



Program Guide

MG126 BLE 接口函数

Revision History:

Rev. No.	History	Issue Date	Remark
1.0	Initial issue	Jan 4, 2019	release

Important Notice:

MACROGIGA reserves the right to make changes to its products or to discontinue any integrated circuit product or service without notice. MACROGIGA integrated circuit products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications. Use in such applications is done at the sole discretion of the customer. MACROGIGA will not warrant the use of its devices in such applications.



目录

1. 概述.....	3
2. 接口函数列表.....	3
3. 接口函数说明.....	4
3.1 radio_initBle.....	4
3.2 radio_initBle.....	5
3.3 ble_run_interrupt_start.....	5
3.4 SetBleIntRunningMode.....	5
3.5 SetLePinCode.....	5
3.6 ble_set_adv_type.....	6
3.7 ble_set_adv_data.....	6
3.8 ble_set_adv_rsp_data.....	6
3.9 ble_set_name.....	7
3.10 ble_set_interval.....	7
3.11 radio_standby.....	7
3.12 radio_resume.....	8
3.13 ble_set_wakeupdly.....	8
3.14 ble_disconnect.....	8
3.15 get_ble_version.....	9
3.16 GetFirmwareInfo.....	9
3.17 ser_write_rsp_pkt.....	9
3.18 att_notFd.....	10
3.19 att_ErrorFd_eCode.....	10
3.20 att_server_rdByGrTypeRspDeviceInfo.....	11
3.21 att_server_rdByGrTypeRspPrimaryService.....	11
3.22 att_server_rd.....	12
3.23 sconn_notifydata.....	13
3.24 sconn_indicationdata.....	13
3.25 SetFixAdvChannel.....	13
3.26 test_carrier.....	14
3.27 test_SRRCCarrier.....	14
3.28 test_SRRCSpurious.....	14
3.29 radio_setBleAddr.....	15
3.30 SIG_ConnParaUpdateReq.....	15
3.31 sconn_GetConnInterval.....	15
3.32 GetRssiData.....	16



1. 概述

本文是 BLE 接口函数说明，适用于 mg_api.h 文件。

2. 接口函数列表

接口函数定义在文件 **mg_api.h** 中，主要包括如下的函数：

- 1) void radio_initBle(unsigned char txpwr, unsigned char**addr);
- 2) unsigned char radio_initBle_TO(unsigned char txpwr, unsigned char**addr, unsigned short ms_timeout);
- 3) void ble_run_interrupt_start(unsigned short adv_interval);
- 4) void SetBleIntRunningMode(void);
- 5) void SetLePinCode(unsigned char *PinCode/*6 0~9 digitals*/);
- 6) void ble_set_adv_type(unsigned char type);
- 7) void ble_set_adv_data(unsigned char* adv, unsigned char len);
- 8) void ble_set_adv_rsp_data(unsigned char* rsp, unsigned char len);
- 9) void ble_set_name(unsigned char* name, unsigned char len);
- 10) void ble_set_interval(unsigned short interval);
- 11) void radio_standby(void);
- 12) void radio_resume(void);
- 13) unsigned char ble_set_wakeupdly(unsigned short counter);
- 14) void ble_disconnect(void);
- 15) unsigned char *get_ble_version(void);
- 16) unsigned char *GetFirmwareInfo(void);
- 17) void ser_write_rsp_pkt(unsigned char pdu_type);
- 18) void att_notFd(unsigned char pdu_type, unsigned char attOpcode, unsigned short attHd);
- 19) void att_ErrorFd_ecode(unsigned char pdu_type, unsigned char attOpcode, unsigned short attHd, unsigned char errCode);
- 20) void att_server_rdByGrTypeRspDeviceInfo(unsigned char pdu_type);
- 21) void att_server_rdByGrTypeRspPrimaryService(unsigned char pdu_type,
unsigned short start_hd,



```
unsigned short end_hd,  
unsigned char*uuid,  
unsigned char uuidlen);
```

- ```
22) void att_server_rd(unsigned char pdu_type,
 unsigned char attOpcode,
 unsigned short att_hd,
 unsigned char* attValue,
 unsigned char datalen);
23) unsigned char sconn_notifydata(unsigned char* data, unsigned char len);
24) unsigned char sconn_indicationdata(unsigned char* data, unsigned char len);
25) void SetFixAdvChannel(unsigned char isFixCh37Flag);
26) void test_carrier(unsigned char freq, unsigned char txpwr);
27) void test_SRRCCarrier(unsigned char freq);
28) void test_SRRCSpurious(unsigned char freq);
29) void radio_setBleAddr(u8 addr[6]);
30) void SIG_ConnParaUpdateReq(unsigned short IntervalMin, unsigned short IntervalMax, unsigned short SlaveLatency,
 unsigned short TimeoutMultiplier);
31) unsigned short sconn_GetConnInterval(void);
32) Unsigned char GetRssiData(void);
```

### 3. 接口函数说明

#### 3.1 radio\_initBle

函数原型: void radio\_initBle(unsigned char txpwr, unsigned char\*\*addr/\*Output\*/);

函数功能: 初始化蓝牙芯片及蓝牙协议栈

输入参数: txpwr 该参数用于初始化蓝牙芯片发射功率, 可取的值有 TXPWR\_0DBM, TXPWR\_3DBM 等

输出参数: addr 该参数返回蓝牙 MAC 地址信息, 6 个字节长度。

返回值: 无

注意事项: 在 SetBleIntRunningMode 之后调用。



### 3.2 radio\_initBle

函数原型: unsigned char radio\_initBle\_TO(unsigned char txpwr, unsigned char\*\*addr, unsigned short ms\_timeout);

函数功能: 初始化蓝牙芯片及蓝牙协议栈

输入参数: txpwr 该参数用于初始化蓝牙芯片发射功率, 可取的值有 TXPWR\_0DBM, TXPWR\_3DBM 等

ms\_timeout 该参数用于蓝牙芯片初始化超时时间, 单位为毫秒, 建议值为 10~50

输出参数: addr 该参数返回蓝牙 MAC 地址信息, 6 个字节长度。

返回值: 0 表示初始化失败, 非 0 表示初始化成功。

注意事项: 在 SetBleIntRunningMode 之后调用。

### 3.3 ble\_run\_interrupt\_start

函数原型: void ble\_run\_interrupt\_start(unsigned short adv\_interval);

函数功能: 采用中断服务程序方式运行蓝牙协议

输入参数: adv\_interval, 参数的单位为 0.625ms, 如 160 表示 100ms 的广播间隔

输出参数: 无

返回值: 无

注意事项: 请参考编程指南“中断服务程序方式运行模式”

### 3.4 SetBleIntRunningMode

函数原型: void SetBleIntRunningMode(void);

函数功能: 设置中断服务程序方式

输入参数: 无

输出参数: 无

返回值: 无

注意事项: 请参考编程指南“中断服务程序方式运行模式”

### 3.5 SetLePinCode

函数原型: void SetLePinCode(unsigned char \*PinCode/\*6 0~9 digitals\*/);



函数功能：设置蓝牙配对 PIN

输入参数：PinCode，6 个 0~9 的数字

输出参数：无

返回值：无

注意事项：在 ble\_run\_interrupt\_start 前调用。缺省 PIN 为 000000，如果需要改变 PIN 可以调用此函数。

### 3.6 ble\_set\_adv\_type

函数原型：void ble\_set\_adv\_type(unsigned char type);

函数功能：设置 BLE 广播 Adv PDU Header

输入参数：type, 0-adv\_ind, 2-adv\_nonconn\_ind, 缺省为 0x80。

输出参数：无

返回值：无

注意事项：本函数可以在任意时候调用，广播数据内容立即起效。

### 3.7 ble\_set\_adv\_data

函数原型：void ble\_set\_adv\_data(unsigned char\* adv, unsigned char len);

函数功能：设置 BLE 广播数据

输入参数：adv，广播数据指针

len，广播数据长度

输出参数：无

返回值：无

注意事项：本函数可以在任意时候调用，广播数据内容立即起效。

### 3.8 ble\_set\_adv\_rsp\_data

函数原型：void ble\_set\_adv\_rsp\_data(unsigned char\* rsp, unsigned char len);

函数功能：设置 BLE 广播扫描应答数据

输入参数：rsp，广播扫描应答数据指针



len, 广播扫描应答数据长度

输出参数: 无

返回值: 无

注意事项: 1) 本函数可以在任意时候调用, 广播数据内容立即起效。

2) 调用本函数会导致函数 ble\_set\_name()失效, 但 app.c 中函数 getDeviceInfoData()依然有效。

### 3.9 ble\_set\_name

函数原型: void ble\_set\_name(unsigned char\* name,unsigned char len);

函数功能: 设置 BLE 广播应答包内名字内容

输入参数: name, 名字数据指针

len, 名字数据长度

输出参数: 无

返回值: 无

注意事项: 1) 本函数可以在任意时候调用, 数据内容立即起效。

2) 调用本函数仅会改变广播扫描应答的数据内容, 为了与 GATT 对应内容一致, 需要同步修改 app.c 中 DeviceInfo 的内容, 具体可参考 app.c 中函数 updateDeviceInfoData()的实现。

### 3.10 ble\_set\_interval

函数原型: void ble\_set\_interval(unsigned short interval);

函数功能: 设置 BLE 广播间隔时间

输入参数: interval, 广播间隔, 范围: 0x0020 ~ 0x4000, 单位 0.625ms

输出参数: 无

返回值: 无

注意事项: 本函数可以在任意时候调用, 数据内容立即起效。

### 3.11 radio\_standby

函数原型: void radio\_standby(void);



函数功能：蓝牙芯片进入 standby, 处于最省电状态

输入参数：无

输出参数：无

返回值：无

注意事项：此函数一般在 MCU 进入 standby 前调用，也可以在 UsrProcCallback 或者 ble 进入 sleep 后调用。系统进入 standby 后功耗约为 2uA。

### 3.12 radio\_resume

函数原型：void radio\_resume(void);

函数功能：蓝牙芯片从 standby 恢复

输入参数：无

输出参数：无

返回值：无

注意事项：本函数可以在 radio 进入 standby 以后需要恢复 radio 的时候调用。

### 3.13 ble\_set\_wakeupdly

函数原型：unsigned char ble\_set\_wakeupdly(unsigned short counter);

函数功能：设置 mcu 唤醒时间。mcu 使用低功耗，irq 中断唤醒会有延迟，需要调用此接口设置唤醒延迟时间。Mcu 不使用低功耗模式，不需要设置唤醒时间。

输入参数：counter: mcu 唤醒延迟时间，单位 1/64 ms。范围：0x0000~0x0040

输出参数：无

返回值：0 - fail; 1 - success

注意事项：本函数在 radio\_initBle 前调用。

### 3.14 ble\_disconnect

函数原型：void ble\_disconnect(void);

函数功能：断开已有的连接

输入参数：无





输出参数: 无

返回值: 无

注意事项: 本函数可以在任意时候调用, 立即起效。

### 3.15 get\_ble\_version

函数原型: `unsigned char *get_ble_version(void);`

函数功能: 获取蓝牙协议栈版本信息字符串

输入参数: 无

输出参数: 无

返回值: 蓝牙协议栈版本信息字符串

注意事项: 按需调用

### 3.16 GetFirmwareInfo

函数原型: `unsigned char *GetFirmwareInfo(void);`

函数功能: 获取蓝牙基带版本信息字符串

输入参数: 无

输出参数: 无

返回值: 蓝牙基带版本信息字符串

注意事项: 按需调用

### 3.17 ser\_write\_rsp\_pkt

函数原型: `void ser_write_rsp_pkt(unsigned char pdu_type);`

函数功能: 对具有 Write With Response 属性特征值写操作后的应答函数

输入参数: pdu 类型参数, 直接引用回调函数 `ser_write_rsp` 中对应参数

输出参数: 无

返回值: 无

注意事项: 对需要写应答的特征值, 如果不应答会导致连接的断开。



### 3.18 att\_notFd

函数原型: void att\_notFd(unsigned char pdu\_type, unsigned char attOpcode, unsigned short attHd );

函数功能: 对无效特征值（或没有定义的特征值）进行操作的应答函数

输入参数: pdu\_type PDU 类型参数

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

attOpcode 操作类型参数

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

attHd 对应特征值句柄值

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

输出参数: 无

返回值: 无

注意事项: 凡是无效特征值的操作需要应答本函数, 可将本函数作为缺省调用。

### 3.19 att\_ErrorFd\_eCode

函数原型: void att\_ErrorFd\_eCode(unsigned char pdu\_type, unsigned char attOpcode, unsigned short attHd, unsigned char errorCode);

函数功能: 对无效操作的应答函数

输入参数: pdu\_type PDU 类型参数

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

attOpcode 操作类型参数

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

attHd 对应特征值句柄值

直接引用回调函数 ser\_write\_rsp、att\_server\_rdyGrType 或 server\_rd\_rsp 中对应参数

errorCode 错误代码, 请参考 ATT Error Code

输出参数: 无

返回值: 无

ATT Error Code define:



```
#define ATT_ERR_INVALID_HANDLE 0x01
#define ATT_ERR_READ_NOT_PERMITTED 0x02
#define ATT_ERR_WRITE_NOT_PERMITTED 0x03
#define ATT_ERR_INVALID_PDU 0x04
#define ATT_ERR_INSUFFICIENT_AUTHEN 0x05
#define ATT_ERR_UNSUPPORTED_REQ 0x06
#define ATT_ERR_INVALID_OFFSET 0x07
#define ATT_ERR_INSUFFICIENT_AUTHOR 0x08
#define ATT_ERR_PREPARE_QUEUE_FULL 0x09
#define ATT_ERR_ATTR_NOT_FOUND 0x0a
#define ATT_ERR_ATTR_NOT_LONG 0x0b
#define ATT_ERR_INSUFFICIENT_KEY_SIZE 0x0c
#define ATT_ERR_INVALID_VALUE_SIZE 0x0d
#define ATT_ERR_UNLIKELY 0x0e
#define ATT_ERR_INSUFFICIENT_ENCRYPT 0x0f
#define ATT_ERR_UNSUPPORTED_GRP_TYPE 0x10
#define ATT_ERR_INSUFFICIENT_RESOURCES 0x11
```

### 3.20 att\_server\_rdByGrTypeRspDeviceInfo

函数原型: void att\_server\_rdByGrTypeRspDeviceInfo(unsigned char pdu\_type);

函数功能: 对缺省 Device Info 内容的应答可调用本接口函数

输入参数: pdu 类型参数, 直接引用回调函数 att\_server\_rdByGrType 中对应参数

输出参数: 无

返回值: 无

注意事项: 如果用户直接使用发布包代码, 可直接调用本接口函数。

### 3.21 att\_server\_rdByGrTypeRspPrimaryService

函数原型: void att\_server\_rdByGrTypeRspPrimaryService(unsigned char pdu\_type,  
unsigned short start\_hd,



```
unsigned short end_hd,
unsigned char*uuid,
unsigned char uuidlen);
```

函数功能：应答 Primary Service 的查询，用户需按特征值实际定义的句柄及 UUID 填充对应数据

输入参数：pdu\_type PDU 类型参数，直接引用回调函数 att\_server\_rdyGrType 中对应参数

start\_hd, 某个 Service 对应的起始句柄值

end\_hd, 某个 Service 对应的结束句柄值

uuid, 某个 Service 对应的 UUID 字串（Hex 值），如 0x180A 表示为 0x0a, 0x18。

uuidlen, 某个 Service 对应 UUID 字串的长度

输出参数：无

返回值：无

注意事项：需要严格按照特征值定义规划填充对应参数。

### 3.22 att\_server\_rd

```
函数原型：void att_server_rd(unsigned char pdu_type,
 unsigned char attOpcode,
 unsigned short att_hd,
 unsigned char* attValue,
 unsigned char datalen);
```

函数功能：读取某特征值的值。

输入参数：pdu\_type PDU 类型参数，直接引用回调函数 server\_rd\_rsp 中对应参数

attOpcode 操作对应的值，直接引用回调函数 server\_rd\_rsp 中对应参数

att\_hd 特征值对应的句柄值，直接引用回调函数 server\_rd\_rsp 中对应参数

attValue 特征值对应的值字符串指针

datalen 特征值字符串长度

输出参数：无

返回值：无

注意事项：需要按需将对应特征值内容作为应答内容，如果对应特征值内容无效可应答 att\_notFd（）。



### 3.23 sconn\_notifydata

函数原型: unsigned char sconn\_notifydata(unsigned char\* data, unsigned char len);

函数功能: 通过蓝牙发送数据

输入参数: data 需要发送的数据指针

len 数据长度

输出参数: 无

返回值: 成功发送的数据字节数

注意事项: 本接口函数会根据系统缓存情况自动拆包发送数据, 但不得阻塞循环调用。调用前需要确认 cur\_notifyhandle 正确。

### 3.24 sconn\_indicationdata

函数原型: unsigned char sconn\_indicationdata(unsigned char\* data, unsigned char len);

函数功能: 通过蓝牙发送数据

输入参数: data 需要发送的数据指针

len 数据长度

输出参数: 无

返回值: 成功发送的数据字节数

注意事项: 本接口函数会根据系统缓存情况自动拆包发送数据, 但不得阻塞循环调用。调用前需要确认 cur\_notifyhandle 正确。

### 3.25 SetFixAdvChannel

函数原型: void SetFixAdvChannel(unsigned char isFixCh37Flag);

函数功能: 在调试阶段, 为了便于空中抓包, 协议栈会固定在 37 通道广播, 协议栈缺省在所有通道广播。

输入参数: isFixCh37Flag 设置是否仅在 37 通道广播。

输出参数: 无

返回值: 无

注意事项: 仅用于调试, 需要调试时在协议栈初始化后调用 SetFixAdvChannel(1)



### 3.26 test\_carrier

函数原型: void test\_carrier(unsigned char freq, unsigned char txpwr);

函数功能: 测试实际载波发射功率, 协议栈会固定在 2400+freq 频点发送载波, 理想发射功率参考 txpwr。

输入参数: freq - 载波中心频率为(2400+freq)MHz

txpwr - 0x43 对应理想射频发射功率 0dBm

输出参数: 无

返回值: 无

注意事项: 用于载波和晶体频偏测试, 在协议栈初始化后调用 test\_carrier(80, 0x43), 然后 while(1)循环等待。

### 3.27 test\_SRRCCarrier

函数原型: void test\_SRRCCarrier(unsigned char freq);

函数功能: 用于 SRRC 测试-载波。

输入参数: freq - 载波中心频率为(2400+freq)MHz

输出参数: 无

返回值: 无

注意事项: 用于载波和晶体频偏测试, 在协议栈初始化后调用, 然后 while(1)循环等待。

### 3.28 test\_SRRCSpurious

函数原型: void test\_SRRCSpurious(unsigned char freq);

函数功能: 用于 SRRC 测试-杂散

输入参数: freq - 载波中心频率为(2400+freq)MHz

输出参数: 无

返回值: 无

注意事项: 用于杂散测试, 在协议栈初始化后调用, 然后 while(1)循环等待。



### 3.29 radio\_setBleAddr

函数原型: void radio\_setBleAddr(unsigned char addr[6]);

函数功能: 设置客户自己的蓝牙设备地址。

输入参数: addr - 蓝牙设备地址, 6 字节

输出参数: 无

返回值: 无

注意事项: 在协议栈初始化后调用

### 3.30 SIG\_ConnParaUpdateReq

函数原型: void SIG\_ConnParaUpdateReq(unsigned short IntervalMin, unsigned short IntervalMax, unsigned short SlaveLatency, unsigned short TimeoutMultiplier);

函数功能: 在连接状态下, 尝试使用用户期望范围内的发射间隔。最终的发射间隔由 central 设备决定。

输入参数: IntervalMin - 最小发射间隔, 单位 1.25ms。范围 6 到 3200。

IntervalMax - 最大发射间隔, 单位 1.25ms。范围 6 到 3200。

SlaveLatency - 回应延迟, 范围 0~500, 还要小于(SupervisionTimeout / (IntervalMax\*2)) -1。

TimeoutMultiplier - 连接断开时间。单位 10ms。范围 10 到 3200。

输出参数: 无

返回值: 无

注: SlaveLatency 建议范围 0-5。

### 3.31 sconn\_GetConnInterval

函数原型: unsigned short sconn\_GetConnInterval(void);

函数功能: 获得目前蓝牙连接当前使用的发射间隔数值, 单位为 1.25ms。

输入参数: 无

输出参数: 无

返回值: 当前蓝牙连接的间隔



### 3.32 GetRssiData

函数原型: `Unsigned char GetRssiData(void);`

函数功能: 接收广播包或者数据包时调用此函数, 可以获取接收信号强度原始数据。可以根据参考的 rssi 数值, 用此数据来计算接收信号强度。

输入参数: 无

输出参数: 无

返回值: 接收信号强度