**GATT**

**1 初始化**

**bt\_gatt\_init()**：注册强制要求的服务（gap\_svc、gatt\_svc），初始化服务改变处理（sc\_process）延迟工作项。

sc\_process()：由工作项获取SC，得到SC服务（sc\_range），清除SC范围改变标志，设置sc的指示参数（sc->params），发送GATT指示（bt\_gatt\_indicate），设置SC指示待处理标志（PENDING）。指示发送后会调用指示响应函数（sc\_indicate\_rsp）来处理指示。

sc\_indicate\_rsp()：清除SC指示待处理标志，提交服务改变工作项。

**2 客户端**

**注意**：客户端读、写或者订阅的基本流程都是一样的。首先，客户端发送请求（读、写等）到服务器，服务器根据请求类型调用不同的处理函数。在处理函数中，会调用用户在声明服务时为各属性注册的读/写函数，之后发送响应（读、写等）。客户端收到响应后，会调用用户设置的参数（读、写等参数）中的回调函数。

**2.1 发现服务**

**bt\_gatt\_discover()**：检查参数是否有效。若连接状态不为已连接，则返回错误。若参数类型（params->type）为发现主/次服务（PRIMARY/SECONDARY）且参数的uuid不为空，则调用查找类型函数（gatt\_find\_type），否则读取组函数（gatt\_read\_group）；若参数类型为发现包含服务（INCLUDE）或特征（CHARACTERISTIC），则调用读取类型函数（gatt\_read\_type）；若参数类型为发现描述符（DESCRIPTOR），则调用查找信息函数（gatt\_find\_info）。

**gatt\_find\_type()**：创建查找类型请求（FIND\_TYPE\_REQ）数据包，设置请求（起始句柄、类型、UUID），发送请求（gatt\_send）并注册查找类型响应函数（gatt\_find\_type\_rsp）。

**gatt\_read\_group()**：创建读取组请求（READ\_GROUP\_REQ）数据包，设置请求（起始句柄、组类型），发送请求（gatt\_send）并注册响应函数（gatt\_read\_group\_rsp）。

**gatt\_read\_group\_rsp()**：若发送错误，则调用回调函数并传递空值。解析服务（parse\_service），若结束句柄（handle）为0，则返回，否则继续发现服务（gatt\_discover\_next）。

**parse\_service()**：若响应长度（rsp->len）等于6，说明是16位的UUID（见Spec Vol3 PartF 3.4.4.10）；若长度等于20，说明是128位的UUID；否则错误。遍历响应数据（rsp->data），获取起始句柄和UUID，并将结果保存到属性中（attr），调用回调函数（params->func）。若数据读取完毕且没有错误，则返回结束句柄（end\_handle）用于继续发现。

注：服务数据由属性句柄，组结束句柄和UUID组成。

**gatt\_discover\_next()**：若开始句柄（start\_handle）等于0xffff，说明发现完成，跳转到完成（done）。完成：调用回调函数并传入空属性（NULL），在回调函数中继续发现下一级服务。

**gatt\_read\_type()**：创建读类型请求（READ\_TYPE\_REQ）数据包，设置请求参数（起始句柄、类型），发送请求（gatt\_send）并注册响应函数（gatt\_read\_type\_rsp）。

**gatt\_read\_type\_rsp()**：若发生错误，则调用回调函数并传入空属性。若参数类型为发现包含（DISCOVER\_INCLUDE），则解析包含（parse\_include），否则解析特征（parse\_characteristic）。

**parse\_characteristic()**：和解析服务过程类似。

注：特征数据由属性句柄和特征值（包括权限、值句柄和UUID）

**gatt\_find\_info()**：创建查找信息请求（FIND\_INFO\_REQ）数据包，设置请求参数（起始句柄），发生请求（gatt\_send）并注册响应函数（gatt\_find\_info\_rsp）。

**gatt\_find\_info\_rsp()**：根据响应格式判断数据类型（UUID16还是UUID128），解析描述符。

注：描述符数据由句柄和UUID组成。

**注意**：这里的响应函数是指发送请求并收到相应的响应后所调用的函数。这里的回调函数是指用户在发现服务时在发现参数中设置的回调函数，由用户实现。

**2.2 读操作**

**bt\_gatt\_read()**：检查参数（params）是否有效。若连接状态非已连接，则返回错误。若句柄数量（params->handle\_count）大于1，则调用读多属性函数（gatt\_read\_multiple）。若单句柄的偏移量(（params->single.offset）不为空，则调用读数据块函数（gatt\_read\_blob）。创建读请求数据包（READ\_REQ），设置句柄，发送请求（gatt\_send）并注册读响应函数（gatt\_read\_rsp）。

**gatt\_send()**：若参数不为空，则设置请求的缓冲（req->buf）、回调函数（req->func）和破坏函数（req->destroy），发送请求（bt\_att\_req\_send），否则直接发送数据（bt\_att\_send）。

gatt\_read\_rsp()：若发送错误或数据长度为0，则调用读参数的回调函数（params->func，由用户实现）且传入的数据为空。否则调用回调函数并传入读取到的数据（pdu）。若数据长度小于ATT\_MTU-1，则说明已经读取完成。否则设置读请求参数的偏移量（params->single.offset）然后继续读属性（bt\_gatt\_read）。

**2.3 写操作**

bt\_gatt\_write()：检查参数（params）是否有效。若参数的偏移量（offset）不为0或长度（length）大于MTU，则调用预写函数（gatt\_prepare\_write）。否则创建读请求（WRITE\_REQ）数据包，设置句柄（req->handle）及写入的数据（req->value），发送请求（gatt\_send）并注册写响应函数（gatt\_write\_rsp）。

gatt\_write\_rsp()：调用写参数的回调函数（params->func）。

**2.4 订阅**

**bt\_gatt\_subscribe()**：检查参数（params）是否有效。根据连接将订阅（sub）添加到订阅数组（subscriptions）。遍历订阅的参数链表（sub->list），若参数已存在，则将此订阅移除。若存在已订阅的参数，则返回，否则写客户端特征配置（gatt\_write\_ccc），

gatt\_write\_ccc()：

bt\_gatt\_attr\_write\_ccc()：获取写入值（value）。遍历配置，根据连接地址（conn->le.dst）检查是否有已存在的配置（ccc->cfg[i].peer）。若配置不存在，则再次遍历配置数组，查找未使用的配置，设置配置的地址（peer）并设置有效标志（valid）。设置配置值（value），若配置值发送了改变，则调用配置改变函数（gatt\_ccc\_changed）。

gatt\_ccc\_changed()：若配置值已改变，调用配置改变回调函数（用户声明服务时注册，ccc->cfg\_changed）。

**2.5 通知处理**

bt\_gatt\_notification()：遍历订阅链表（subscriptions），根据连接和属性句柄找到对应的订阅参数（params），调用参数的通知回调函数（params->notify）。

**3 服务器**

**3.1 注册服务**

bt\_gatt\_service\_register()：检查参数是否有效。不允许再次注册强制性服务。注册服务（gatt\_register），发送服务改变指示（sc\_indicate）。

gatt\_register()：若服务链表（db）为空，则设置尾句柄（handle）为0，转到插入。否则找到服务链表的尾节点（last），将句柄设置为尾节点的最后一个属性的句柄。插入：逐个设置属性的句柄并将其添加到服务链表尾部（sys\_slist\_append）。

**3.2 读属性**

**bt\_gatt\_attr\_read\_service()**：获取uuid，调用读属性函数（bt\_gatt\_attr\_read）。

注意：用户在声明主/次服务时（BT\_GATT\_PRIMARY\_SERVICE），会为属性注册一个读函数（bt\_gatt\_attr\_read\_service）。当客户端发起读组请求（Read by Group Type Request）时，服务器会发送响应，在生成响应数据时就会调用此函数来读取属性值（即服务的UUID）。

**bt\_gatt\_attr\_read\_chrc()**：获取特性（pdu），调用读属性函数。

注：与读服务函数类似，用于客户端发起读类型请求（Read By Type Request）时的响应。

**bt\_gatt\_attr\_read()**：将属性值（value）复制到缓冲（buf）中。

**3.3 发送通知**

**bt\_gatt\_notify()**：若连接不为空，则发送通知到对应连接（gatt\_notify），否则设置通知数据（nfy），遍历每个属性，根据属性句柄找到对应属性并调用通知回调函数（notify\_cb）。

**gatt\_notify()**：创建通知数据包，设置句柄（nfy->handle）和通知数据（nfy->value），发送通知。

**notify\_cb()**：

**3.4 发送指示**

**bt\_gatt\_indicate()**：若连接不为空（即指定了连接），则发送指示到对应连接（gatt\_indicate），否则设置通知数据（nfy），遍历每个属性，根据属性句柄找到对应属性并调用通知回调函数（notify\_cb）。

**gatt\_indicate()**：

**gatt\_indicate\_rsp()**：调用指示参数中的回调函数（params->func，由用户实现）。

注意：通知和指示的区别在于，通知只是将数据发送到客户端，而不在乎客户端是否收到数据。指示将数据发送到客户端，但要求客户端收到数据后发送确认信息到服务器，服务器再将结果传给用户注册的回调函数，这样用户就能知道客户端是否收到了指示。

