

## 在 Profiles 中添加特征值

### 前言：

上面对 SimpleGATTProfile 的几个特征值都做了介绍，接下来通过添加一个新的特征值来对这个属性有更深入的理解。

**实现平台：** WeBee CC2540 模块及功能底板

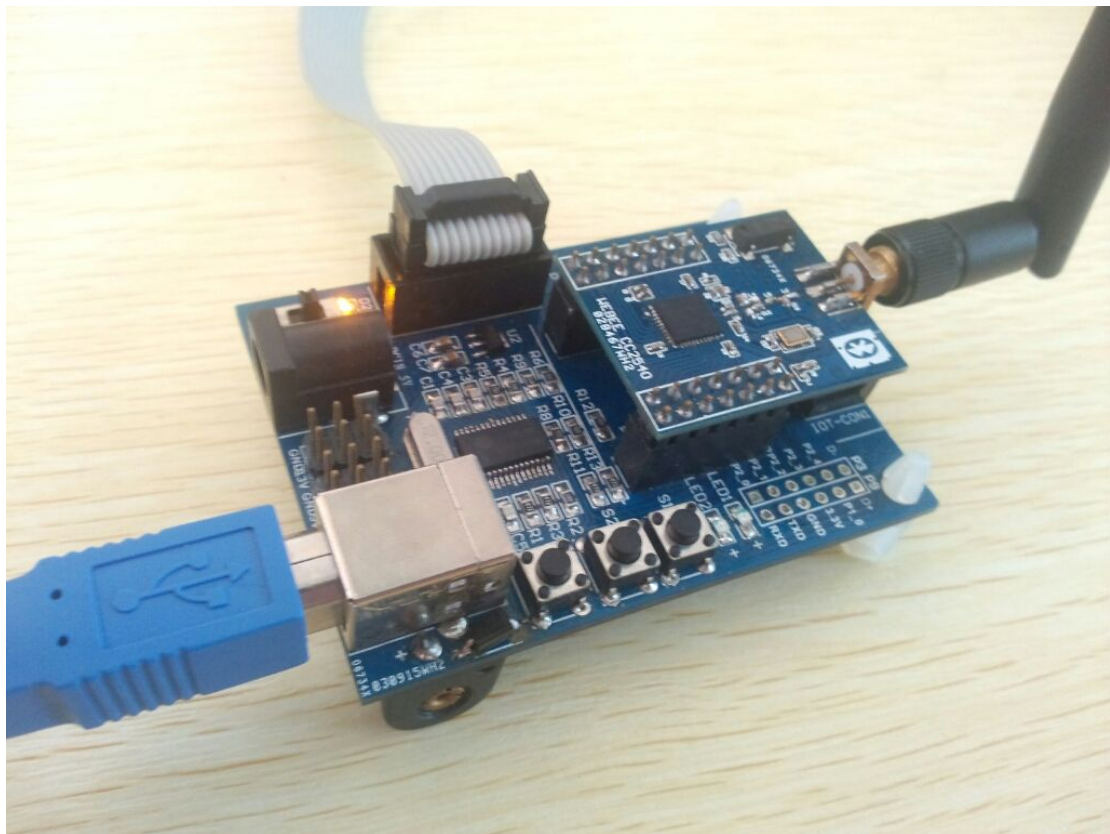


图 3.98 网峰蓝牙 4.0 开发套件

### 实验现象：

两个模块建立连接并利用 Btool 对添加的特征值进行读写，整个过程可通过串口观察到变化。这个实验在协议栈（BLE-CC254x-1.2.1）中进行。

### 实验讲解：

整个实验用到两个模块，分别为 SimpleBLEPeripheral 、HostTestRelease。

下面对主要几个文件进行修改

`simpleGATTprofile.h` 文件添加以下定义

```
#define SIMPLEPROFILE_CHAR6 5

#define SIMPLEPROFILE_CHAR6_UUID 0xFFFF6

#define SIMPLEPROFILE_CHAR6_LEN 5
```

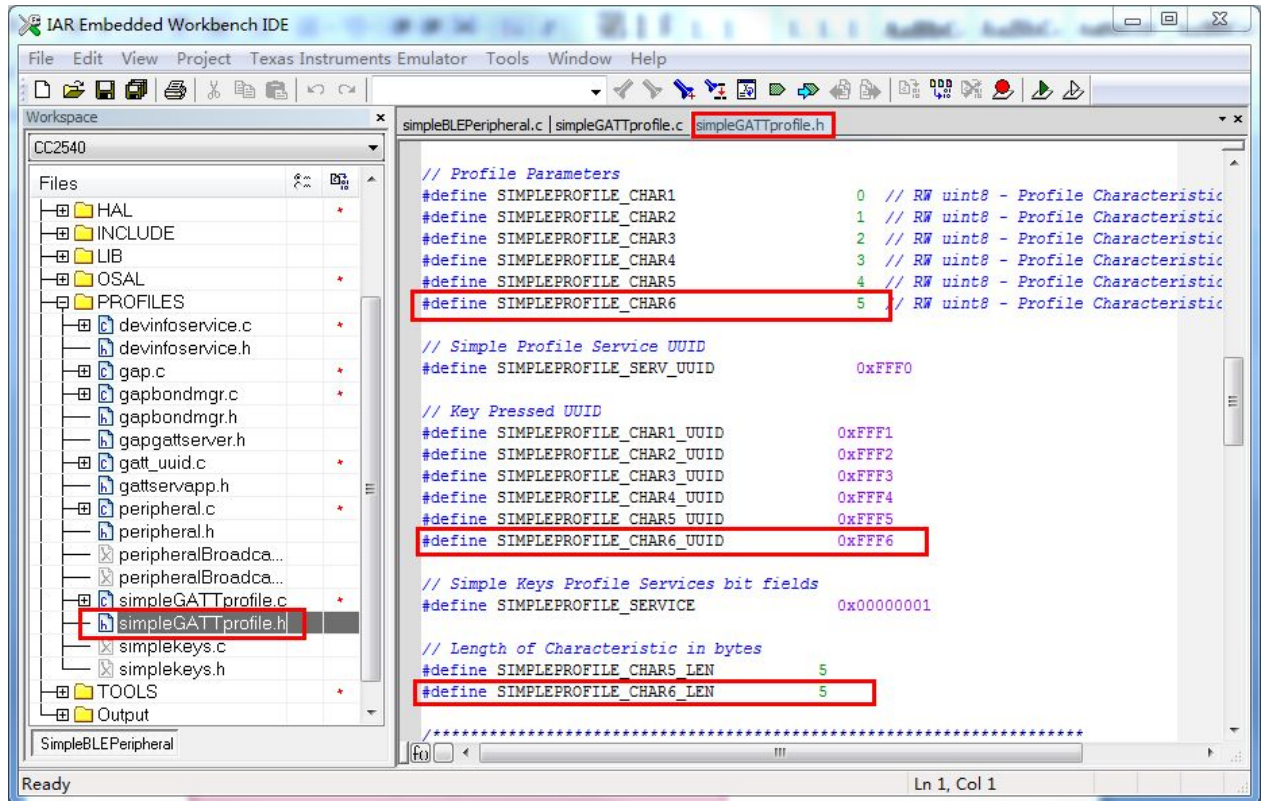


图 3.99 添加 CHAR6 特征值

`simpleGATTprofile.c`

### 1、添加特征值 UUID

```
// Characteristic 6 UUID: 0xFFFF6

CONST uint8 simpleProfilechar6UUID[ATT_BT_UUID_SIZE] =
{
    LO_UINT16(SIMPLEPROFILE_CHAR6_UUID),
    HI_UINT16(SIMPLEPROFILE_CHAR6_UUID)
};
```

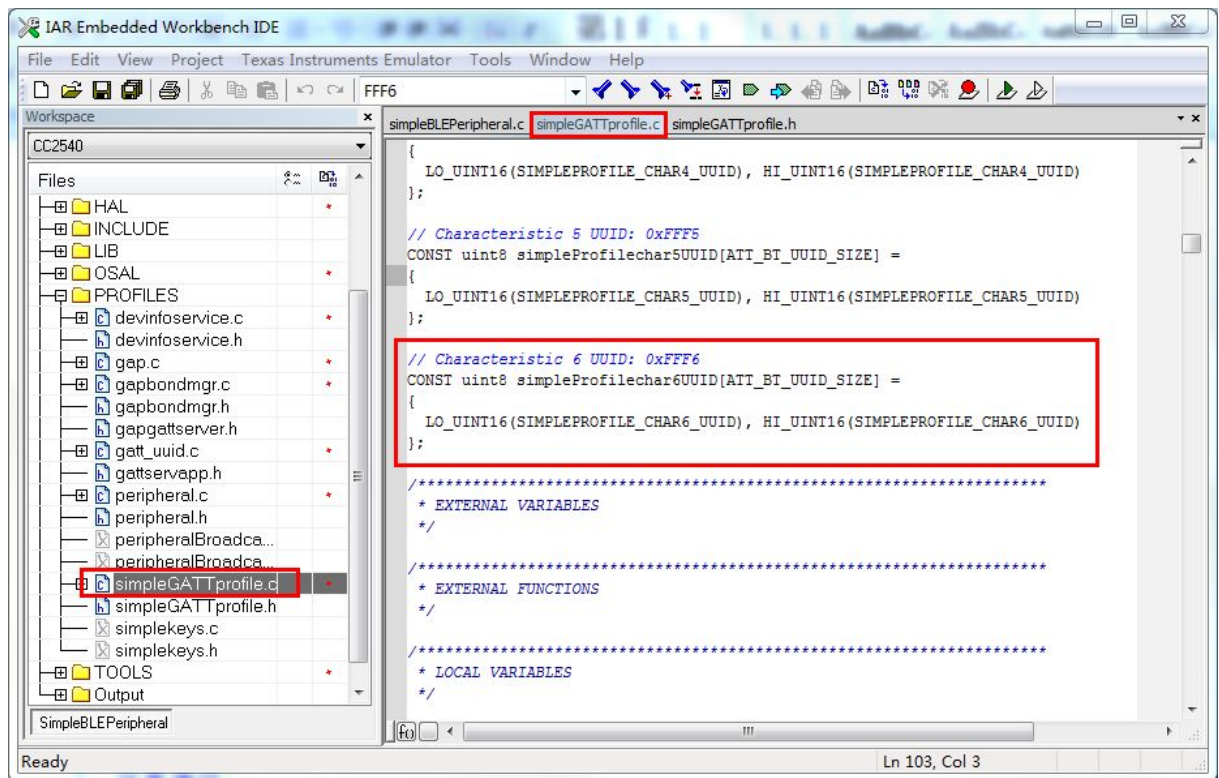


图 3.100 添加 UUID

## 2、设置属性

```
// Simple Profile Characteristic 6 Properties    可读可写
static uint8 simpleProfileChar6Props = GATT_PROP_READ |
GATT_PROP_WRITE;

// Characteristic 6 Value
static uint8 simpleProfileChar6[SIMPLEPROFILE_CHAR6_LEN] = { 0, 0, 0,
0, 0 };

// Simple Profile Characteristic 1 User Description
static uint8 simpleProfileChar6UserDesp[17] = "Characteristic 6\0";
```

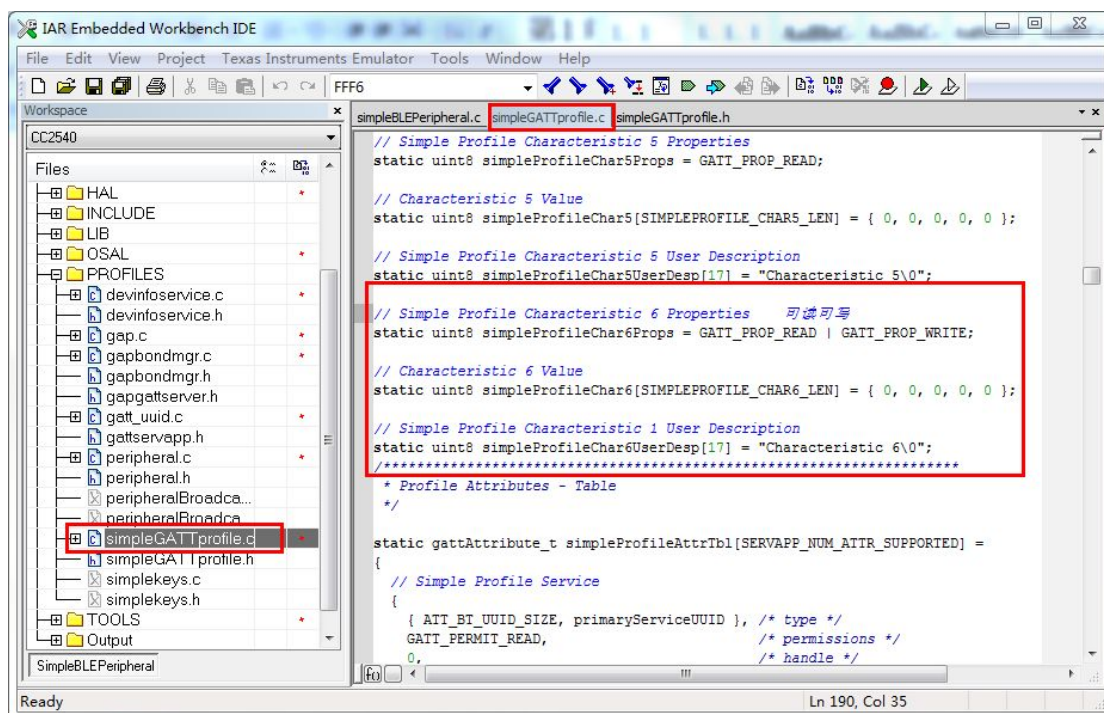


图 3.101 设置属性

### 3、属性表

static gattAttribute\_t simpleProfileAttrTbl[SERVAPP\_NUM\_ATTR\_SUPPORTED]

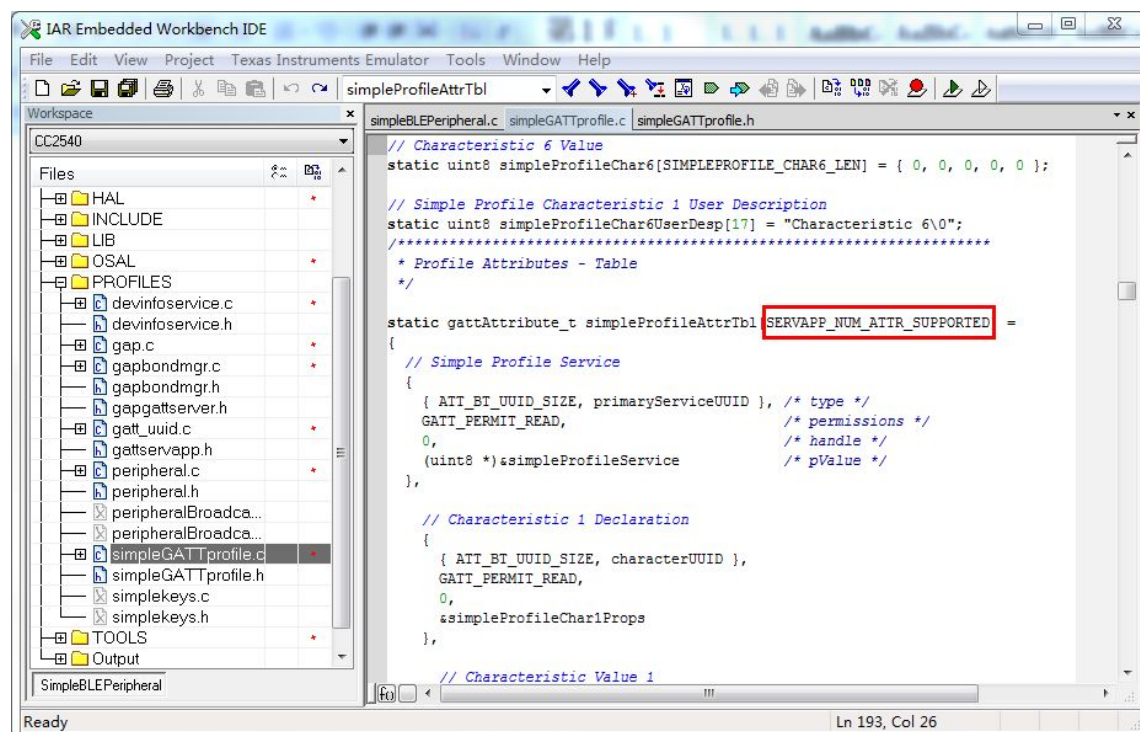


图 3.102 属性表

这里要把数组改为

```
#define SERVAPP_NUM_ATTR_SUPPORTED 20
```

并把 CHAR6 添加进去

```
// Characteristic Value 6
{
    { ATT_BT_UUID_SIZE, simpleProfilechar6UUID },
    GATT_PERMIT_READ | GATT_PERMIT_WRITE,
    0,
    simpleProfileChar6
},

// Characteristic 6 User Description
{
    { ATT_BT_UUID_SIZE, charUserDescUUID },
    GATT_PERMIT_READ,
    0,
    simpleProfileChar6UserDesp
},
```

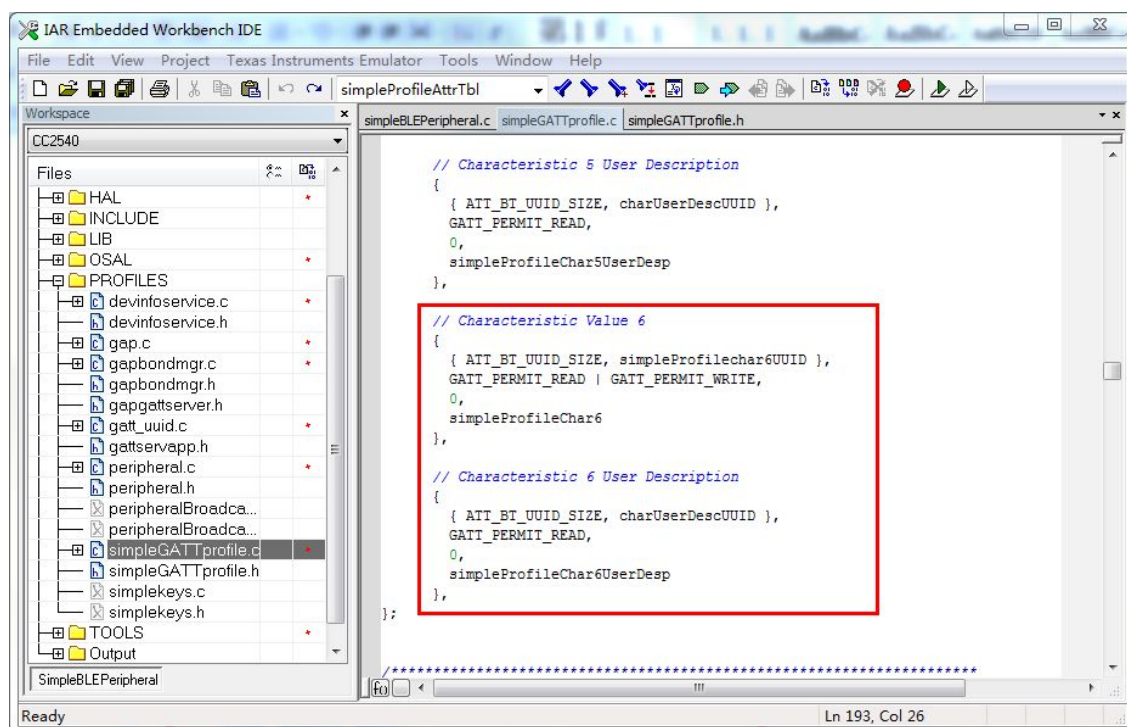


图 3.103 在属性表中添加 CHAR6

### 3、设置参数函数

bStatus\_t SimpleProfile\_SetParameter( uint8 param, uint8 len, void \*value )中:

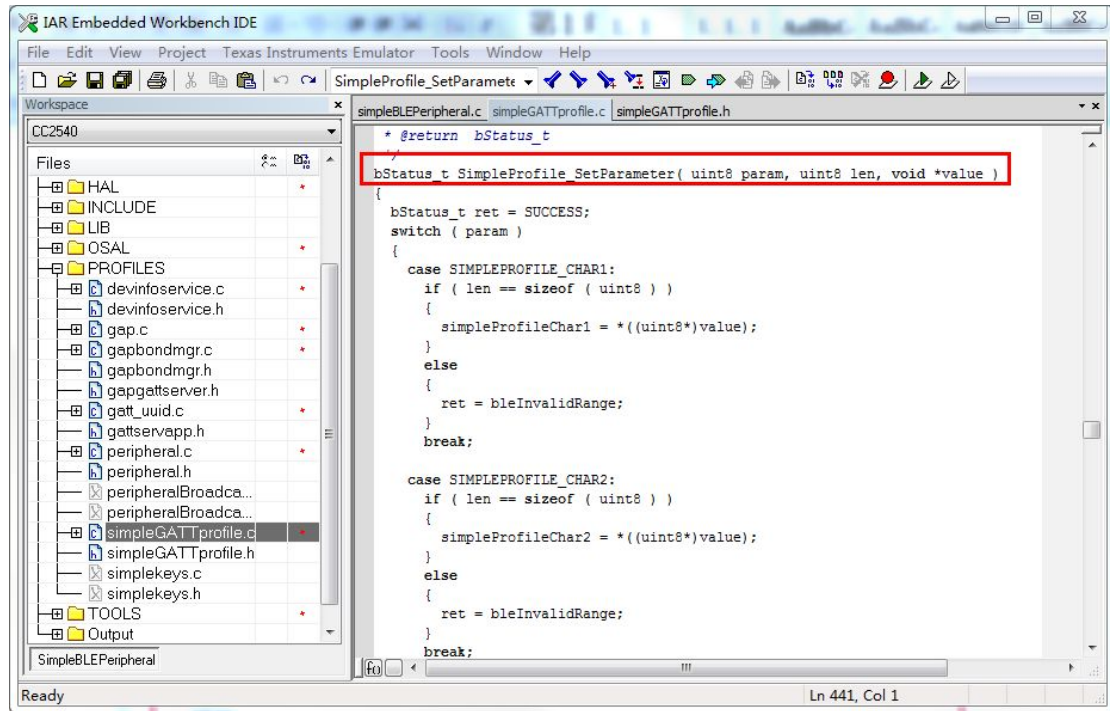


图 3.104 SimpleProfile\_SetParameter( uint8 param, uint8 len, void \*value )

添加以下代码:

```
case SIMPLEPROFILE_CHAR6:
    if ( len == SIMPLEPROFILE_CHAR6_LEN )
    {
        VOID osal_memcpy( simpleProfileChar6, value,
SIMPLEPROFILE_CHAR6_LEN );
    }
    else
    {
        ret = bleInvalidRange;
    }
    break;
```

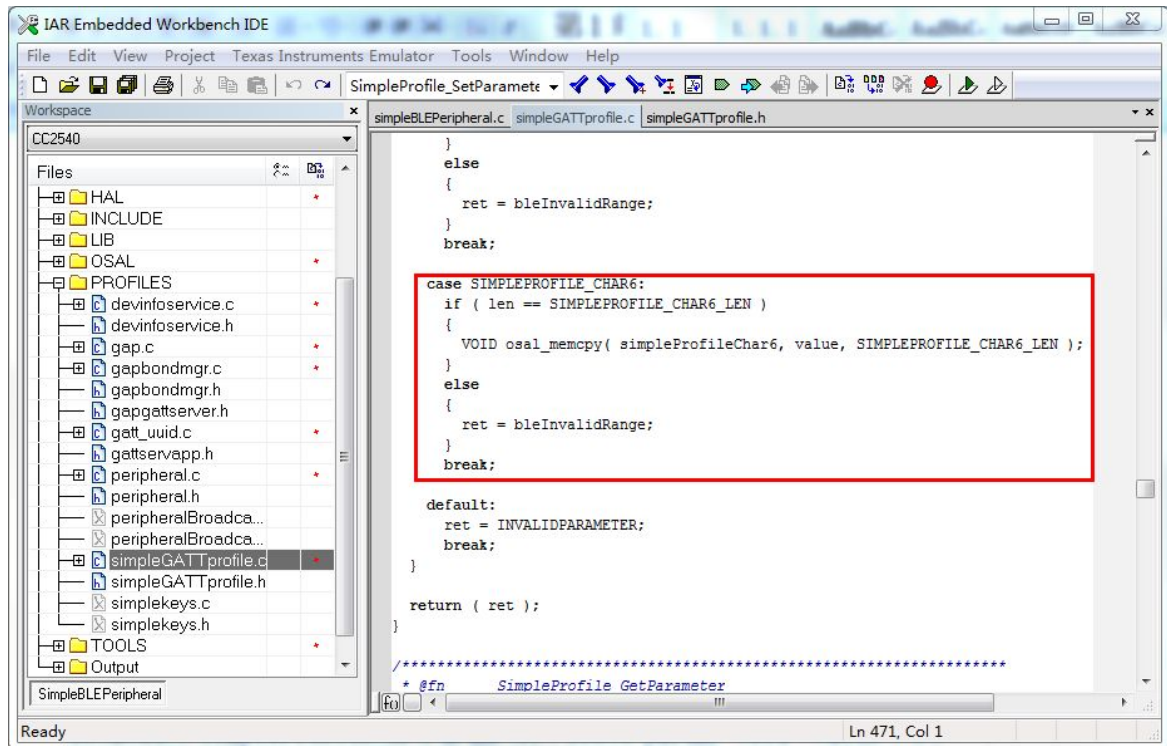


图 3.105 CHAR6 写入代码

bStatus\_t SimpleProfile\_GetParameter( uint8 param, void \*value )中添加:

```

case SIMPLEPROFILE_CHAR6:
    VOID      osal_memcpy(      value,      simpleProfileChar6,
SIMPLEPROFILE_CHAR6_LEN );
    break;

```

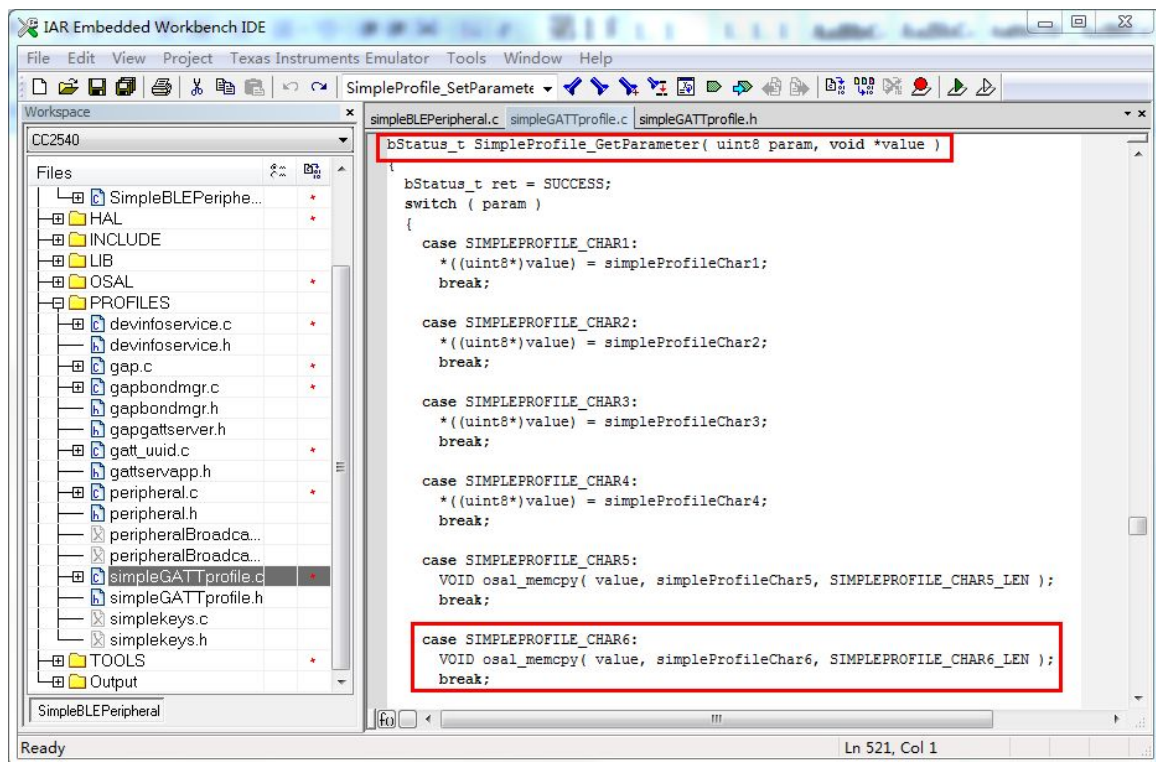


图 3.106 CHAR6 读取代码

#### 4、读写特征值函数

static uint8 simpleProfile\_ReadAttrCB( uint16 connHandle,  
 gattAttribute\_t \*pAttr,  
 uint8 \*pValue,  
 uint8 \*pLen,  
 uint16 offset,  
 uint8 maxLen )中:

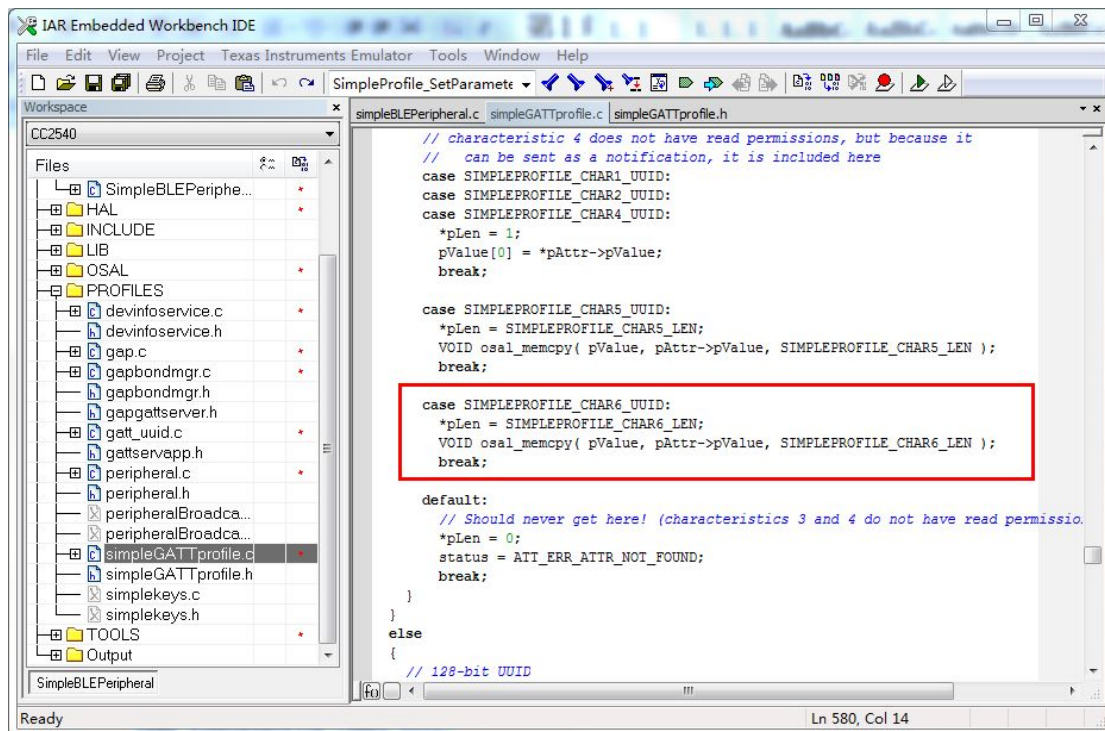


图 3.107 simpleProfile\_ReadAttrCB( ... )

```
static bStatus_t simpleProfile_WriteAttrCB( uint16 connHandle,
                                           gattAttribute_t *pAttr,
                                           uint8 *pValue, uint8 len,
                                           uint16 offset )中:
```

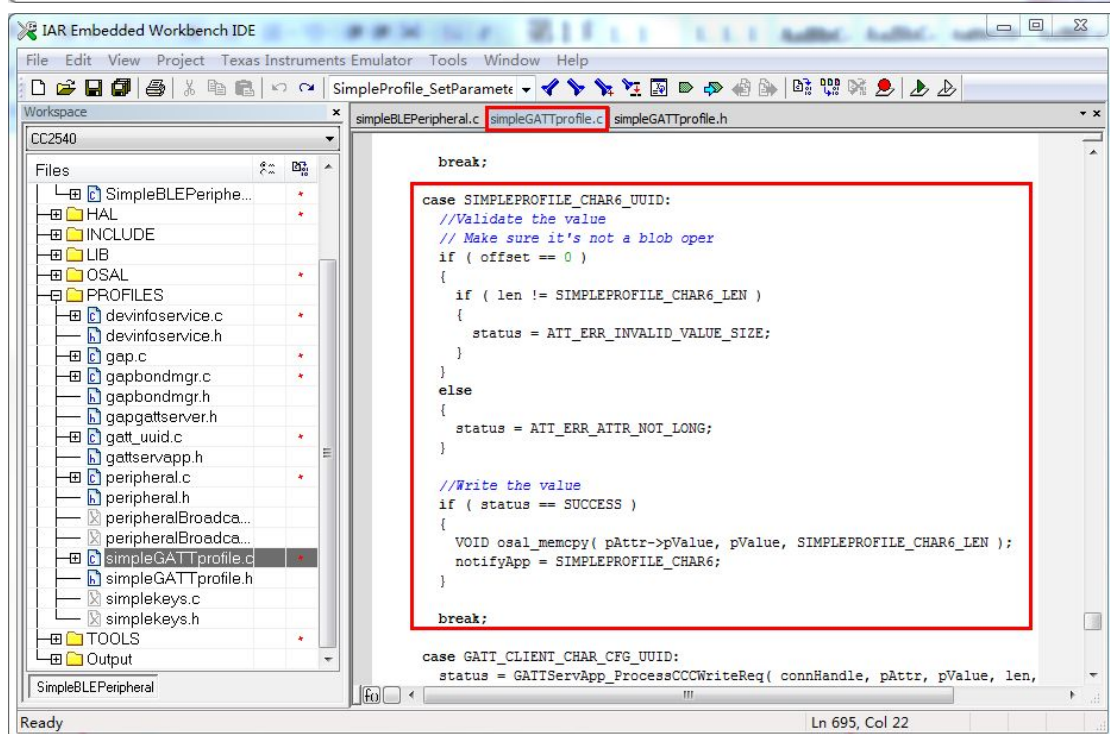
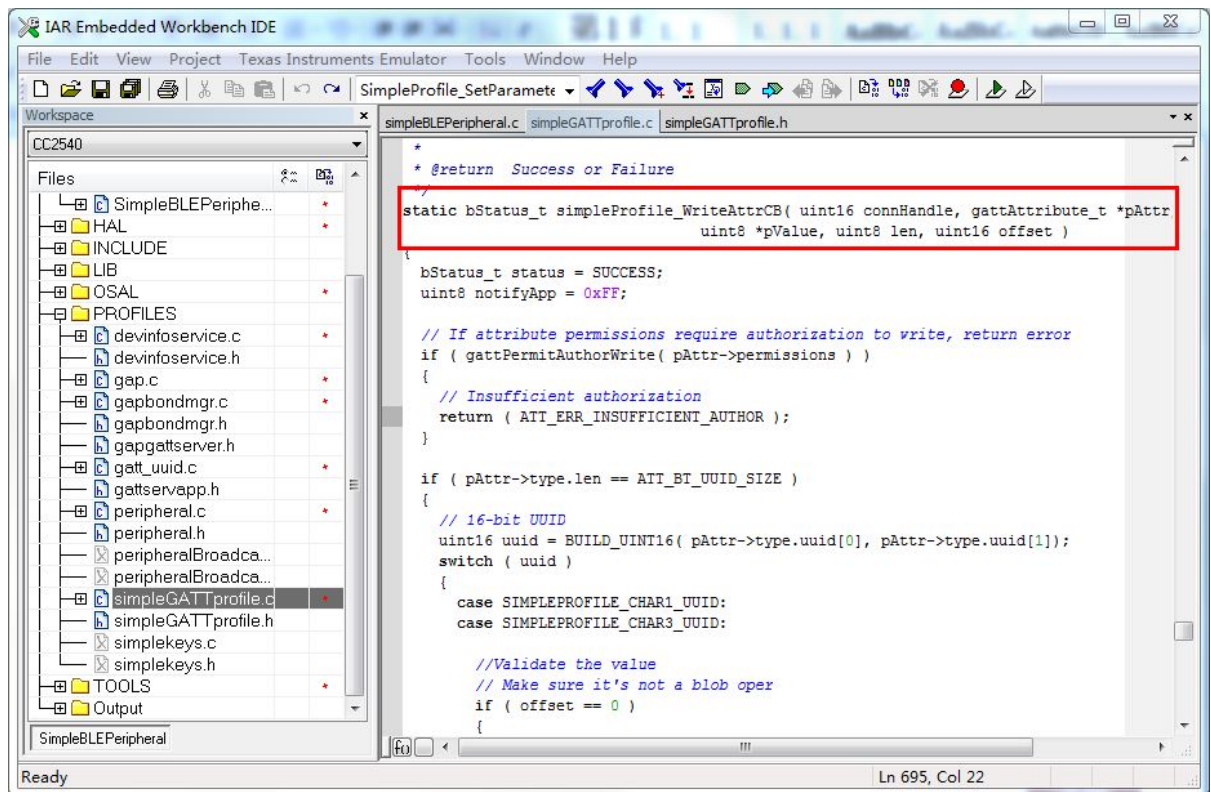


图 3.108 simpleProfile\_WriteAttrCB( ... )

simpleBLEPeripheral.c

在 void SimpleBLEPeripheral\_Init( uint8 task\_id ) 函数中初始化参数

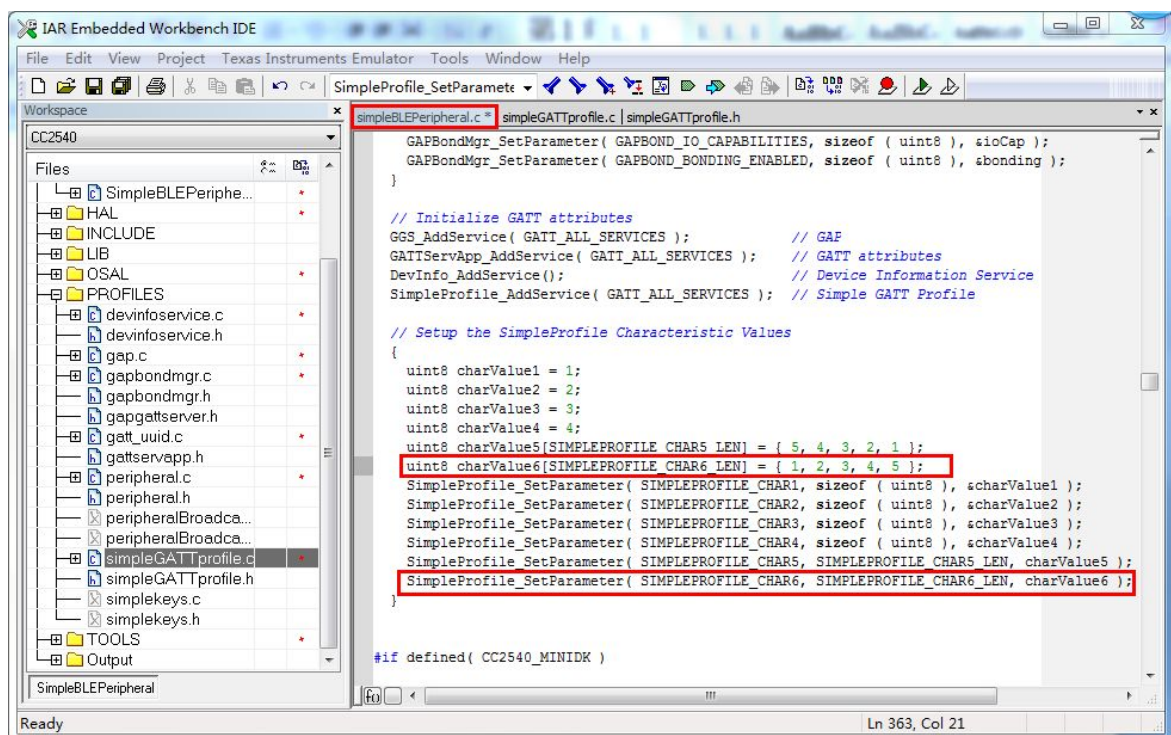


图 3.109 初始化 CHAR6 值

把 SimpleBLEPeripheral 工程下载到模块中，另外一个下载 HostTestRelease 工程。了解 PC，分别打开 BTool 和串口调试助手。建立连接，利用 UUID 读取特征值，此处的 UUID 为 0xffff6，可读到如下数据

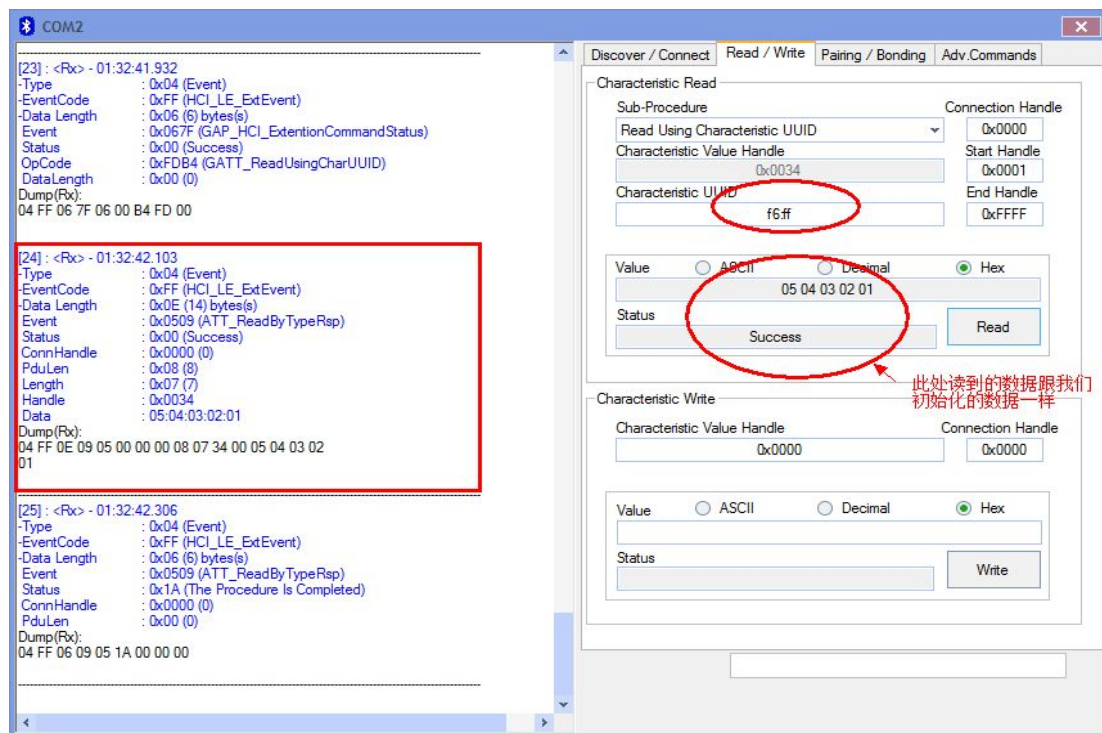


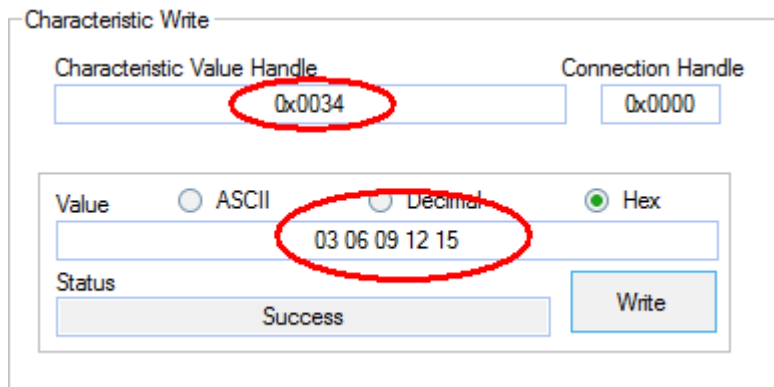
图 3.110 BTool 读取 CHAR6

从上面的数据可知 CHAR6 的 handle 为 0x0034

```
[44] : <Rx> - 01:39:54.160
-Type           : 0x04 (Event)
-EventCode      : 0xFF (HCI_LE_ExtEvent)
-Data Length    : 0x0E (14 bytes(s))
-Event          : 0x0509 (ATT_ReadByTypeRsp)
-Status         : 0x00 (Success)
-ConnHandle     : 0x0000 (0)
-PduLen        : 0x08 (8)
-Length         : 0x07 (7)
-Handle         : 0x0034
-Data           : 05:04:03:02:01
Dump(Rx):
04 FF 0E 09 05 00 00 00 08 07 34 00 05 04 03 02
01
```

图 3.111 CHAR6 Handle 值

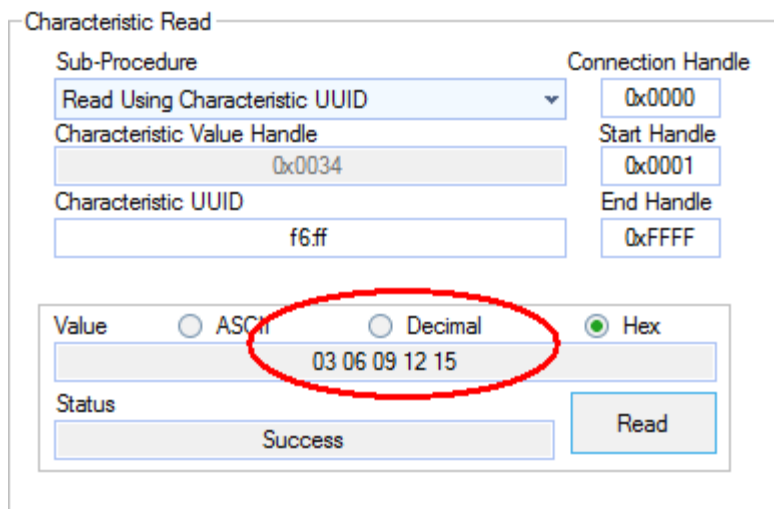
在 handle 为 0x0034 写入其他数据



The 'Characteristic Write' dialog box is shown. The 'Characteristic Value Handle' field contains '0x0034' and is circled in red. The 'Connection Handle' field contains '0x0000'. The 'Value' field contains '03 06 09 12 15' and is also circled in red. The 'Status' field shows 'Success'. The 'Write' button is visible.

图 3.112 写入 CHAR6

再读取一次可看到数据已更改



The 'Characteristic Read' dialog box is shown. The 'Sub-Procedure' is set to 'Read Using Characteristic UUID'. The 'Characteristic Value Handle' field contains '0x0034'. The 'Characteristic UUID' field contains 'f6ff'. The 'Value' field contains '03 06 09 12 15' and is circled in red. The 'Status' field shows 'Success'. The 'Read' button is visible.

图 3.113 写入成功

在这里可能会问，SimpleBLEPeripheral 为什么没有输出，在上面的实验

中改变 CHAR1 是串口会输出改变后的值。那是因为我们还未在 SimpleBLEPeripheral 添加处理。

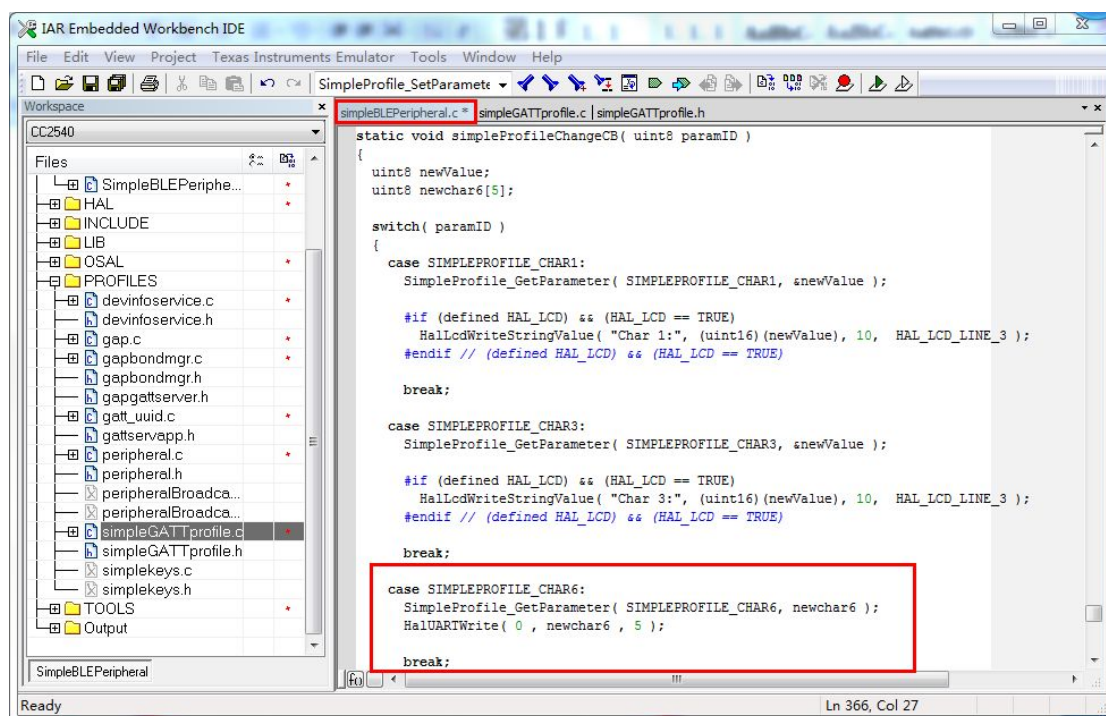


图 3.114 CHAR6 处理函数

再次写入

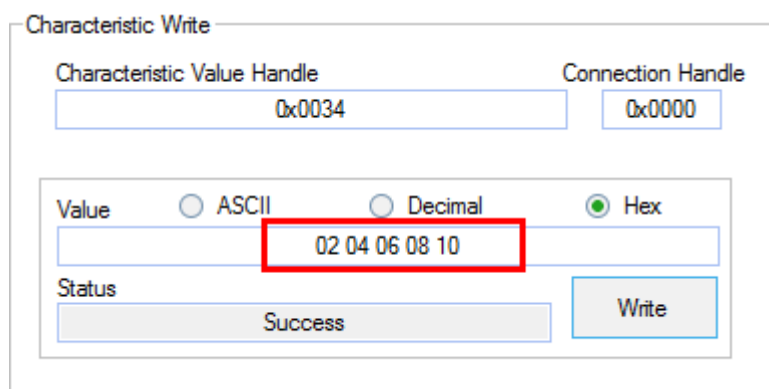


图 3.115 再次写入 CHAR6



图 3.116 打印 CHAR6 值

可看到 SimpleBLEPeripheral 接收到变化。