

# CH57x BLE 协议栈库说明

版本：1B

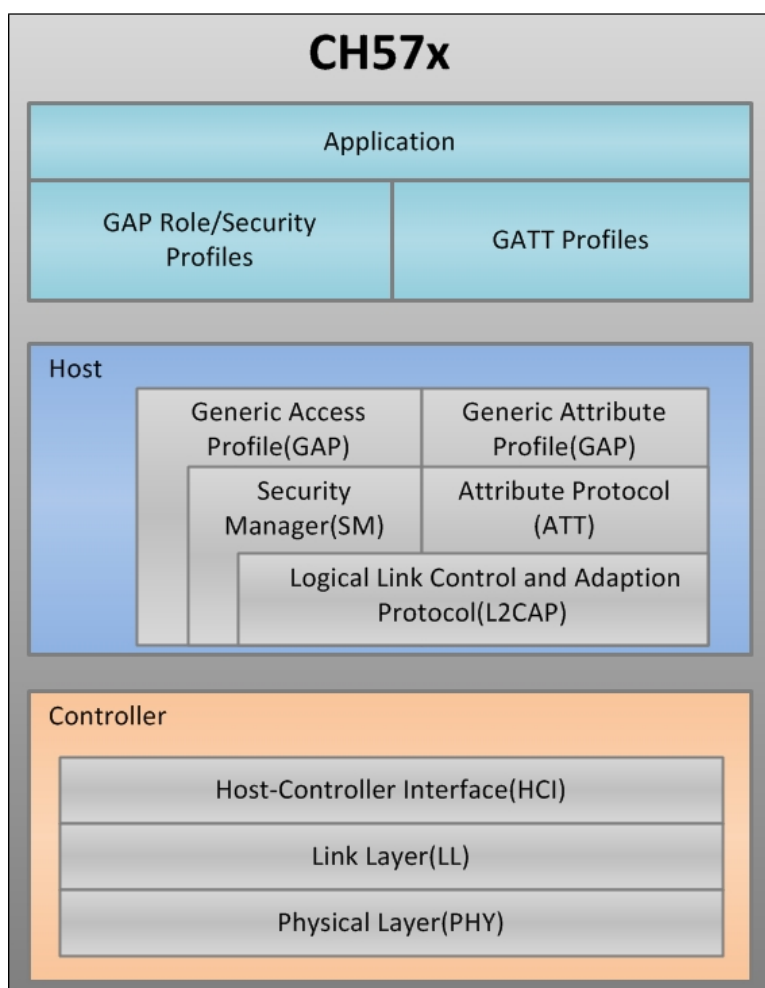
<http://wch.cn>

## 1、概述

CH57x是集成低功耗蓝牙无线通讯的ARM内核32位微控制器，片上集成2.4GHz RF收发器和基带及链路控制单端RF接口，无需外部电感，简化板级设计。

CH57xBLE.LIB提供了低功耗蓝牙4.2子程序库，支持4种广播类型，连接通信，支持长包传输；以函数库的形式实现，提供应用层的API调用，快速开发低功耗蓝牙产品。

蓝牙协议栈库由主机协议层(Host)和控制协议层(Controller)组成，只需添加应用层(Application)和服务(Profile)即可完成低功耗蓝牙开发。整体程序结构如下：



### 链路层 (Link Layer)

RF控制器。它控制设备处于准备 (standby)、广播 (advertising)、监听/扫描 (scanning)、初始化 (initiating)、连接 (connected) 这五种状态中一种。围绕这几种状态，BLE设备可以执行广播和连接等操作，链路层定义了在各种状态下的数据包格式、时序规范和接口协议。

### 通用访问规范 (Generic Access Profile)

BLE 设备内部功能对外接口层。它规定了三个方面：GAP 角色、模式和规程、安全问题。主要管理蓝牙设备的广播，连接和设备绑定。GAP 层总是作为下面四种角色之一：

广播者——不可以连接的一直在广播的设备

观察者——可以扫描广播设备，但不能发起建立连接的设备

从机——可以被连接的广播设备，可以在单个链路层连接中作为从机

主机——可以扫描广播设备并发起连接，在单个链路层或多链路层中作为主机

### 逻辑链路控制及自适应协议（Logical Link Control and Adaptation Protocol）

主机与控制器之间的适配器，提供数据封装服务。它向上连接应用层，向下连接链路层，使上层应用的操作无需关心控制器的数据细节。

### 安全管理协议（Security Manager）

提供配对和密钥分发服务，实现安全连接和数据交换。

### 属性传输协议（Attribute Protocol）

定义了属性实体的概念，包括 UUID、句柄和属性值，规定了属性的读、写、通知等操作方法和细节。

### 通用属性规范（Generic Attribute Profile）

定义了使用 ATT 的服务框架和配置文件的结构，两个设备应用数据的通信是通过协议栈的 GATT 层实现。包括 GATT 服务端和 GATT 客户端，如下：

GATT 服务端——为 GATT 客户端提供数据服务的设备

GATT 客户端——从 GATT 服务器读写应用数据的设备

## 2、库配置变量

### 2.1 协议栈库全局变量

通过 BLE\_LibInit 函数配置。参考 CH57x\_BLEInit 函数。

MEMAddr	蓝牙协议栈缓存首地址（默认值：无，必须配置）
MEMLen	蓝牙协议栈缓存的长度（默认值：无，必须配置）
SNVAddr	存储配置信息首地址（默认值：无，不保存配对信息，需配置）
SNVBlock	存储配置信息块大小（默认值：512）
SNVNum	存储配置信息块数（默认值：1）
AttMtu	配置 ATT_MTU，需发送长包则配置（默认值：27）
RxNumMax	缓存数据包个数（默认值：10）
TxNumEvent	一个连接事件发送最大包个数，需快速发送数据包则配置（默认值：1）
TxPower	发送功率（默认值：0x15（0dBm））
WakeUpTime	配置唤醒后需要的时间，以 RTC 计数值为单位（默认值：70）
OscRC32KEnable	使能内部 RC 作为 32K 时钟源（默认：0，外部 32K 晶振；1，内部 RC）
WindowWidening	窗口扩展，以 RTC 一个计数为单位（默认：8，一般不需要配置）
BLEIrqOff	写 1 关闭蓝牙中断（默认开启蓝牙中断；不建议配置）
MacAddr	配置 MAC 地址，小端（默认值：出厂值，如配置 MAC 地址则优先使用配置值）

注：灰色标记为需要配置的参数。

### 2.2 GAP 参数

读 GAP 参数函数：GAP\_GetParamValue，写 GAP 参数函数：GAP\_SetParamValue。

TGAP_CONN_PAUSE_PERIPHERAL	从机建立连接后，到发送参数更新请求的超时时间，单位秒（默认值：5）
TGAP_LIM_DISC_ADV_INT_MIN	限时可发现广播模式下的最小广播间隔，单位：0.625ms，范围：0x0020 到 0x4000（默认值：160）
TGAP_LIM_DISC_ADV_INT_MAX	限时可发现广播模式下的最大广播间隔，单位：0.625ms，范围：0x0020 到 0x4000（默认值：160）
TGAP_GEN_DISC_ADV_INT_MIN	通用广播模式下的最小广播间隔，单位：0.625ms，范围：0x0020 到 0x4000（默认值：160）
TGAP_GEN_DISC_ADV_INT_MAX	通用广播模式下的最大广播间隔，单位：0.625ms，范围：

	0x0020 到 0x4000（默认值：160）
--	--------------------------

2.3 GAP Role 参数

读 GAP Role 参数函数：GAPRole\_GetParameter，写 GAP Role 参数函数：GAPRole\_SetParameter。

GAPROLE_PROFILEROLE	GAP 角色配置
GAPROLE_ADVERT_ENABLED	广播使能配置，0：关闭广播，1：打开广播
GAPROLE_ADVERT_DATA	广播数据配置，最大 31 字节
GAPROLE_SCAN_RSP_DATA	扫描应答数据配置，最大 31 字节
GAPROLE_ADVERT_OFF_TIME	限时可发现广播模式下关闭广播的时间配置，单位秒（默认值为 30 秒）
GAPROLE_PARAM_UPDATE_ENABLE	连接参数更新使能，主动发起连接变量更新请求
GAPROLE_MIN_CONN_INTERVAL	连接参数更新建议的最小连接间隔，默认值为 6，单位 1.25ms
GAPROLE_MAX_CONN_INTERVAL	连接参数更新建议的最大连接间隔，默认值为 3200，单位 1.25ms
GAPROLE_SLAVE_LATENCY	连接参数更新建议的从机延迟应答，默认值为 0，范围 0-499
GAPROLE_TIMEOUT_MULTIPLIER	连接参数更新建议的超时，默认值为 1000，范围 10-3200，单位 10ms

2.4 GAP 服务参数

读 GAP 服务参数函数：GGS\_GetParameter，写 GAP 服务参数函数：GGS\_SetParameter。

GGG_DEVICE_NAME_ATT	最大 21 字节，设备名称，连接后显示在系统中
---------------------	-------------------------

2.5 GAP 绑定参数

读 GAP 绑定参数函数 GAPBondMgr\_GetParameter，写 GAP 绑定参数函数：GAPBondMgr\_SetParameter。

GAPBOND_BONDING_ENABLED	连接绑定信息是否保存 0：关闭（默认） 1：开启
GAPBOND_IO_CAPABILITIES	连接绑定过程中配对的交互能力 0x00：只显示（默认） 0x01：只显示，只可以输入是或否 0x02：只有输入 0x03：无显示无输入 0x04：既有显示，又可以输入
GAPBOND_PAIRING_MODE	连接绑定过程中的配对模式 0x00：不允许配置 0x01：等待配对请求 0x02：主动发起配对请求（默认）
GAPBOND_DEFAULT_PASSCODE	连接绑定过程中的配对 PIN 码，范围：000000-999999 默认值：6 个 0

3、库函数

3.1 库子程序

分类	函数名	简要说明
基本函数	BLE_LibInit	库初始化

	TMOS_RandRegister	注册产生随机值种子函数
	TMOS_Rc32KRegister	注册内部 32K 时钟校准函数
	TMOS_SleepRegister	注册执行睡眠唤醒函数
	TMOS_SystemProcess	执行系统处理
GAP 函数	GAP_GetParamValue	读 GAP 参数
	GAP_SetParamValue	写 GAP 参数
	GAPRole_GetParameter	读 GAP Role 参数
	GAPRole_SetParameter	写 GAP Role 参数
	GGG_GetParameter	读 GAP 服务参数
	GGG_SetParameter	写 GAP 服务参数
	GAPBondMgr_GetParameter	读绑定相关参数
	GAPBondMgr_SetParameter	写绑定相关参数
GATT 函数	GATTServApp_InitCharCfg	初始化特性配置值
	GATTServApp_ReadCharCfg	读特性配置值
	GATTServApp_RegisterService	注册一个 GAAT 属性列表并添加该服务的读写回调函数
	pfnGATTReadAttrCB_t	读属性值回调函数类型定义
	pfnGATTWriteAttrCB_t	写属性值回调函数类型定义
	GATT_Notification	主动通知函数

### 3.2 BLE\_LibInit

函数原型	u32 BLE_LibInit( bleConfig_t* pCfg )
输入	pCfg: 库配置结构体指针
输出	无
返回	0: 成功, 非 0: 错误
作用	初始化蓝牙库, 并配置功能, 必须在使用其他库函数之前调用

### 3.3 TMOS\_RandRegister

函数原型	void TMOS_RandRegister( pfnSrandCB srandCB )
输入	随机值种子子程序
输出	无
返回	无
作用	注册一个产生随机值种子函数, 建议在 LL_Init 函数之前调用

### 3.4 TMOS\_SleepRegister

函数原型	void TMOS_Rc32KRegister( pfnSleepCB sleepCB )
输入	
输出	无

返回	无
作用	开启睡眠功能时调用，注册一个睡眠唤醒回调函数

### 3.5 TMOS\_Rc32KRegister

函数原型	void TMOS_Rc32KRegister( pfnHSECalibrationCB rcCB )
输入	内部 32K 时钟校准子程序
输出	无
返回	无
作用	使用内部 32K RC 时调用，注册一个 32K RC 校准函数，用于校准时钟

### 3.6 TMOS\_SystemProcess

函数原型	void TMOS_SystemProcess( void )
输入	无
输出	无
返回	无
作用	执行系统处理

### 3.7 GAP\_GetParamValue

函数原型	uint16 GAP_GetParamValue( uint16 paramID )
输入	paramID: 需要读的 GAP 参数的索引值
输出	无
返回	参数值
作用	获取当前 GAP 参数值

### 3.8 GAP\_SetParamValue

函数原型	bStatus_t GAP_SetParamValue( uint16 paramID, u16 paramValue )
输入	paramID: 需要写的 GAP 参数的索引值 paramValue: 写入的值
输出	无
返回	0: 成功, 2: 参数无效
作用	写 GAP 参数值

### 3.9 GAPRole\_GetParameter

函数原型	bStatus_t GAPRole_GetParameter( uint16 param, void *pValue )
输入	paramID: 需要读的 GAP 参数的索引值 pValue: 数组指针

输出	获取的 GAP Role 参数
返回	0: 成功, 2: 参数无效
作用	获取当前 GAP Role 参数值

### 3.10 GAPRole\_SetParameter

函数原型	bStatus_t GAPRole_SetParameter( uint16 param, uint8 len, void *pValue )
输入	paramID: 需要写的 GAP 参数的索引值 len: 写入参数长度 pValue: 写入参数值
输出	无
返回	0: 成功, 2: 参数无效
作用	写 GAP Role 参数值

### 3.11 GGS\_GetParameter

函数原型	bStatus_t GGS_GetParameter( uint8 param, void *value )
输入	paramID: 需要读的 GAP 参数的索引值 value: 数组指针
输出	获取的 GAP 服务参数
返回	0: 成功, 2: 参数无效
作用	获取当前 GAP 服务参数值

### 3.12 GGS\_SetParameter

函数原型	bStatus_t GGS_SetParameter( uint8 param, uint8 len, void *value )
输入	paramID: 需要写的 GAP 参数的索引值 len: 写入参数长度 value: 写入参数值
输出	无
返回	0: 成功, 2: 参数无效
作用	写 GAP 服务参数值

### 3.13 GAPBondMgr\_GetParameter

函数原型	bStatus_t GAPBondMgr_GetParameter( uint16 param, void *pValue )
输入	paramID: 需要读的 GAP 参数的索引值 pValue: 数组指针
输出	获取的 GAP 绑定参数
返回	0: 成功, 2: 参数无效
作用	获取当前 GAP 绑定参数值

3.14 GAPBondMgr\_SetParameter

函数原型	bStatus_t GGS_SetParameter( uint8 param, uint8 len, void *value )
输入	paramID: 需要写的 GAP 参数的索引值 len: 写入参数长度 value: 写入参数值
输出	无
返回	0: 成功, 2: 参数无效
作用	写 GAP 绑定参数值

3.15 GATTServApp\_InitCharCfg

函数原型	void GATTServApp_InitCharCfg( uint16 connHandle, gattCharCfg_t *charCfgTbl )
输入	connHandle: 连接句柄. charCfgTbl: 特性值列表
输出	无
返回	无
作用	初始化特性值置值

3.16 GATTServApp\_ReadCharCfg

函数原型	uint16 GATTServApp_ReadCharCfg( uint16 connHandle, gattCharCfg_t *charCfgTbl )
输入	connHandle: 连接句柄. charCfgTbl: 特性值列表
输出	无
返回	特性值
作用	读取特性值

3.17 GATTServApp\_RegisterService

函数原型	bStatus_t GATTServApp_RegisterService( gattAttribute_t *pAttrs, uint16 numAttrs, uint8 encKeySize, gattServiceCBs_t *pServiceCBs )
输入	pAttrs: 要注册的属性表 numAttrs: 属性数量 encKeySize: 加密 KEY 最小值 pServiceCBs: 服务回调函数指针
输出	无
返回	0: 成功, 非 0: 错误
作用	注册一个 GAAT 属性列表并添加该服务的读写回调函数

GATTServApp\_RegisterService 函数内部会分配一个 gattService\_t 结构体, 并且把 pAttrs,

numAttrs, encKeySize 这 3 个参数赋值给 gattService\_t 结构体, CH579 协议栈会把所有 gattService\_t 结构体链成一个列表。所有服务都是有 gattAttribute\_t 数组组成, gattServiceCBs\_t 结构体中的 pfnReadAttrCB, pfnWriteAttrCB 回调函数提供对当前服务的读写操作, 如果属性权限中定义 GATT\_PERMIT\_AUTHOR\_READ 或者 GATT\_PERMIT\_AUTHOR\_WRITE, 协议栈会调用 pfnAuthorizeAttrCB 回调函数来处理, 只有 pfnAuthorizeAttrCB 返回成功后, 才会允许读写当前属性。

### 3.18 pfnGATTReadAttrCB\_t

函数类型	typedef uint8 (*pfnGATTReadAttrCB_t)( uint16 connHandle, gattAttribute_t *pAttr, uint8 *pValue, uint16 *pLen, uint16 offset, uint16 maxLen, uint8 method );
输入	connHandle: 连接句柄, 表示不同的连接; pAttr: 指向被读取的属性 pValue: 指向被读取的属性值 pLen: 函数返回后, 实际被读取的数据长度 Offset: 被读取的属性值的偏移, 如果属性长度超过 ATT_MTU-1 时, 用块读取 (Read Blob) 才会用到 Offset, 其他情况都是 0 maxLen: pValue 缓冲区的最大长度 Method: 读属性时所用的方法
输出	读取的数据内容和长度
返回	0: 成功, 非 0: 错误
作用	读属性值回调

### 3.19 pfnGATTWriteAttrCB\_t

函数类型	typedef uint8 (*pfnGATTWriteAttrCB_t)( uint16 connHandle, gattAttribute_t *pAttr, uint8 *pValue, uint16 len, uint16 offset, uint8 method );
输入	connHandle: 连接句柄, 表示不同的连接 pAttr: 指向被写入的属性 pValue: 指向被写入的属性值 Len: 被写入的属性值长度 Offset: 被写入的属性值的偏移 Method: 写属性时所用的方法
输出	无
返回	0: 成功, 非 0: 错误
作用	写属性值回调

### 3.20 GATT\_Notification



函数原型	bStatus_t GATT_Notification( uint16 connHandle, attHandleValueNoti_t *pNoti, uint8 authenticated )
输入	connHandle: 连接句柄, 表示不同的连接 pNoti: 通知信息 authenticated: 是否需要认证连接
输出	无
返回	0: 成功, 非 0: 错误
作用	用于服务端主动发送数据包, 并且不需要应答

4、使用指南

4.1 库配置步骤

- 1、调用基本函数初始化全局变量,注册随机值回调函数,根据配置选择是否注册内部 32K RC 校准、睡眠唤醒等回调函数;
  - 2、可调用 GAP 函数配置广播参数,连接相关参数,设备名称,绑定相关参数;不配置则使用协议栈库的默认值;
  - 3、调用 GAP 函数配置 GAP 角色;
  - 4、如果是连接通信,配置 GATT 参数,注册 GATT 服务和读写回调函数;
  - 5、启动设备进入广播状态,等待连接;
  - 6、建立连接后,协议栈库通过回调应用层函数自动完成枚举过程;
  - 7、应用层通过读属性值回调和写属性值回调来收发数据,或使用通知方式主动发送数据。
- 注:初始化蓝牙协议栈库后,RTC 被占用,应用不可以再初始化 RTC,只可以读取当前 RTC 计数值,配置 RTC 中断操作。

4.2 系统管理

协议栈库提供了丰富的系统管理 API 接口,主要包括任务管理和消息管理,系统的单位时间是 625uS。

任务管理——多任务管理方式可以实现数据共享,合理分配各任务来提高效率,多任务运行实际上只有一个任务在运行,但是可以使用任务调度的策略将多个任务进行调度,每个任务占用一定的时间,所有的任务通过时间分片方式处理。

消息管理——消息是一个带有数据的事件,主要用于协议栈库各层之间传递数据,支持同时添加多个消息。

消息和任务的区别:

任务是通过事件来触发,然后执行一个相应的操作,这个操作一般是调用其他子程序来处理一些事情,通过设置事件的触发时间来合理安排每一个任务。

消息也是通过事件来触发,只是它带有数据,一般只需要处理这些数据,或者根据这些数据作进一步的分析处理,消息的实时性较高,都是立即触发的。

4.3 属性、特性和服务的关系

一个配置文件最少要包含一个服务,一个服务必须包含一个服务声明,这里可以是主服务或者次服务。可能包含一个或者多个引用声明,但至少包含一个特性。

4.3.1 属性

- (1) 属性句柄: 一个索引值,可以用来区别不同的属性,句柄值有效范围是 0x0001~0xFFFF.
- (2) 属性类型: 2 字节或者 16 字节的 UUID.
- (3) 属性值: 属性值和长度是根据属性类型来确定的。例如: 属性类型是 0x2800,属性值就是一个服务的 UUID; 如果属性类型是自定义的 0xFFFF0,属性值就是应用层自定义的数据。



(4) 特性用户描述声明：

属性句柄	属性类型	属性值	属性许可
0xNNNN	0x2901 《特性用户描述》	UTF-8 字符串	由上层定义

(5) 客户端特性配置声明：

属性句柄	属性类型	属性值	属性许可
0xNNNN	0x2902 《客户端特性配置》	特性配置位	由上层定义

特性配置位：

- 0x0001：通知。服务器可以主动通知属性值给客户端。
- 0x0002：指示。服务器将属性值指示给客户端，并得到客户端的确认。

4.3.3 服务

(1) 服务声明：

- 主要服务声明属性类型是 0x2800；
- 次要服务声明属性类型是 0x2801；
- 属性值是 2 字节或 16 字节的 UUID。

属性句柄	属性类型	属性值	属性许可
0xNNNN	0x2800 《首要服务》， 或 0x2801 《次要服务》	16 或 128 位 服务 UUID	可读

(2) 引用声明：

- 引用声明必须在服务声明之后，特性声明之前；
- 属性类型被蓝牙联盟组织分配的 UUID 是 0x2802；
- 属性值是 2 字节或 16 字节的 UUID；
- 服务的引用声明不能形成环型，如：服务 A 中定义了一个引用声明引用到服务 B，服务 B 中也定义一个引用声明引用到服务 A。

属性句柄	属性类型	属性值			属性许可
0xNNNN	0x2802 《引用》	引用服务的 属性句柄	结束句柄	服务 UUID	可读