

新增 GATT Service 说明文档



OPULINKS

<http://www.opulinks.com/>

Copyright © 2020, Opulinks. All Rights Reserved.

新增 GATT Service 说明文档 | Vversion 1.0

版本纪录

日期	版本	更新内容
2020/04/17	1.0	● 初版

目录

1.	介绍	1
1.1.	文档应用说明	1
1.2.	参考文献	1
2.	代码说明	2

1. 介绍

1.1. 文档应用说明

以 blewifi example 为例，请参考在 blewifi_server_app_gatt.c^[1] 中的 blewifi service。

1.2. 参考文献

[1] blewifi_server_app_gatt.c

<https://github.com/Opulinks-Tech/OPL1000A2-lot->

[Solution/blob/master/Example/blewifi/src/blewifi_server_app_gatt.c](https://github.com/Opulinks-Tech/OPL1000A2-lot-Solution/blob/master/Example/blewifi/src/blewifi_server_app_gatt.c)

2. 代码说明

1. 宣告 service

```
static LE_GATT_ATTR_T gBwpSvcDb[BWP_IDX_TOP] =
{
    // BLEWIFI Service Declaration
    [BWP_IDX_SVC]      = PRIMARY_SERVICE_DECL_UUID16(&gBleWifiSvcUuid),
    // BLEWIFI Data In Characteristic
    [BWP_IDX_DATA_IN_CHAR] = CHARACTERISTIC_DECL_UUID16(gBleWifiDataInCharVal),
    [BWP_IDX_DATA_IN_VAL]  = CHARACTERISTIC_UUID16(&gBleWifiDataInUuid,
    LE_GATT_PERMIT_AUTHOR_WRITE, sizeof(gBleWifiDataInVal), 0, gBleWifiDataInVal),
    // BLEWIFI Data Out Characteristic
    [BWP_IDX_DATA_OUT_CHAR] = CHARACTERISTIC_DECL_UUID16(gBleWifiDataOutCharVal),
    [BWP_IDX_DATA_OUT_VAL]  = CHARACTERISTIC_UUID16(&gBleWifiDataOutUuid, 0,
    sizeof(gBleWifiDataOutVal), 0, gBleWifiDataOutVal),
    [BWP_IDX_DATA_OUT_CFG] =
    CHAR_CLIENT_CONFIG_DESCRIPTOR(LE_GATT_PERMIT_AUTHOR_READ |
    LE_GATT_PERMIT_AUTHOR_WRITE, &gBleWifiDataOutClientCfg)
};
```

2. 注册 service

```
static void BleWifi_Ble_ServerAppGattMsgHandler_InitCfm(TASK task, MESSAGEID id, MESSAGE
message)
{
    T_MwFim_GP11_BleServiceUUID tBleServiceUUID;
```

```

//!!! after LeGattInit

BLEWIFI_INFO("APP-LE_GATT_MSG_INIT_CFM\r\n");

// get the settings of BLE service UUID

if (MW_FIM_OK != MwFim_FileRead(MW_FIM_IDX_GP11_PROJECT_BLE_SERVICE_UUID, 0,
    MW_FIM_GP11_BLE_SERVICE_UUID_SIZE, (uint8_t*)&tBleServiceUUID))
{
    // if fail, get the default value

    memcpy(&tBleServiceUUID,&g_tMwFimDefaultGp11BleServiceUUID,
        MW_FIM_GP11_BLE_SERVICE_UUID_SIZE);
}

gBleWifiSvcUuid = tBleServiceUUID.uwServiceUUID;
gBleWifiDataInUuid = tBleServiceUUID.uwDataInUUID;
gBleWifiDataInCharVal[3] = UINT16_LO(gBleWifiDataInUuid);
gBleWifiDataInCharVal[4] = UINT16_HI(gBleWifiDataInUuid);
gBleWifiDataOutUuid = tBleServiceUUID.uwDataOutUUID;
gBleWifiDataOutCharVal[3] = UINT16_LO(gBleWifiDataOutUuid);
gBleWifiDataOutCharVal[4] = UINT16_HI(gBleWifiDataOutUuid);

gGattSvc=LeGattRegisterService(gGattSvcDb,sizeof(gGattSvcDb)/sizeof(LE_GATT_ATTR_T));
if (gGattSvc)
{ BLEWIFI_INFO("APP-LeGattRegisterService gGattSvc success\r\n"); }

gGapSvc= LeGattRegisterService(gGapSvcDb, sizeof(gGapSvcDb) / sizeof(LE_GATT_ATTR_T));
if (gGapSvc)
{ BLEWIFI_INFO("APP-LeGattRegisterService gGapSvc success\r\n"); }

// 注册 service

```

```

gBwpSvc= LeGattRegisterService(gBwpSvcDb, sizeof(gBwpSvcDb) / sizeof(LE_GATT_ATTR_T));
if (gBwpSvc)
{
    BLEWIFI_INFO("APP-LeGattRegisterService gBwpSvc success\r\n");
    BleWifi_Ble_GetEntity()->store.send_hdl = gBwpSvcDb[BWP_IDX_DATA_OUT_VAL].handle;
}

```

3. read 的处理

```

static void BleWifi_Ble_GattHandleAccessRead(LE_GATT_MSG_ACCESS_READ_IND_T *ind)
{
    if ((ind->handle >= gGattSvc->startHdl) && (ind->handle <= gGattSvc->endHdl))
    {
        BleWifi_Ble_AppHandleGattServiceRead(ind);
    }
    else if ((ind->handle >= gGapSvc->startHdl) && (ind->handle <= gGapSvc->endHdl))
    {
        BleWifi_Ble_AppHandleGapServiceRead(ind);
    }
    else if ((ind->handle >= gBwpSvc->startHdl) && (ind->handle <= gBwpSvc->endHdl))
    {
        BleWifi_Ble_AppHandleBwpServiceRead(ind); // 处理 read
    }
    else
    {
        LeGattAccessReadRsp(ind->conn_hdl, ind->handle, E_ATT_ERR_READ_NOT_PERMITTED);
    }
}

```

4. write 的处理

```

static void BleWifi_Ble_GattHandleAccessWrite(LE_GATT_MSG_ACCESS_WRITE_IND_T *ind)
{
    if ((ind->handle >= gGattSvc->startHdl) && (ind->handle <= gGattSvc->endHdl))

```

```

    {      BleWifi_Ble_AppHandleGattServiceWrite(ind);  }
    else if ((ind->handle >= gGapSvc->startHdl) && (ind->handle <= gGapSvc->endHdl))
    {      BleWifi_Ble_AppHandleGapServiceWrite(ind);  }
    else if ((ind->handle >= gBwpSvc->startHdl) && (ind->handle <= gBwpSvc->endHdl))
    {      BleWifi_Ble_AppHandleBwpServiceWrite(ind); // 处理 write  }
    else
    {
        LeGattAccessWriteRsp(ind->conn_hdl, ind->handle, E_ATT_ERR_WRITE_NOT_PERMITTED);
    }
}

```

5. notify 呼叫

```
LE_ERR_STATE LeGattCharValNotify(UINT16 conn_hdl, UINT16 hdl, UINT16 len, UINT8 *pval);
```

6. notify confirm 的处理

```

static void BleWifi_Ble_ServerAppGattMsgHandler_NotifyCfm(TASK task, MESSAGEID id, MESSAGE
message)
{
    // !!! after BLE data out (after LeGattCharValNotify)
    // trigger the next "send data from buffer to peer"
    BLEWIFI_INFO("APP-LE_GATT_MSG_NOTIFY_CFM\r\n");
    BleWifi_Ble_SendAppMsgToBle(BLEWIFI_APP_MSG_SEND_TO_PEER_CFM, 0, NULL);
}

```


CONTACT

sales@Opulinks.com