Ble白名单机制

我们可以通过白名单来过滤那些我们不希望获得的广播消息。

我们使用的开发板是frdm-kw41作为扫描者，任意另一块kw板子做为广播者，我使用的是frdm-kw36。例程使用的是temperature\_collector，我们首先实现一个扫描周边设备，显示mac地址的功能，然后再加入指定mac地址到白名单，来观察效果。

直接扫描周边设备，需要关闭连接功能，所以要注释掉所有的App\_Connect这个函数。还有将关于一旦扫描到指定设备后，就关闭扫描的函数Gap\_StopScanning注释掉。在ScanningTimeoutTimerCallback里把Gap\_StopScanning也关掉

最后到CheckScanEvent这个函数，将最后的这段的if语句判断注释掉。

if (foundMatch)

{

/\* UI \*/

shell\_write("\r\nFound device: \r\n");

shell\_writeN((char\*)name, nameLength-1);

SHELL\_NEWLINE();

shell\_writeHex(pData->aAddress, 6);

}

改为

shell\_write("\r\nFound device: \r\n");

shell\_writeHex(pData->aAddress, 6);

然后烧写代码，我们就可以看到结果。搜索出了不同的mac地址



下面添加白名单。首先到app\_config.c文件，将扫面参数最后一个修改掉，使用白名单过滤

gapScanningParameters\_t gScanParams =

{

/\* type \*/ *gScanTypePassive\_c*,

/\* interval \*/ gGapScanIntervalDefault\_d,

/\* window \*/ gGapScanWindowDefault\_d,

/\* ownAddressType \*/ *gBleAddrTypePublic\_c*,

/\* filterPolicy \*/ *gScanWithWhiteList\_c*

};

找到函数App\_StartScanning，将最后一个参数从true改成false，然后我们到BleApp\_Start函数，在if语句里加入这两句

bleDeviceAddress\_t temp = {0xa0,0xb9,0x33,0x37,0x60,0};

Gap\_AddDeviceToWhiteList(*gBleAddrTypePublic\_c*, temp);

Temp变量放mac地址，然后加入白名单，需要注意的点是，mac地址是倒着写的。我的另一块板子mac地址是0x00603744b9a0，但是组成数组时候，要倒着顺序放入数组。

然后就可以看到这样结果。都是相同的mac地址

