客户端多连接

两个以上 KW36

1 导出 temp_coll 的 freertos 工程

2 修改 app_preinclude.h，作如下修改

```
#define gAppMaxConnections_c 8
#define gTmrStackTimers_c (6 + gAppMaxConnections_c)
```

不使用低功耗，则

```
#define cPWR_UsePowerDownMode 0
```

3 修改 temperature_collector.c

3.1 修改 `static appPeerInfo_t mPeerInformation`

改为 `static appPeerInfo_t mPeerInformation[gAppMaxConnections_c];`

3.2 定义全局变量 `uint8_t mActiveConnections = 0;`

3.3 修改以下函数

```
1)  static void BleApp_StoreServiceHandles
        (
                        deviceId_t      peerDeviceId,
                        gattService_t   *pService
        );

2)  static void BleApp_StoreDescValues
        (
                        deviceId_t      peerDeviceId,
                        gattAttribute_t     *pDesc
        );

3)  static void BleApp_PrintTemperature
        (


                deviceId_t      peerDeviceId,
                uint16_t temperature

  );
```

3.4 注释`static void DisconnectTimerCallback(void* pParam);`

3.5 找到`BleApp_HandleKeys`函数，作如下修改

```
case gKBD_EventLongPB1_c:
{
        for(uint8_t i = 0; i < gAppMaxConnections_c; i++)
        {
                if (mPeerInformation[i].deviceId != gInvalidDeviceId_c)
                        Gap_Disconnect(mPeerInformation[i].deviceId);
        }
        break;
}
```

3.6 找到`BleApp_Config`函数，修改`mPeerInformation`

```
for(uint8_t i = 0; i < gAppMaxConnections_c; i++)
{
    mPeerInformation[i].appState = mAppIdle_c;
}
```

3.7 找到 BleApp_ScanningCallback

在 case 为 *gScanStateChanged_c* 下

修改为

```
#if (cPWR_UsePowerDownMode)
                    Led1Off();
                    /* Go to sleep */
                    #ifdef MULTICORE_HOST
                        #if gErpcLowPowerApiServiceIncluded_c
                            PWR_ChangeBlackBoxDeepSleepMode(3);
                        #endif
                    #else
                        if(mActiveConnections > 0)
                        {
                         PWR_ChangeDeepSleepMode(1);
                        }
                        else
                        {
                         PWR_ChangeDeepSleepMode(3);
                        }
                    #endif
#else
                    LED_StopFlashingAllLeds();
                    Led1Flashing();
                    Led2Flashing();
                    Led3Flashing();
                    Led4Flashing();

#endif
```

3.8 找到 BleApp_ConnectionCallback

3.8.1 在 case 为 *gConnEvtConnected_c*

修改如下

```
1)  mPeerInformation[peerDeviceId].deviceId = peerDeviceId;
2)  mPeerInformation[peerDeviceId].isBonded = FALSE;
3)  Gap_CheckIfBonded(peerDeviceId, &mPeerInformation[peerDeviceId].isBonded);
4)  if ((mPeerInformation[peerDeviceId].isBonded) &&
            (gBleSuccess_c == Gap_LoadCustomPeerInformation(peerDeviceId,
            (void*) &mPeerInformation[peerDeviceId].customInfo, 0, sizeof (appCustomInfo_t))))
5)  BleApp_StateMachineHandler(mPeerInformation[peerDeviceId].deviceId, mAppEvt_PeerConnected_c);
```

在该 case 结束最后几行

```
mPeerInformation[peerDeviceId].isBonded = FALSE;
    mActiveConnections++;
```

3.8.2 在 case 为 *gConnEvtDisconnected_c*

修改如下

```
1)  mPeerInformation[peerDeviceId].deviceId = gInvalidDeviceId_c;
2)  mPeerInformation[peerDeviceId].appState = mAppIdle_c;
```

在 case 最后几行加

```
mPeerInformation[peerDeviceId].appState = mAppIdle_c;
    mActiveConnections--;
```

找到低功耗代码作如下修改

```
#if (cPWR_UsePowerDownMode)
            /* Go to sleep */
            #ifdef MULTICORE_HOST
                #if gErpcLowPowerApiServiceIncluded_c
                    PWR_ChangeBlackBoxDeepSleepMode(3);
                #endif
            #else
                if(mActiveConnections > 0)
                {
                PWR_ChangeDeepSleepMode(1);
                }
                else
                {
                PWR_ChangeDeepSleepMode(3);
                }
            #endif
            Led1Off();
#else
            LED_TurnOffAllLeds();
            LED_StartFlash(LED_ALL);
#endif
```

3.8.3 在 case 为 gConnEvtPairingComplete_c

修改为

```
BleApp_StateMachineHandler(mPeerInformation[peerDeviceId].deviceId,
mAppEvt_PairingComplete_c);
```

3.8.4 在 case 为 gConnEvtEncryptionChanged_c

修改如下

```
if( gBleSuccess_c != BleApp_ConfigureNotifications(peerDeviceId) )
```

3.9 找到BleApp_ServiceDiscoveryCallback

在case为 gServiceDiscovered_c 作如下修改

```
case gServiceDiscovered_c:
{

     BleApp_StoreServiceHandles(peerDeviceId,pEvent->eventData.pService);
}
break;
```

3.10 找到BleApp_StoreServiceHandles

添加参数

```
static void BleApp_StoreServiceHandles
(
    deviceId_t peerDeviceId,
    gattService_t   *pService
);
```

3.10.1 修改如下代码

```
mPeerInformation[peerDeviceId].customInfo.tempClientConfig.hService=
pService->startHandle;

mPeerInformation[peerDeviceId].customInfo.tempClientConfig.hTemperature    =
```

```
pService->aCharacteristics[i].value.handle;
```

3.10.2 找到 case 为 gBleSig_CharPresFormatDescriptor_d

将 `mPeerInformation` 改为 `mPeerInformation[peerDeviceId]`

3.10.3 case 为 gBleSig_CCCD_d

将 `mPeerInformation` 改为 `mPeerInformation[peerDeviceId]`

3.11 找到 BleApp_StoreDescValues

做如上操作

3.12 找到 BleApp_PrintTemperature

做如上操作

3.13 找到 BleApp_GattClientCallback，将函数修改

```
BleApp_StoreDescValues(serverDeviceId ,mpCharProcBuffer);
```

3.14 找到 BleApp_GattNotificationCallback，修改如下

```
static void BleApp_GattNotificationCallback
(
    deviceId_t  serverDeviceId,
    uint16_t    characteristicValueHandle,
    uint8_t*    aValue,
    uint16_t    valueLength
)
{
    if (characteristicValueHandle == mPeerInformation[serverDeviceId].customInfo.tempClientConfig.hTemperature)
    {
        BleApp_PrintTemperature(serverDeviceId, *(uint16_t*)aValue);
/*
#if (cPWR_UsePowerDownMode)
         Restart Wait For Data timer
        TMR_StartLowPowerTimer(mAppTimerId,
                    gTmrLowPowerSecondTimer_c,
                    TmrSeconds(gWaitForDataTime_c),
                    DisconnectTimerCallback, NULL);
#endif
*/

    }
}
```

3.15 找到 BleApp_StateMachineHandler

修改为 `switch (mPeerInformation[peerDeviceId].appState)`

3.15.1 在 case 为 mAppIdle_c 将 `mPeerInformation` 改为 `mPeerInformation[peerDeviceId]`

3.15.2 在 case 为 mAppExchangeMtu_c，做如上相同操作

3.15.3 在 case 为 mAppServiceDisc_c，做如上相同操作

3.15.4 在 case 为 mAppReadDescriptor_c，做如上修改，同时修改

if(gBleSuccess_c != BleApp_ConfigureNotifications(peerDeviceId))

3.15.5 在 case 为 mAppRunning_c，修改

```
(void)Gap_SaveCustomPeerInformation(mPeerInformation[peerDeviceId].deviceId
```

,

```
(void *)&mPeerInformation[peerDeviceId].customInfo, 0,
sizeof(appCustomInfo_t));
```

然后注释掉 TMR_StartLowPowerTimer 函数

再修改

```
if(gBleSuccess_c != BleApp_ConfigureNotifications(peerDeviceId))
```

3.16 找到 BleApp_ConfigureNotifications

修改如下

```
1)  static bleResult_t BleApp_ConfigureNotifications(deviceId_t peerDeviceId)
2)  mpCharProcBuffer->handle = mPeerInformation[peerDeviceId].customInfo.tempClientConfig.hTempCccd;
3)  GattClient_WriteCharacteristicDescriptor(mPeerInformation[peerDeviceId].deviceId,
                                             mpCharProcBuffer,
                                                    sizeof(value), (void*)&value);
```

3.17 注释掉 DisconnectTimerCallback 所有实现

4 一个板子烧 temp_sense，一个烧该程序，当该程序连接成功后，按下 sw2，可以再次扫描其他广播，并连接。其他实验过程与 server 相似。该程序大部分都在改

`mPeerInformation` 为 `mPeerInformation[peerDeviceId]`如果编译有问题，大概率是有些地方没有做这样修改。

如果想要不断收数据，则和之前 server 做一样操作，要分配定时器，但是要修改定时器回调函数

```
static void TimerNotificationCallback(void *pParam)
{
    uint16_t value = (uint16_t)gCccdNotification_c;

    /* Allocate buffer for the write operation */
    if( mpCharProcBuffer == NULL )
    {
        mpCharProcBuffer = MEM_BufferAlloc(sizeof(gattAttribute_t) + gAttDefaultMtu_c);
    }

    if( mpCharProcBuffer != NULL )
    {
        for(uint8_t mClientId = 0; mClientId < mActiveConnections; mClientId++)
        {
            /* Populate the write request */
            mpCharProcBuffer->handle = mPeerInformation[mClientId].customInfo.tempClientConfig.hTempCccd;
            mpCharProcBuffer->uuid.uuid16 = gBleSig_CCCD_d;
            mpCharProcBuffer->valueLength = 0;
            (void)GattClient_WriteCharacteristicDescriptor(mPeerInformation[mClientId].deviceId,
                                                           mpCharProcBuffer,
                                                           (uint16_t)sizeof(value), (void*)&value);
        }
    }
}
```

结果如下，数字是随机生成的


COM10 - PuTTY
Found device:
FSL_TEMP
006037D1C47A
Scan stopped.
Connecting...
Connected!
Temperature collector -> Press SCANSW to connect to a Temperature Sensor.
Scanning...
Found device:
FSL_TEMP
006037D1C47A
Scan stopped.
Connecting...
Connected!
Scanning...
Found device:
NXP_TEMP
00603733B73E
Scan stopped.
Connecting...
Connected!
Temperature: 594 C