佛山大学《物联网通信技术课程》实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 基于BLE协议的串口通信 | | | 成绩 |  |
| 专业班级 | 22物联网工程2班 | | |
| 姓名 |  | 学号 |  | 学期 | 2024-2025（1） |
| 指导教师 | 陆海波 | 实验性质 | 验证性 | 实验日期 | 2024.12.06 |
| 一、实验目的 | | | | | |
| （1）了解BLE协议栈的结构  （2）了解OSAL操作系统的工作原理  （3）体会OSAL系统提供的串口相关的函数  （4）学习协议栈的串口机制  （5）学会调用协议栈提供的函数来接收和发送串口数据 | | | | | |
| 二、实验内容 | | | | | |
| 实验步骤 1 找到源码，打开工程文件  在电脑里面即找到并打开工程的项目，如下图4.1所示：    图4.1 工程界面  2 修改工程参数  选中该工程，打开Project->Options选项卡，如图4.2所示：    图4.2 打开Options选项卡  按图4.3操作，在宏该工程的POWER\_SAVING前面加x，这里的作用是告诉编译器不用编译这个宏定义。    图4.3 更改参数  4.3 编译下载  用CC-debug连接电脑与9号节点的CC2541下载接口，把开关打到左边然后点击编译和下载，如图4.4所示：    图4.4 编译和下载  下载完成后，点击IMG_256，然后点击IMG_256，如图4.5所示：    图4.5 运行代码&关闭调试  4.4 实验效果  第一步：用串口线将CC2541的串口与电脑的串口相连（S1拨打至右侧），结果如下图所示：  IMG_20161130_153943  第二步：打开电脑里面的串口调试助手，按下图配置串口参数，然后打开串口：    第三步：重启单片机主机，串口工具提示如下图中的信息：    第四步：在串口工具输入区输入数据点击发送，然后在蓝牙模块收到就原样输出到串口助手。并且LED2会闪烁，表示串口有数据，如下图所示： | | | | | |
| 三、实验结果 | | | | | |
| 问题分析：  BLE 是一种短距离无线通信技术，目的在以较低的功耗实现不同设备之间的数据传输。它工作在无线 2.4GHz 频段，采用跳频技术来避免干扰。与传统蓝牙技术相比，BLE 更专注于简单、低功耗的应用场景，如传感器数据传输、设备连接等  本次实验关键在如何实现蓝牙模块和串口助手的交互。  源代码分析：  /\*  打印一个字符串  str不可以包含0x00，除非结尾  \*/  void NPI\_PrintString(uint8 \*str)  {  NPI\_WriteTransport(str, osal\_strlen((char\*)str));  }  /\*  打印指定的格式的数值  参数  title,前缀字符串  value,需要显示的数值  format,需要显示的进制，十进制为10,十六进制为16  \*/  void NPI\_PrintValue(char \*title, uint16 value, uint8 format)  {  uint8 tmpLen;  uint8 buf[128];  uint32 err;  tmpLen = (uint8)osal\_strlen( (char\*)title );  osal\_memcpy( buf, title, tmpLen );  buf[tmpLen] = ' ';  err = (uint32)(value);  \_ltoa( err, &buf[tmpLen+1], format );  NPI\_PrintString(buf);  }  添加串口的回调函数。在应用层 simpleBLECentral.c中添加以下代码。  // 串口回调函数， 下面把该回调函数里实现的功能讲解一下  /\*1, 思路: 当串口收到数据后，就会马上调用以下回调函数，在实际测试中发现，此回调  函数调用频繁， 如果你不执行NPI\_ReadTransport函数进行读取， 那么这个回调函数就会  频繁地被执行，但是，你通过串口发送一段数据， 你本意是想处理这一完整一段的数据，所以，我们在下面引入了时间的处理方法， 也即接收的数据够多或者超时，就读取一次数据，然后根据当前的状态决定执行，如果没有连接上，就把所有数据当做AT命令处理， 如果连接上了，就把数据送到对端。 ---------------linzhengcai 2016.02.27\*/  static void NpiSerialCallback( uint8 port, uint8 events )  {  (void)port;//加个 (void)，是未了避免编译告警，明确告诉缓冲区不用理会这个变量  if (events & (HAL\_UART\_RX\_TIMEOUT | HAL\_UART\_RX\_FULL)) //串口有数据  {  uint8 numBytes = 0;  HalLedBlink(HAL\_LED\_2,5,30,100); //添加LED2闪烁指示串口接收到数据  numBytes = NPI\_RxBufLen(); //读出串口缓冲区有多少字节  if(numBytes == 0)  {  return;  }  else  {  //申请缓冲区buffer  uint8 \*buffer = osal\_mem\_alloc(numBytes);  if(buffer)  {  //读取读取串口缓冲区数据，释放串口数据  NPI\_ReadTransport(buffer,numBytes);  //把收到的数据发送到串口-实现回环  NPI\_WriteTransport(buffer, numBytes);  //释放申请的缓冲区  osal\_mem\_free(buffer);  }  }  }  }  运行结果：  用串口线将CC2541的串口与电脑的串口相连，打开串口调试助手，按下图配置串口参数，然后打开串口，重启主机，在输入区输入数据点击发送，然后蓝牙模块收到就原样输出到串口助手。并且LED2 会闪烁，表示串口有数据， | | | | | |
| 四、实验总结 | | | | | |
| 在最近的通信实验中，我获得了宝贵的经验和深刻的见解。BLE技术通过采用跳频策略来规避潜在的干扰，它专注于实现简单且能耗低的应用，这使得它在物联网领域中占有一席之地。在研究源代码的过程中，我特别关注了字符串和数值输出的函数以及串口通信的回调机制，这些让我对数据的处理流程有了更深入的理解。串口通信的回调函数设计得非常精妙，它有效地解决了因频繁接收串口数据而引发的重复调用问题，通过集成时间管理策略，确保数据的完整性和根据设备的连接状态执行正确的操作。  当我亲自通过串口线将设备连接起来，设置好串口参数，并在重启设备后，看到我在输入端发送的数据能够通过蓝牙模块无损地传输到串口助手上，我感到无比自豪。这种将理论知识转化为实际操作成果的过程，让我深刻体会到了学习的乐趣。通过这次实验，我对BLE技术的理解更加深刻，并且对协议栈的串口操作机制和相关函数的应用也变得更加熟练。 | | | | | |
|  | | | | | |
|  | | | | | |