1,仅用 Flag B 盘存;2,轮流使用 Flag A 和 Flag B)。(可选)</p>
说明	持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验用于配置读写器使用的国标基带参数。

### 6.2.6. 配置与查询标签上传参数

配置类	MsgBaseSetTagLog
查询类	MsgBaseGetTagLog

属性	<p><b>RepeatedTime</b>: 重复标签过滤时间（<a href="#">可选</a>）（表示在一个读卡指令执行周期内，在指定的重复过滤时间内相同的标签内容只上传一次，0~65535，时间单位：<b>10ms</b>）。</p> <p><b>RssiTV</b>: RSSI 阈值（<a href="#">可选</a>）（标签 RSSI 值低于阈值时标签数据将不上传并丢弃）。</p>
说明	<p>持久化配置，断电保存，请勿做重复配置操作，需要频繁配置时，请先查询校验用于对读写器上传参数进行配置。</p>

### 6.2.7. 读 EPC 标签

类	<p><a href="#">MsgBaseInventoryEpc</a></p>
属性	<p><b>antennaEnable</b>: 天线端口(使用天线常量,详见<a href="#">快速上手</a>)</p> <p><b>inventoryMode</b>: 连续/单次读取 (<b>0</b>: 单次读取模式，读写器尽在各个使能的天线上进行一轮读卡操作便结束读卡操作并自动进入空闲状态;<b>1</b>: 连续读取模式，读写器一直进行读卡操作直到读写器收到停止指令后结束读卡)</p> <p><b>filter</b>: 选择读取参数（<a href="#">可选</a>）（<a href="#">详见参数说明</a>）</p> <p><b>readTid</b>: TID 读取参数（<a href="#">可选</a>）（<a href="#">详见参数说明</a>）</p> <p><b>readUserdata</b>: 用户数据区读取参数（<a href="#">可选</a>）（<a href="#">详见参数说明</a>）</p> <p><b>readReserved</b>: 保留区读取参数（<a href="#">可选</a>）（<a href="#">详见参数说明</a>）</p> <p><b>hexPassword</b>: 访问密码（<a href="#">可选</a>）</p>
说明	<p>用于配置读写器的标签读取参数并启动读卡操作，任何读取标签数据操作都需要先获取到标签 EPC 码，所以任何读卡操作都会得到 EPC 码。</p>

### 6.2.8. 写 EPC 标签

类	<p><a href="#">MsgBaseWriteEpc</a></p>
属性	<p><b>AntennaEnable</b>: 天线端口</p> <p><b>Area</b>: 待写入的标签数据区(<b>0</b>，保留区；<b>1</b>，EPC 区；<b>2</b>，TID 区；<b>3</b>，用户数据区)</p> <p><b>Start</b>: 待写入标签数据区的<b>字</b>起始地址</p> <p><b>HexWriteData</b>: 待写入的数据内容（<a href="#">可选</a>）(<b>16</b> 进制)</p> <p><b>Filter</b>: 选择读取参数（<a href="#">可选</a>）（<a href="#">详见参数说明</a>）</p> <p><b>HexPassword</b>: 访问密码（<a href="#">可选</a>）</p>

说明	<ul style="list-style-type: none"> <li>➢ 读写器对 EPC 标签进行写操作，本指令定义的写操作为单次操作</li> <li>➢ ISO18000-6C 协议规定读写操作最小数据单为<u>字</u>。</li> <li>➢ EPC 区由 <b>CRC-16(第 0 个字) + PC(第 1 个字) + EPC</b> 组成：            CRC16: <b>第 0 个字，不可写</b>。            PC: <b>第 1 个字</b>，前 5 个 bit 位表示 EPC <u>字长度</u>，即 PC 计算方法为 <b>EPC 的字长度左移 11 位</b>。</li> </ul>
写入结果	<ul style="list-style-type: none"> <li>0, 写入成功</li> <li>1, 天线端口参数错误</li> <li>2, 选择参数错误</li> <li>3, 写入参数错误</li> <li>4, CRC 校验错误</li> <li>5, 功率不足</li> <li>6, 数据区溢出</li> <li>7, 数据区被锁定</li> <li>8, 访问密码错误</li> <li>9, 其他标签错误</li> <li>10, 标签丢失</li> <li>11, 读写器发送指令错误</li> </ul>

### 6.2.9. 锁 EPC 标签

类	MsgBaseLockEpc
属性	<p><b>AntennaEnable:</b> 天线端口</p> <p><b>Area:</b> 待锁定的标签数据区(0, 灭活密码区; 1, 访问密码区; 2, EPC 区; 3, TID 区; 4, 用户数据区)</p> <p><b>Mode:</b> 锁操作类型(0, 解锁; 1, 锁定; 2, 永久解锁; 3, 永久锁定)</p> <p><b>Filter:</b> 选择读取参数 (可选) (详见参数说明)</p> <p><b>HexPassword:</b> 访问密码 (可选)</p>
说明	对标签进行锁或解锁操作，本指令定义的操作为单次操作

### 6.2.10. 灭活 EPC 标签

类	MsgBaseDestoryEpc
---	-------------------

属性	<p><b>AntennaEnable:</b> 天线端口</p> <p><b>HexPassword:</b> 销毁密码</p> <p><b>Filter:</b> 选择读取参数（可选）（详见参数说明）</p>
说明	对标签进行灭活操作，进行灭活后的标签将永久失效，此操作为不可逆操作。本指令定义的操作为单次操作

### 6.2.11. 读 6B 标签

类	MsgBaseInventory6b
属性	<p><b>AntennaEnable:</b> 天线端口</p> <p><b>InventoryMode:</b> 连续/单次读取 (<b>0</b>: 单次读取模式，读写器仅在各个使能的天线上进行一轮读卡操作便结束读卡操作并自动进入空闲状态; <b>1</b>: 连续读取模式，读写器一直进行读卡操作直到读写器收到停止指令后结束读卡)</p> <p><b>Area:</b> 读取内容(<b>0</b>, 仅读取 6B TID; <b>1</b>, 读取 6B TID+用户数据; <b>2</b>, 仅读取用户数据)</p> <p><b>ReadUserdata:</b> 用户数据区读取参数（可选）（详见参数说明）</p> <p><b>HexMatchTid:</b> 待匹配 6B 标签的 TID 码（可选）(16 进制)</p>
说明	用于 ISO18000-6B 标签的数据读取操作

### 6.2.12. 写 6B 标签

类	MsgBaseWrite6b
属性	<p><b>AntennaEnable:</b> 天线端口</p> <p><b>HexMatchTid:</b> 待匹配 6B 标签的 TID 码（可选）(16 进制)</p> <p><b>Start:</b> 待写入标签数据区的字节起始地址</p> <p><b>HexWriteData:</b> 待写入的数据内容（可选）(16 进制)</p>
说明	对 6B 标签进行写操作，本指令定义的写操作为单次操作

### 6.2.13. 6B 标签锁定

类	MsgBaseLock6b
属性	AntennaEnable: 天线端口 HexMatchTid: 待匹配 6B 标签的 TID 码 (可选) (16 进制) LockIndex: 待锁定数据的字节地址
说明	对 6B 标签数据进行锁定操作, 该操作不可撤销和逆转, 本指令定义的锁定操作为单次操作

### 6.2.14. 6B 标签锁定查询

类	MsgBaseLock6bGet
属性	AntennaEnable: 天线端口 HexMatchTid: 待匹配 6B 标签的 TID 码 (可选) (16 进制) LockIndex: 待锁定查询数据的字节地址
说明	对 6B 标签数据锁定状态进行查询, 本指令定义的锁定查询操作为单次操作

### 6.2.15. 读 GB 标签

类	MsgBaseInventoryGb
---	--------------------

属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>InventoryMode:</b> 连续/单次读取（<b>0:</b> 单次读取模式，读写器尽在各个使能的天线上进行一轮读卡操作便结束读卡操作并自动进入空闲状态;<b>1:</b> 连续读取模式，读写器一直进行读卡操作直到读写器收到停止指令后结束读卡）</p> <p><b>Filter:</b> 选择读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>ReadTid:</b> TID 读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>ReadUserdata:</b> 用户数据区读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（<b>可选</b>）</p>
说明	<p>用于配置读写器的标签读取参数并启动读卡操作，任何读取标签数据操作都需要先获取到标签编码，所以任何读卡操作都会得到标签编码（EPC）。</p>

## 6.2.16. 写 GB 标签

类	MsgBaseWriteGb
属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>Area:</b> 待写入的标签数据区(0x10, 标签编码区 0x20, 标签安全区 0x30~0x3F, 用户子区 0~15 区)</p> <p><b>Start:</b> 待写入标签数据区的<b>字</b>起始地址</p> <p><b>HexWriteData:</b> 待写入的数据内容（<b>可选</b>）(16 进制)</p> <p><b>Filter:</b> 选择读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（<b>可选</b>）</p>
说明	<ul style="list-style-type: none"> <li>➤ 读写器对 GB 标签进行写操作，本指令定义的写操作为单次操作</li> <li>➤ GB 协议规定读写操作最小数据单为<b>字</b>。</li> </ul>

## 6.2.17. 锁 GB 标签

类	MsgBaseLockGb
---	---------------

属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>Area:</b> 待锁定的标签数据区(0x00 标签信息区，0x10 标签编码区，0x20 标签安全区，0x30~0x3F 用户子区 0~15)</p> <p><b>Mode:</b> 锁操作类型</p> <p>0x00, 可读可写。</p> <p>0x01, 可读不可写。</p> <p>0x02, 不可读可写。</p> <p>0x03, 不可读不可写。</p> <p>0x11, 安全模式设置为不需要鉴别；此操作区域必须为标签安全区。</p> <p>0x12, 安全模式设置为需要鉴别，不需要安全通信；此操作区域必须为标签安全区。</p> <p>0x13, 安全模式设置为需要鉴别，需要安全通信；此操作区域必须为标签安全区。</p> <p><b>Filter:</b> 选择读取参数（可选）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（可选）</p>
说明	对标签进行锁或解锁操作，本指令定义的操作为单次操作

## 6.2.18. 灭活 GB 标签

类	MsgBaseDestoryGb
属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>HexPassword:</b> 销毁密码</p> <p><b>Filter:</b> 选择读取参数（可选）（详见<a href="#">参数说明</a>）</p>
说明	对标签进灭活操作，进行灭活后的标签将永久失效，此操作为不可逆操作。本指令定义的操作为单次操作

## 6.2.19. 读 GJB 标签

类	MsgBaseInventoryGJb
---	---------------------

属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>InventoryMode:</b> 连续/单次读取（<b>0:</b> 单次读取模式，读写器尽在各个使能的天线上进行一轮读卡操作便结束读卡操作并自动进入空闲状态;<b>1:</b> 连续读取模式，读写器一直进行读卡操作直到读写器收到停止指令后结束读卡）</p> <p><b>Filter:</b> 选择读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>ReadTid:</b> TID 读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>ReadUserdata:</b> 用户数据区读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（<b>可选</b>）</p>
说明	<p>用于配置读写器的标签读取参数并启动读卡操作，任何读取标签数据操作都需要先获取到标签编码，所以任何读卡操作都会得到标签编码（EPC）。</p>

## 6.2.20. 写 GJB 标签

类	MsgBaseWriteGJb
属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>Area:</b> 待写入的标签数据区(0x01, 标签编码区 0x02, 标签安全区 0x03, 用户区)</p> <p><b>Start:</b> 待写入标签数据区的<b>字</b>起始地址</p> <p><b>HexWriteData:</b> 待写入的数据内容（<b>可选</b>）(16 进制)</p> <p><b>Filter:</b> 选择读取参数（<b>可选</b>）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（<b>可选</b>）</p>
说明	<ul style="list-style-type: none"> <li>➤ 读写器对 GJB 标签进行写操作，本指令定义的写操作为单次操作</li> <li>➤ GJB 协议规定读写操作最小数据单为<b>字</b>。</li> </ul>

## 6.2.21. 锁 GJB 标签

类	MsgBaseLockGJb
---	----------------

属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>Area:</b> 待锁定的标签数据区(0x00 标签信息区，0x01 标签编码区，0x02 标签安全区，0x03 用户区)</p> <p><b>Mode:</b> 锁操作类型</p> <p>0x00, 可读可写。</p> <p>0x01, 可读不可写。</p> <p>0x02, 不可读可写。</p> <p>0x03, 不可读不可写。</p> <p><b>Filter:</b> 选择读取参数（可选）（详见<a href="#">参数说明</a>）</p> <p><b>HexPassword:</b> 访问密码（可选）</p>
说明	对标签进行锁或解锁操作，本指令定义的操作为单次操作

## 6.2.22. 灭活 GJB 标签

类	<b>MsgBaseDestoryGJb</b>
属性	<p><b>AntennaEnable:</b> 天线端口（使用天线枚举，详见<a href="#">快速上手</a>）</p> <p><b>HexPassword:</b> 销毁密码</p> <p><b>Filter:</b> 选择读取参数（可选）（详见<a href="#">参数说明</a>）</p>
说明	对标签进灭活操作，进行灭活后的标签将永久失效，此操作为不可逆操作。本指令定义的操作为单次操作

## 7. 参数说明

### 7.1.1. 6C 标签选择参数

类	<b>ParamEpcFilter</b>
---	-----------------------

属性	<p><b>Area:</b> 要匹配的数据区(1, EPC 区; 2, TID 区; 3, 用户数据区)</p> <p><b>BitStart:</b> 匹配数据起始位地址</p> <p><b>BitLength:</b> 需要匹配的数据位长度</p> <p><b>HexData:</b> 需要匹配的数据内容 (可选) (16 进制)</p>
说明	可选参数(EPC 前 32 位为 PC 值, 故区别 EPC 时的起始位地址通常为 32)

### 7.1.2. 6C 标签读取 TID 参数

类	ParamEpcReadTid
属性	<p><b>Mode:</b> TID 读取模式配置, (0, TID 读取长度自适应, 但最大长度不超过字节 1 定义的长度; 1, 按照字节 1 定义的长度读取 TID)</p> <p><b>Len:</b> 读写器需要读取 TID 数据的字(word, 16bits, 下同)长度</p>
说明	可选参数

### 7.1.3. 6C 标签读取用户数据区参数

类	ParamEpcReadUserdata
属性	<p><b>Start:</b> 起始字地址</p> <p><b>Len:</b> 读写器需要读取的用户数据的字长度</p>
说明	可选参数

### 7.1.4. 6B 标签读用户数据区参数

类	Param6bReadUserdata
属性	<p><b>Start:</b> 用户数据起始字节地址</p> <p><b>Len:</b> 用户数据字节长度</p>
说明	可选参数

### 7.1.5. GB 标签选择参数

类	ParamEpcFilter
属性	Area: 要匹配的数据区(0x00, 标签信息区; 0x10, 标签编码区; 0x20, 标签安全区; 0x30-0x3F, 用户数据子区) BitStart: 匹配数据起始位地址 BitLength: 需要匹配的数据位长度 HexData: 需要匹配的数据内容 (可选) (16 进制)
说明	可选参数(EPC 前 16 位为 PC 值, 故区别 EPC 时的起始位地址通常为 16)

### 7.1.6. GB 标签读用户数据区参数

类	ParamGbReadUserdata
属性	ChildArea: 用户子区 Start: 用户数据起始字地址 Len: 用户数据字长度
说明	可选参数

### 7.1.7. GJB 标签选择参数

类	ParamEpcFilter
属性	Area: 要匹配的数据区(0x00, 标签信息区; 0x01, 标签编码区; 0x02, 标签安全区; 0x03, 用户数据区) BitStart: 匹配数据起始位地址 BitLength: 需要匹配的数据位长度 HexData: 需要匹配的数据内容 (可选) (16 进制)
说明	可选参数(EPC 前 16 位为 PC 值, 故区别 EPC 时的起始位地址通常为 16)

### 7.1.8. GJB 标签读取用户数据区参数

类	ParamEpcReadUserdata
---	----------------------

属性	<p><b>Start:</b> 起始字地址</p> <p><b>Len:</b> 读写器需要读取的用户数据的字长度</p>
说明	可选参数

## 8. 附录 1

读写器所支持的频段列表

索引	说明
0	国标 920~925MHz
1	国标 840~845MHz
2	国标 840~845MHz 和 920~925MHz
3	FCC, 902~928MHz
4	ETSI, 866~868MHz