



华中农业大学

HUAZHONG AGRICULTURAL UNIVERSITY

# 物联网工程 实验报告

题目: WSN-Zigbee 通信实验

姓名: 高星杰

学号: 2021307220712

学院: 信息学院

指导老师: 朱容波

中国 武汉

2024 年 5 月

---

## 一 实验目的

1. 通过上位机发指令给协调器, 协调器把串口接收到的指令通过 Zigbee 协议无线发送给光敏传感器的终端节点, 获取到数据后把数据放回给上位机器, 实现无线获取数据的目的。
2. 掌握光敏传感器的使用。
3. 了解和学习光敏传感器工作原理和使用方式; 通过利用端口收发器完成对光敏传感器的调试, 理解光敏传感器的原理。
4. 通过终端机和上位机的配置, 熟悉物联网开发的基本流程, 学习和掌握通过 ZigBee 网络通信。
5. 加深对无线传感器的概念的理解及应用, 通过实践掌握物联网相关知识。
6. 使用 Python 第三方库 Serial 读取串口数据, GUI 实时显示传感器数据数据变化。

## 二 实验设备

- (1) **硬件:** Zigbee 节点两个, CC Debugger 仿真器一套, J-Link 下载线, 物联网实验开发箱一套;
- (2) **软件:** IAR Embedded WorchBench, ZStack 协议栈, Keli4 For ARM 开发软件, 串口调试工具;
- (3) **参考资料:** 协议栈 API 接口说明文档, Zigbee 源码, STM32 工程源码

## 三 实验原理

### 1. 温度传感器

温度传感器是一种用于测量温度的电子装置。它们通过感知温度变化并将其转换为电信号输出, 广泛应用于工业、医疗、家用电器、气象等多个领域。根据工作原理和应用场景的不同, 温度传感器可以分为多种类型, 常见的包括热电偶、热电阻、热敏电阻和红外温度传感器等。

#### 温度传感器的类型及原理

##### ➤ 热电偶 (Thermocouple)

原理: 热电偶由两种不同材料的导体构成, 形成一个闭合回路。当接合点的温度发生变化时, 不同材料间会产生电动势 (Seebeck 效应), 从而在回路中产生电压, 该电压与温度成正比。

##### ➤ 热电阻 (Resistance Temperature Detector, RTD)

原理: 热电阻利用金属材料 (如铂、铜、镍) 电阻值随温度变化的特性。常见的热电阻是铂电阻 (Pt100、Pt1000), 其电阻值随温度呈线性变化。

##### ➤ 热敏电阻 (Thermistor)

原理: 热敏电阻是利用半导体材料的电阻值随温度变化的特性, 通常分为正温度系数热敏电阻 (PTC) 和负温度系数热敏电阻 (NTC)。NTC 热敏电阻在温度升高时电阻值下降, PTC 热敏电阻在温度升高时电阻值上升。

##### ➤ 红外温度传感器 (Infrared Temperature Sensor)

---

原理：红外温度传感器通过检测物体发出的红外辐射来测量温度。根据黑体辐射原理，任何温度高于绝对零度的物体都会发射红外辐射。传感器接收这些辐射并将其转换为电信号，经过处理后得到物体的温度。

#### **温度传感器的工作原理：**

温度传感器的工作原理主要通过材料的物理特性变化来实现，具体步骤如下：

- **温度感应：**温度传感器的敏感元件（如热电偶的接合点、热电阻的电阻元件、热敏电阻的半导体材料等）感应到周围环境的温度变化。
- **信号转换：**感应到的温度变化会引起传感器内部物理参数（如电压、电阻、辐射强度等）的变化。
- **信号处理：**传感器内部电路将这些物理参数的变化转换为电信号，通常是电压或电流信号。
- **输出和显示：**电信号经过进一步处理后，以数字或模拟的形式输出，并通过显示设备显示温度值。

## **2. 光敏传感器**

首先我们的光敏传感器用到的是光敏电阻，光敏电阻器是一种对光敏感的元件，它的电阻值能随着外界光照强弱（明暗）变化而变化，即光敏电阻器是利用半导体光电导效应制成的一种特殊电阻器，对光线十分敏感。它在无光照射时，呈高阻状态；当有光照射时，其电阻值迅速减小。

光敏传感器的类型及原理：

#### ➤ **光电二极管 (Photodiode)**

**原理：**光电二极管是最常见的光敏传感器。当光子击中二极管的半导体材料时，会激发电子，从而产生电流。这种电流与入射光的强度成正比。

#### ➤ **光电晶体管 (Phototransistor)**

**原理：**光电晶体管类似于普通晶体管，但其基极是通过光信号来控制的。入射光子会激发基极，使晶体管导通，并在集电极和发射极之间产生电流。光电晶体管的灵敏度比光电二极管高。

#### ➤ **光电池 (Photovoltaic Cell)**

**原理：**光电池将光能直接转换为电能。它基于光生伏特效应，当光子被吸收时，会在半导体材料中产生电子-空穴对，从而产生电压和电流。

#### ➤ **光敏电阻 (Photoresistor)**

**原理：**光敏电阻的电阻值随入射光强度的变化而变化。光子照射到光敏材料上时，会减少材料的电阻值，从而改变通过的电流。

#### **光敏传感器的工作原理：**

光敏传感器的核心是将光信号转化为电信号。这个过程通常包括以下几个步骤：

- **光子吸收：**传感器中的光敏材料吸收入射光子，产生电子-空穴对。
- **电荷分离：**在电场作用下，电子和空穴被分离并分别向不同方向移动，形成电流或电压。
- **信号处理：**电流或电压信号通过放大、电平转换等处理，转换成可以被测量和分析的电信号。

## **3. Zigbee 协议**

Zigbee 是一种基于 IEEE 802.15.4 标准的无线通信协议，专为低功耗、低数据速率的设备设计，主要用于物联网 (IoT) 应用。Zigbee 协议广泛应用于智能家居

---

居、工业自动化、医疗设备、传感器网络等领域。以下是对 Zigbee 协议的详细介绍。

### Zigbee 协议的架构

Zigbee 协议栈分为以下几层

- 物理层 (PHY)：负责无线信号的传输和接收，包括信号调制、解调、频道选择等。
- 媒体访问控制层 (MAC)：负责数据帧的封装和解封、信道访问控制、数据的发送和接收，以及提供基本的安全功能。
- 网络层 (NWK)：负责网络的建立、维护和路由选择，支持不同的网络拓扑结构，如星型、树型和网状网络。
- 应用支持层 (APS)：提供应用程序之间的数据传输服务，负责应用间的消息路由和安全管理。
- 应用层 (APL)：包含应用对象和 Zigbee 设备对象，定义了设备的具体功能和行为。

### Zigbee 网络的组成

一个 Zigbee 网络由以下三类设备组成：

- 协调器 (Coordinator)：每个 Zigbee 网络有且仅有一个协调器，负责网络的初始化和配置，分配网络地址，管理路由表，是整个网络的核心。
- 路由器 (Router)：路由器负责数据的转发和路由，扩展网络的覆盖范围，可以接收和发送数据包。
- 终端设备 (End Device)：终端设备只负责数据的采集和传输，不参与数据的转发和路由，通常为低功耗设备。

本次中没有使用路由器，因为并不是要搭建很大的网络，而只使用了终端设备和协调器，他们两者之间的通信协议就是使用的 Zigbee。

## 4. Modbus 协议

Modbus 协议是一种用于工业设备间通信的简单而常用的协议。它在工厂自动化和工业控制系统中被广泛应用。

### Modbus 协议的工作方式

- 主从结构：
  - ◇ 主设备 (Master)：主设备就像是一个指挥官，它负责发送命令并询问从设备的状态。
  - ◇ 从设备 (Slave)：从设备就像是士兵或队员，它们接收指挥官的命令，并报告自己的状态。
- 数据交换：主设备会发送一个“请求”（就像问一个问题）。从设备收到请求后，会做出“响应”（就像回答这个问题）。

### Modbus 协议的优点

- 简单易用：Modbus 协议非常简单，不需要复杂的设置就能实现设备间的通信。
- 兼容性强：因为它简单且广泛应用，很多工业设备都支持 Modbus 协议。
- 灵活性高：Modbus 可以通过不同的物理媒介传输数据，比如串口 (RS-232、RS-485) 或以太网 (Modbus TCP)

详细的描述请参考官方资料：配套光盘\附件\ModBus 协议；

下面举一个例子：

发送指令：02 03 00 2C 00 01 45 f0

02：表示设备地址

03：功能码，表示读取数据

00 2C：读取开始的地址

00 01：表示读取多少个寄存器，1 个寄存器有 2 个字节

45 f0：由前面数据通过 CRC-16 计数得来的校验码

返回数据：02 03 02 00 00 FC 44

02：表示从地址为 02 的设备返回

03：表示返回的数据是读取

02：表示 2 个有效字节

00 00：就是 2 个有效字节

FC 44：由前面数据通过 CRC-16 计数得来的校验码

本次实验就是使用 Modbus 协议来使上位机和协调器通信的，并且符合上位机是主，而协调器是次的主从结构。

## 5. 获取光敏传感器数据大概流程

第一步：上位机（电脑）通过串口发送获取光敏传感器数据的指令给协调器。

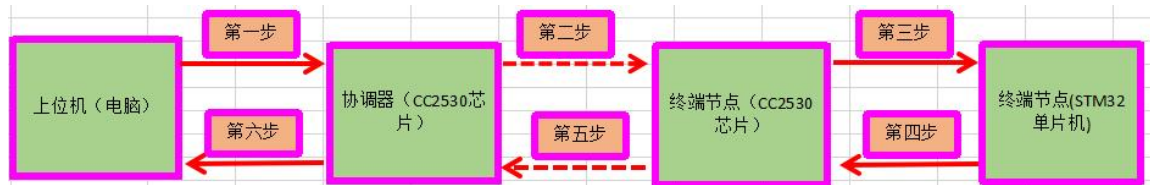
第二步：协调器由串口接收到指令以后，通过 Zigbee 协议无线把指令发送给终端节点。

第三步：终端节点接收到指令以后，通过串口把指令发给终端节点上面的 STM32 芯片。

第四步：STM32 芯片串口接收到指令以后，作出响应，然后把带有光敏传感器数值的数据通过串口返回给终端节点上面的 CC2530 芯片。

第五步：终端节点 CC2530 芯片串口接收到光敏传感器数据以后，通过 Zigbee 协议无线把数据发送给协调器（CC2530 芯片）。

第六步：协调器（CC2530 芯片）无线接收到光敏传感器数据以后，通过串口把数据返回给上位机。



## 四 实验步骤


### 1. 有线传输

这里的有线传输是指，光敏传感器获得数据直接通过 Modbus 协议传输给上位机（电脑）。

---

(1) 将光敏传感器安装在节点上, 打开 Keil 集成开发环境, 打开配套光盘\源代码\Zigbee 无线传感网原理与应用\实验 8 基于 ZStack 光敏传感器实验-V20170222\STM32 固件库 V3.5 移植--实验 8 光敏传感器实验\RVMDK 的 Test.uvproj 工程文件。


**这里使用的软件版本不对, 会导致没有办法编译**

(2) 点击左上角的 Rebuild 按钮, 编译整个工程, 将生成 Test.hex 可执行文件, 该可执行文件自动保存在路径下源代码\传感器原理及应用\实验 8 光敏传感器实验-V20170222\STM32 固件库 V3.5 移植--实验 8 光敏传感器实验\RVMDK\Output 目录下。(注意: 请根据该目录下 Test.hex 文件的生成时间, 判断该文件是否是自己刚刚编译完成的。)

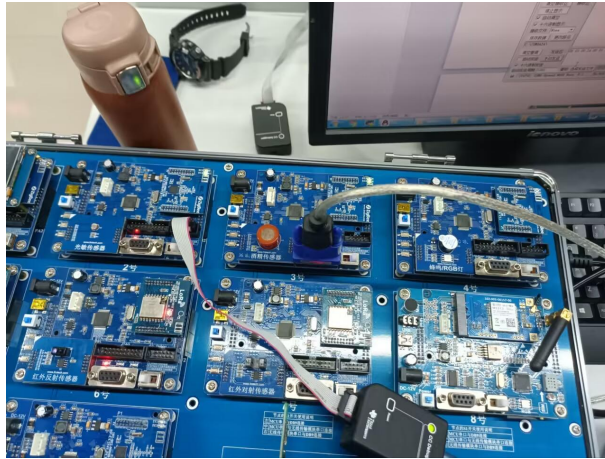
实验中找到源代码编译, 发现报错后, 后来查阅资料后发现是因为 C51 中没有 ARM 环境, 于是将 ARM 所在的内容复制到 C51 中, 重新编译程序, 编译成功了。

```
compiling stm32f10x_i2c.c...
compiling stm32f10x_iwdg.c...
compiling stm32f10x_pwr.c...
compiling stm32f10x_rcc.c...
compiling stm32f10x_rtc.c...
compiling stm32f10x_sdio.c...
compiling stm32f10x_spi.c...
compiling stm32f10x_tim.c...
compiling stm32f10x_usart.c...
compiling stm32f10x_wwdg.c...
linking...
Program Size: Code=7644 RO-data=268 RW-data=64 ZI-data=2016
FromELF: creating hex file...
".\Output\Test.axf" - 0 Error(s), 0 Warning(s).
```

实验中找到了该文件路径所在的位置, 并成功编译和运行了文件。

(3) 把 J-LINK 与节点、电脑连接, 点击左上角的将程序下载到节点板, 使用配套的串口线将计算机串口与节点的 DB9 串口接头相连, 如图 4.1 所示。

➤ **注意事项:** 将节点右下角上的白色三位拨打开关拨至左边, 给节点重新上电, 打开配套光盘\第三方应用软件\串口调试及 CRC 软件\串口调试工具.exe 程序, 打开正确的端口, 进行 9600-8-N-1 设置, 打开串口, 就可以与芯片进行数据交互 (使用的是 Modbus 协议进行数据交互)



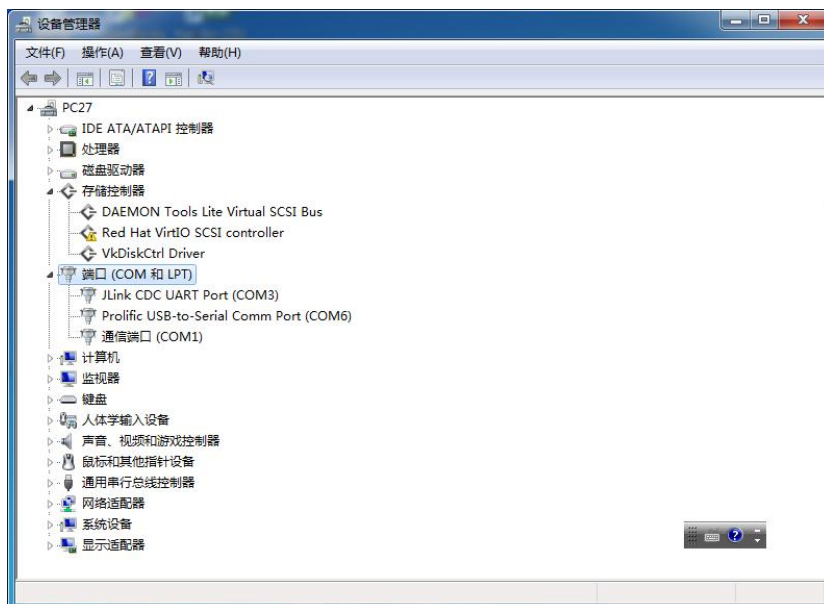
(图 4.1) 注意: 要连接好下载器再进行下载, 并且确保三位开关已经拨到最左边。(这里是实验成功的关键。)

#### (4) 进行下载

```
-----  
Device: STM32F103RB  
VTarget = 3.337V  
State of Pins:  
TCK: 1, IDI: 0, TDO: 0, TMS: 0, TRES: 1, TRST: 1  
Hardware-Breakpoints: 6  
Software-Breakpoints: 8192  
Watchpoints: 4  
JTAG speed: 2000 kHz  
  
Erase Done.  
Programming Done.  
Verify OK.  
* JLink Info: Found Cortex-M3 r1p1, Little endian.  
* JLink Info: TPIU fitted.  
* JLink Info: FPUnit: 6 code (BP) slots and 2 literal slots  
Application running ...
```

#### (5) 配置串口调试小工具

查看端口



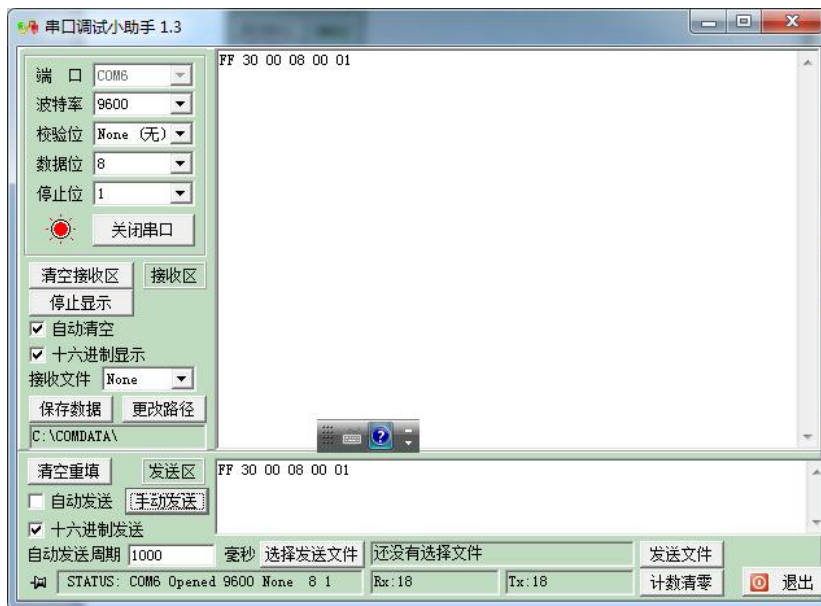
可以发现是端口 6, 然后在串口调试小工具中选择对应的配置



(4) 配置节点地址,如下图。

发送: FF 30 00 08 00 01

返回: FF 30 00 08 00 01



下面我们来解析一下发送的数据和返回的数据为什么是这样的:

指令格式:

我们将数据帧按照 Modbus RTU 协议的常见格式进行拆分:

每个字节的意义如下:

- 设备地址 (1 字节) : 标识目标从设备的地址。
- 功能码 (1 字节) : 指定要执行的操作。
- 数据字段 (4 字节) : 具体操作的数据内容。
- CRC 校验 (2 字节, 未显示) : 用于数据校验 (假设未包含在示例数据中)。

发送数据解析:

**FF 30 00 08 00 01**



- 设备地址 (FF) :  
FF: 广播地址, 所有设备都会响应。
- 功能码 (30) :  
30: 假设为某种特定功能码, 不是标准的 Modbus 功能码 (标准功能码范围在 0x01 到 0x10) 。这个功能码可能是自定义的或特定设备协议中的功能码。
- 数据字段 (00 08 00 01) :
  - ✧ 00 08: 假设为要配置的新地址, 0x0008 表示新地址 8。
  - ✧ 00 01: 可能是命令参数或其他数据, 具体含义取决于设备协议。

接收数据解析:

**FF 30 00 08 00 01**

- 设备地址 (FF) :  
FF: 广播地址, 表示所有设备都接收到了这个请求
- 功能码 (30) :  
30: 与发送请求中的功能码一致, 表示确认执行相同的功能。
- 数据字段 (00 08 00 01) :
  - ✧ 00 08: 确认已将设备地址设置为 8。
  - ✧ 00 01: 确认命令参数或其他数据一致。

#### (5) 获取传感器数据

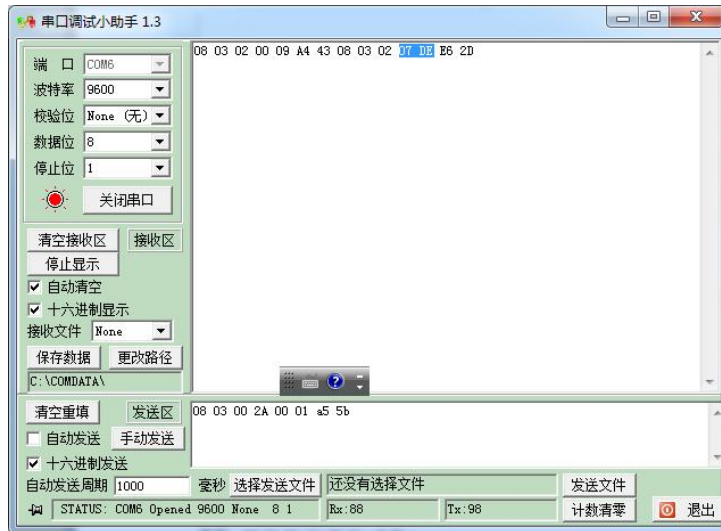
发送: 08 03 00 2A 00 01 a5 5b

返回: 08 03 02 **00 72** E4 60 , 其中 00 72 就是采集光敏传感器的 AD 值, 不同光照条件下这个值会相应不同。如图 4.3.



(图 4.3)

如图 4.4, 在遮光后和放开后分别获取光线数据: 是 00 09 -> 00 DE 很明显光线变强的了, 所以返回的数据也变大了。

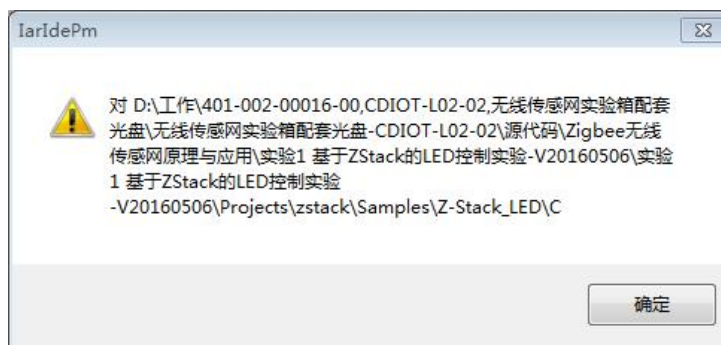


(图 4.4)

## 2. 无线传输

### (1) 解压源码

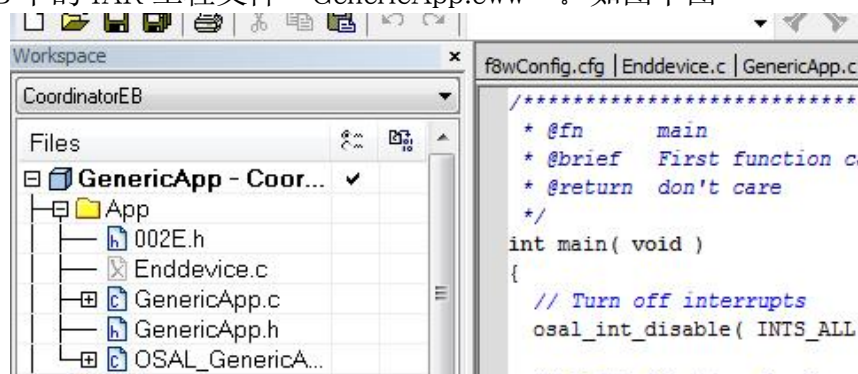
解压缩配套光盘\源代码\Zigbee 无线传感网原理与应用下的实验 8 基于 ZStack 光敏传感器实验-V20170222 文件到电脑磁盘第一级目录, 建议放在桌面, 因为我们的配套光盘目录比较深, 如果直接在配套光盘打开 Zigbee 工程文件的话, 会出现图 4.4 所示报错等原因而打不开工程。



后来找到了文件路径解决了这个问题。

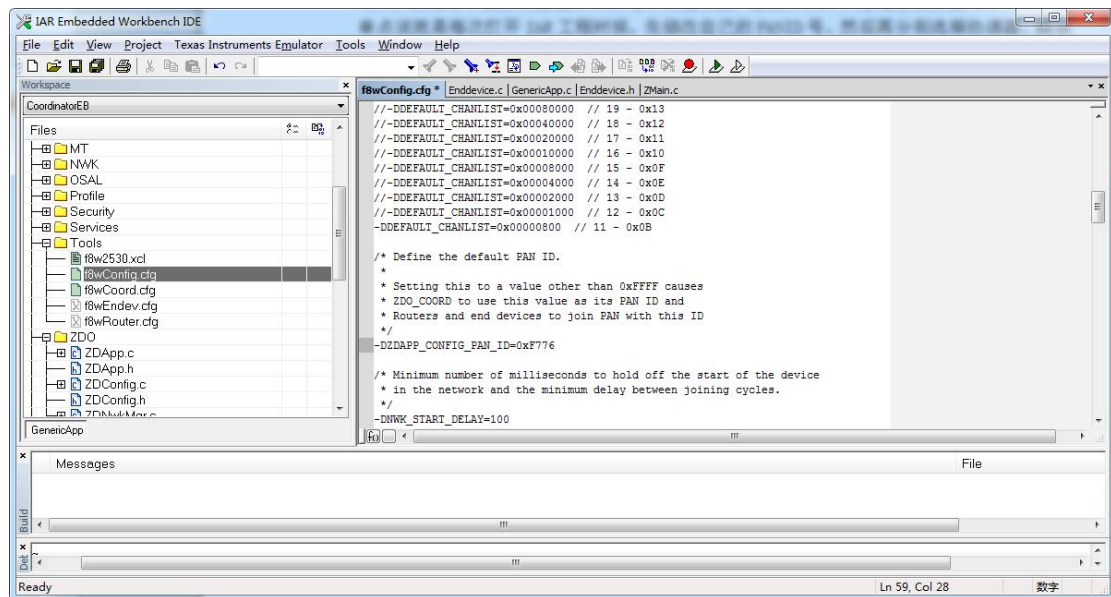
### (2) 打开源码

在电脑桌面打开刚解压以后的文件实验 8 基于 ZStack 光敏传感器实验-V20170222\ZStack-CC2530-2.5.0\Projects\zstack\Samples\GenericAppV2.3\CC2530DB 下的 IAR 工程文件“GenericApp.eww”。如图下图

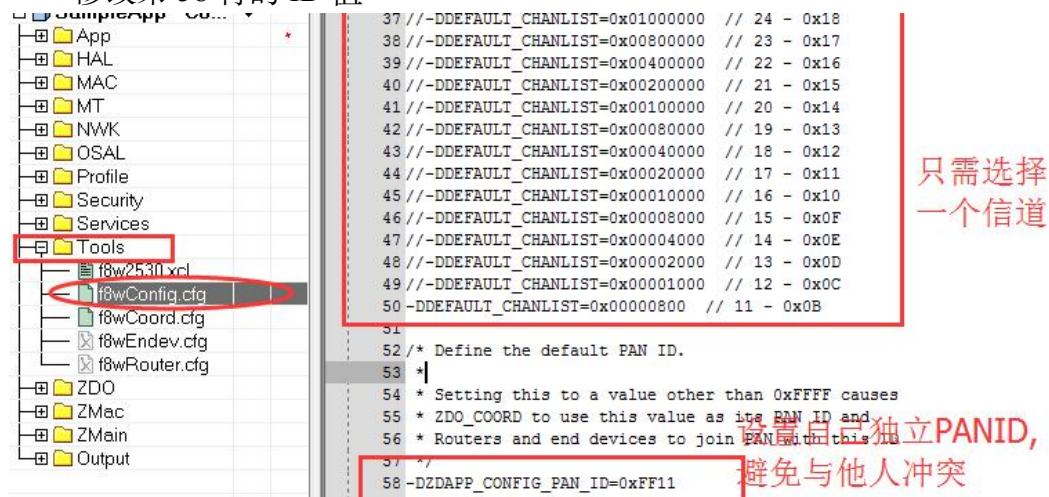


### (3) 修改 Zigbee 的 PANID

先找到 Tools/f8wConfig.cfg 配置文件中的 PANID, 然后修改为自己的设定的一个值。



修改第 58 行的 ID 值

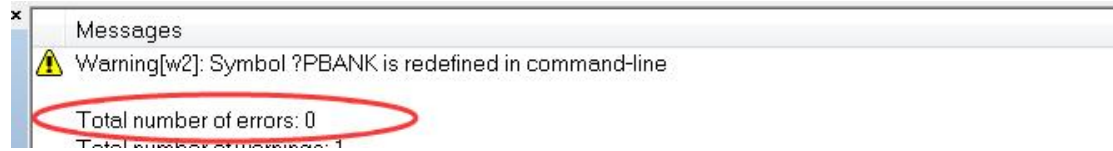


### (4) 选择不同项目进行编译下载

首先是选择协调器的项目



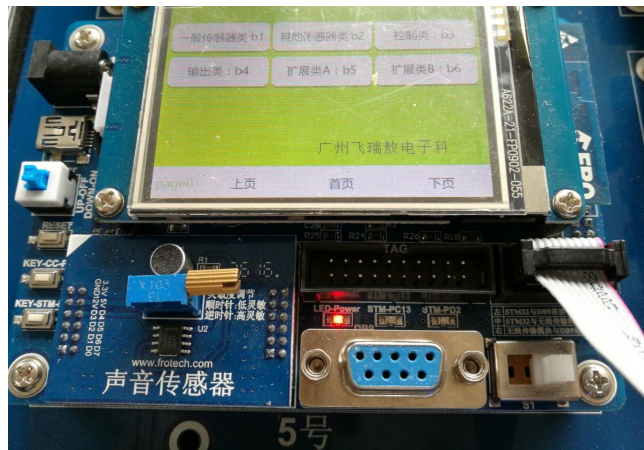
点击上边工具栏 project->Rebuild All, 编译整个工程, 将生成 GenericApp.hex 可执行文件, 该可执行文件自动保存在 实验 8 基于 ZStack 光敏传感器实验 -V20170222\ZStack-CC2530-2.5.0\Projects\zstack\Samples\GenericAppV2.3\CC2530\CoordinatorEB\Exe 目录下。(注意: 请根据该目录下 GenericApp.hex 文件的生成时间, 判断该文件是否是自己刚刚编译完成的。)编译成功, 则下边信息栏会提示 0 错误。如图 4.8 所示



(图 4.8)

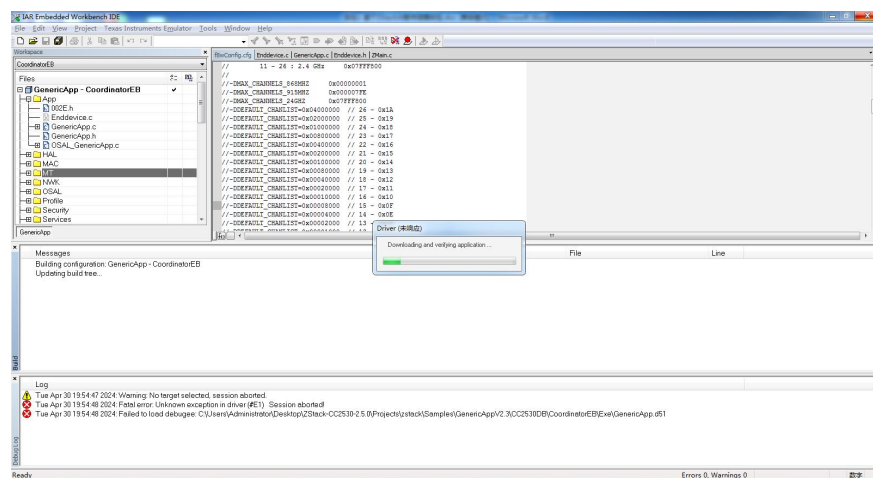
(5) 下载到 CC2530 (协调器)



用 CCDebugger 仿真器将电脑与协调器节点下载接口相连。(注意观察 CCDebugger 仿真器的灯是否变绿, 如果是红色则按仿真器的复位键; 让仿真器灯变绿) 如图 4.9 所示。

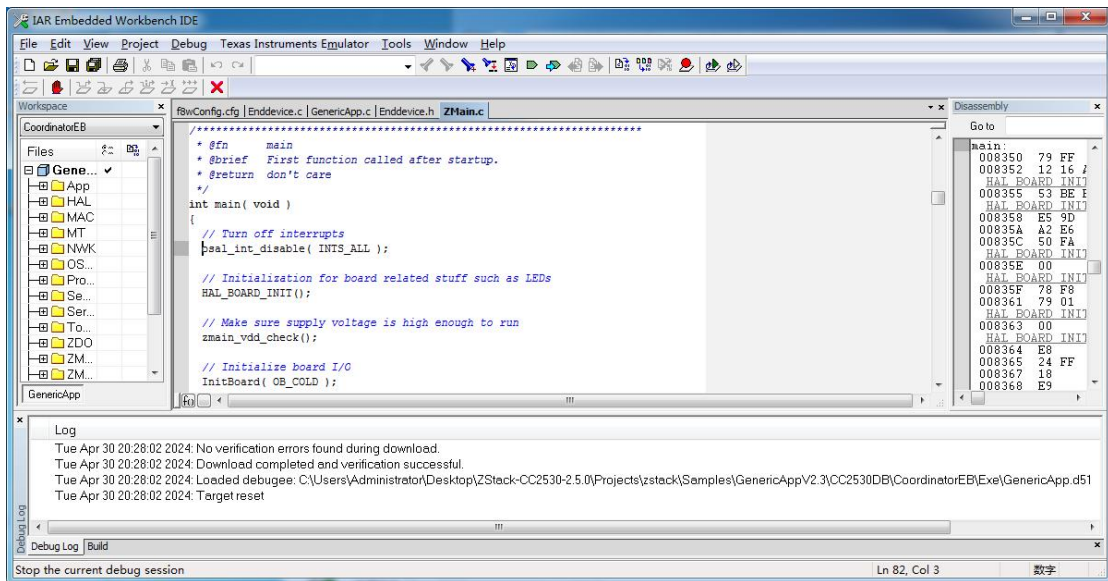


(图 4.9)

点击工具栏的下载按钮 , 将程序下载到 CC2530 上



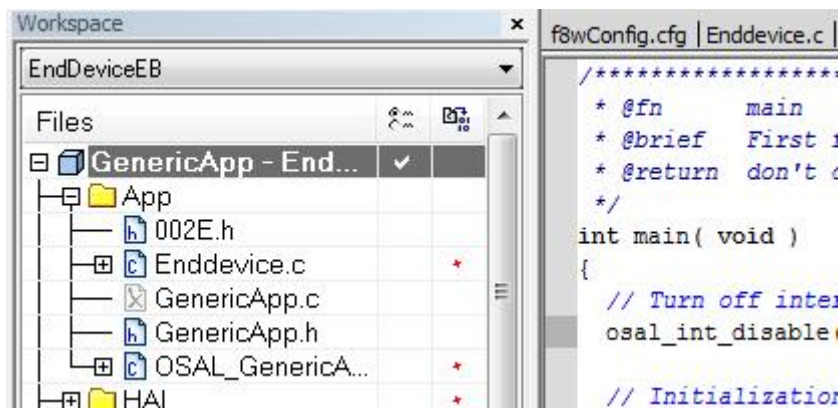
然后点击 , 然后点击  如图 4.10 所示。



(图 4.10)

(6) 下载到终端中

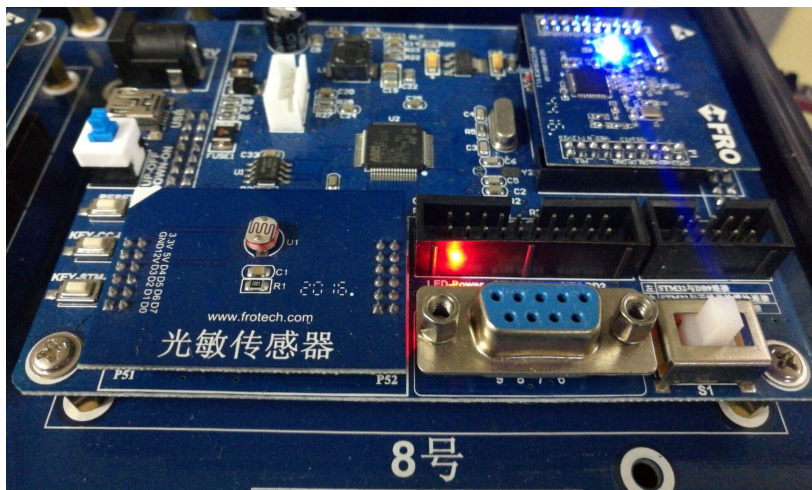
在 IAR 里面选择“EndDeviceEB”工程，如图 4.11。重复第之前的步骤下载“EndDeviceEB”工程代码到终端节点。



(图 4.11)

(7) 使终端和协调器工作

把节点的三档开关拨打到中间



重新打开或按 Reset 复位节点底板。用公母直连串口线将协调器节点的串口与电脑的 9 针 COM 口相连，协调器的开关 S1 拨打到右边，让 DB9 和 CC2530 相连。如图 4.13 所示。

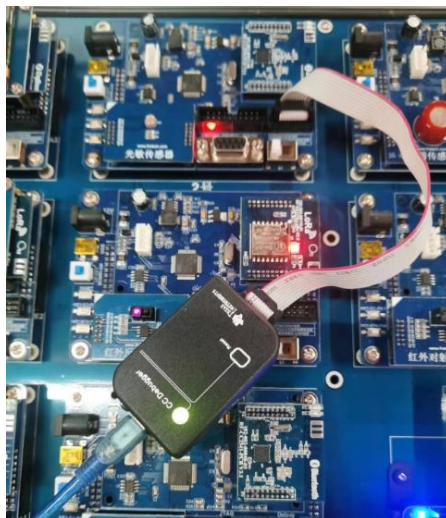


图 4.13

## 五 实验结果

### 1. 光敏传感器无线传输数据

(1) 将程序下载到光敏传感器后，利用串口调试工具测试。打开串口调试助手，串口参数：9600-8-N-1，如图 4.14

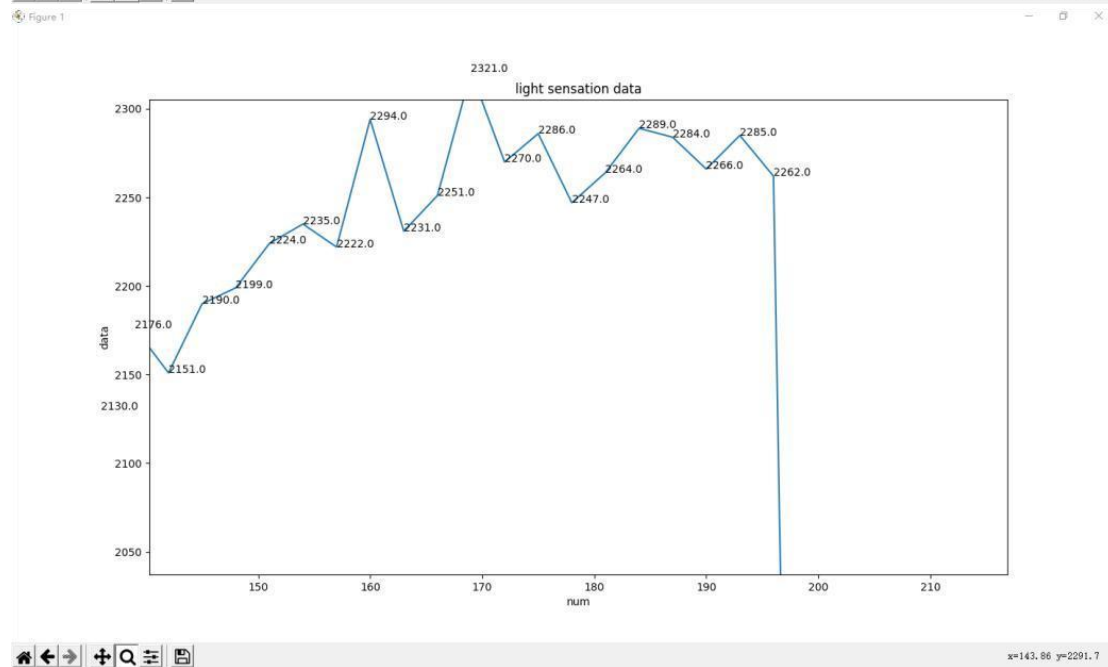
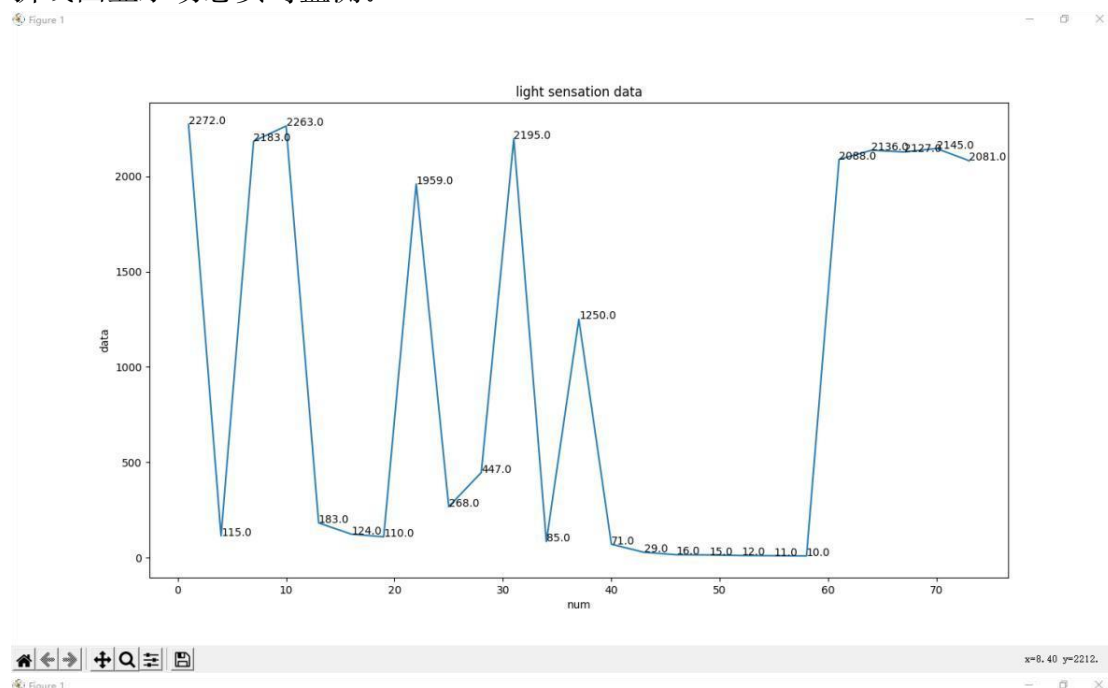
发送：08 03 00 2A 00 01 a5 5b

返回：08 03 02 07 9F 26 1D; 08 03 02 00 03 24 44; 08 03 02 07 B8 66 07; ;  
其中 07 9F -> 00 03 -> 07 B8 就是采集光敏传感器的 AD 值，不同光照条件下这个值会相应不同，这就是在遮挡前后的光线变化。如图 4.14。



(图 4.14)

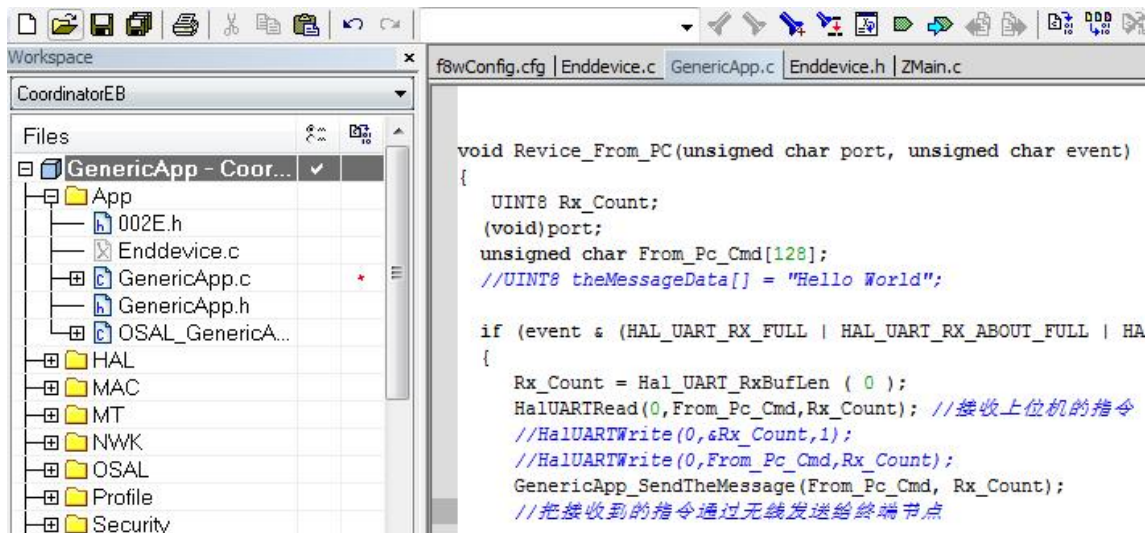
(2) 利用 python 实现端口数据读取，得到光敏传感器的返回数据，并在界面中以折线图显示动态实时监测。



## 2. 项目源码分析

(1) 在 GenericAPP.c 里面

协调器接收到上位机发送的指令: HalUARTRead(0,From\_Pc\_Cmd,Rx\_Count);如图 4.15



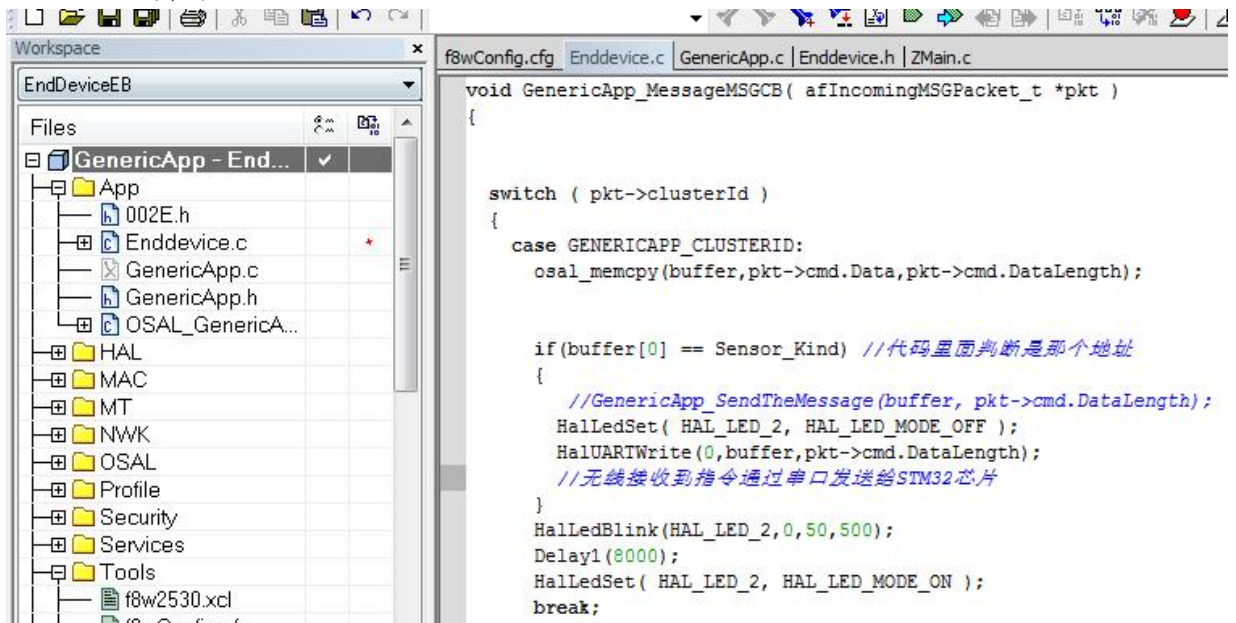
(图 4.15)

(2) 在 GenericAPP.c 里面

协调器把接收到的指令通过 Zigbee 无线发送给终端节点:

GenericApp\_SendTheMessage(From\_Pc\_Cmd, Rx\_Count);, 如图 4.15

(3) 在“Enddevice.c”里终端节点接收到指令以后, 把指令通过串口发送给 STM32 芯片, 如图 4.16

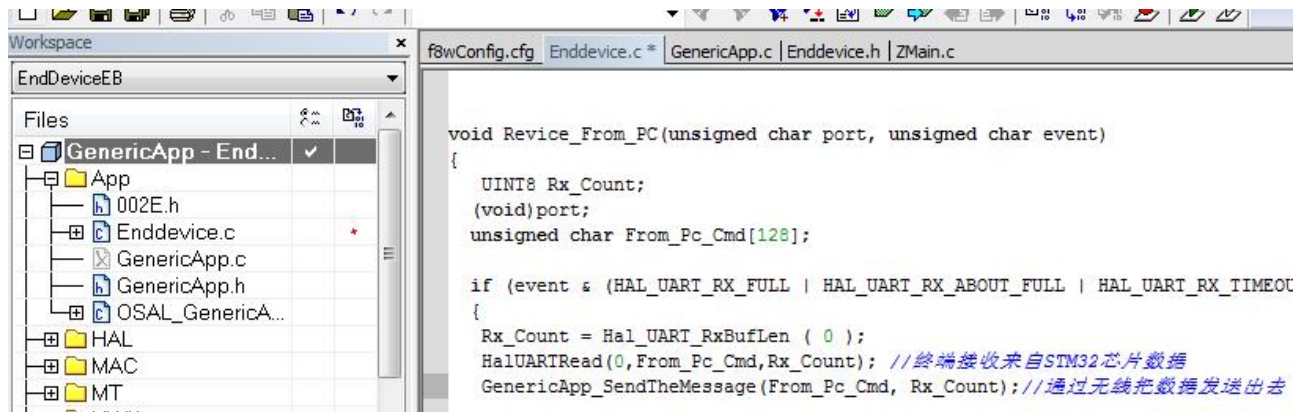


(图 4.16)

(4) STM32 芯片串口接收到数据以后, 处理指令, 然后通过串口把带有光敏传感器数值的数据返回给终端节点。

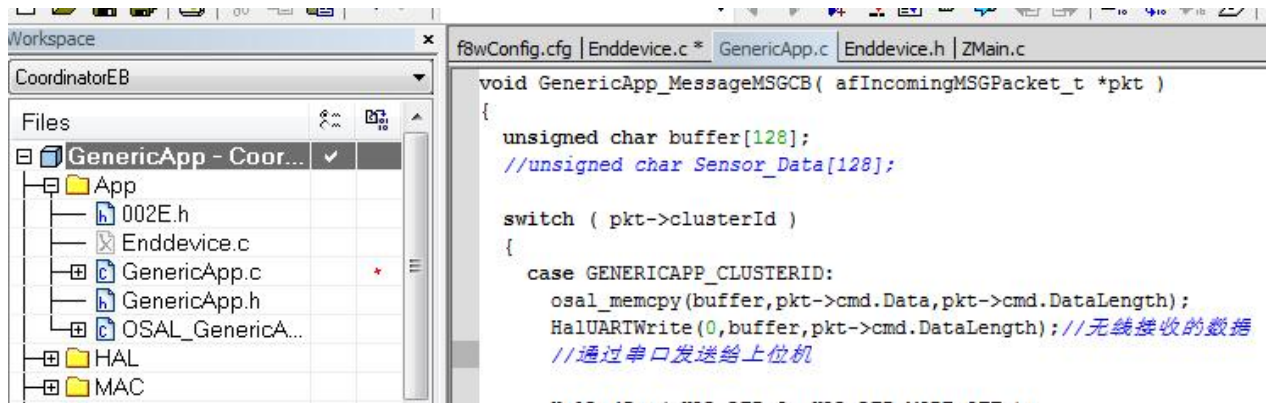
(5) 在“Enddevice.c”里, 终端节点通过串口接收到数据以后通过无线把数据发送给协调器。如图 4.17.





(图 4.17)

(6) 在 GenericAPP.c 里面，协调器接收到无线数据，并把数据通过串口发送给上位机。如图 4.18



(图 4.18)

### 3. python 可视化界面代码

#### (1) 代码 (逐行注释)

```

1. import binascii, time # 导入 binascii 库和 time 库
2. prot = "COM5" # 串口号设置为 COM5
3. bit = 8 # 数据位设置为 8 位
4.
5. # 初始化串口设置: 端口号 COM5, 波特率 9600, 数据位 8, 超
   时设为 50, 停止位为 1
6. ser = serial.Serial(prot, 9600, bits=8, timeout
   =50, stopbits=1)

```

```
7.
8. a = "98 03 0a 2a 00 01 a5 5b" # 定义命令字符串 a
9. d = bytes.fromhex(a) # 将字符串 a 从十六进制字符串转换为字节
10.
11. data = [] # 创建空列表 data, 用于存储读取的数据
12. num = [] # 创建空列表 num, 用于存储数据点编号
13.
14. i = 1 # 初始化计数器 i 为 1
15.
16. while True: # 开始无限循环
17.     result = ser.write(d) # 向串口写入数据 d
18.     count = ser.inWaiting() # 返回接收缓冲区的字节数
19.     data = ser.read(count) # 读取 count 个字节的数据
20.
21.     if data != b'': # 如果 data 非空
22.         receive_data = str(binascii.b2a_hex(data)) # 将读取的数据转换为十六进制字符串
23.         print("data is " + receive_data) # 打印接收到的数据
24.
```

```
25.         # 从接收到的数据中截取前 9 个字符，并转换为十六
进制整数
26.         value = int(receive_data[0:9], 16)
27.         data.append(value) # 将转换后的值添加到
data 列表中
28.         num.append(i) # 将当前计数器值添加到 num 列
表中
29.         i += 3 # 计数器增加 3
30.
31.         # 用 matplotlib 库绘图
32.         plt.clf() # 清除当前图形
33.         plt.title('Light sensation data') # 设置图
表标题
34.         plt.xlabel('num') # 设置 x 轴标签
35.         plt.ylabel('data') # 设置 y 轴标签
36.         plt.plot(num, data) # 绘制 num 和 data 列表的数
据点
37.
38.         # 遍历 num 和 data 列表，标记每个点
39.         for a, b in zip(num, data):
40.             plt.text(a, b, f'{a}, {b}', fontsize=1
0) # 在图中标记点的坐标
41.
```

```
42. plt.pause(0.1) # 暂停0.1秒
43. plt.ioff() # 关闭交互模式
44. time.sleep(1) # 程序休眠1秒
45.
46. ser.close() # 关闭串口连接
```

## (2) 代码分析:

代码主要分为两个部分，第一部分读取端口数据，第二部分对 AD 值部分进行选取并实时汇出折线图。

第一部分，我们调用了 serial 库，进而实现与串口通信，设置好自己电脑的端口，校验码，波特率，数据位等值，然后写入'0803002a0001a55b'命令，读取端口数据。

第二部分，成功读取后，选取 receive\_data[6:9]，这四位是采集光敏传感器的 AD 值，不同光照条件下值不同。将其转换为十进制后，进行 matplotlib 绘制折线图即可。

## 六 实验心得

首先，在对所学的理论课而言，实验给了我们一个很好的把理论应用到实践的平台，让我们能够很好的把书本知识转化到实际能力，提高了对于理论知识的理解，认识和掌握。其次，对于个人能力而言，实验很好的解决了我们实践能力不足且得不到很好锻炼机会的矛盾，通过实验，提高了自身的实践能力和思考能力，并且能够通过实验很好解决自己对于理论的学习中存在的一些知识盲点。对于团队协作与待人处事方面，实验让我们懂得了团队协作的重要性，教导我们以谦虚严谨的态度对待生活中的人与事，以认真负责的态度对待队友，通过实验的积极的讨论，理性的争辩，可以让我们更加接近真理。

实验中应注意的有几点。

一是一定要先弄清楚原理。这样在做实验，才能做到心中有数，从而把实验做好做细。一开始，实验比较简单，可能会不注重此方面，但当实验到后期，需要思考和理解的东西增多，个人能力拓展的方面占一定比重时，如果还是没有很好的做好预习和远离学习工作，那么实验大部分会做的很不尽人意。

二是在养成习惯方面。一定要真正的做好实验前的准备工作，把预习报告真正的学习研究过，并进行初步的实验数据的估计和实验步骤的演练，这样才能在真正实验中手到擒来，做到了然于心。

不过说实话，在做试验之前，我以为不会难做，就像以前做的实验一样，操作应该不会很难，做完实验之后两下子就将实验报告写完，直到做完实验后，我

---

才知道其实并不容易做,要注意其中的很多细节,而且要真正的领悟其中的道理,而不是无脑跟着视频操作,这样我们不会有真正的收获的。它真的不像我想象中的那么简单,天真的以为自己把平时的理论课学好就可以很顺利的完成实验,事实证明我错了。

幸运的是,在这次实验中我获益良多。本着实践性、创新性、可行性和有意义性的原则,我完成了实验设计。虽然实验原理本身并不十分复杂,但对于我这样的初学者而言,掌握实验原理本身就是一大挑战。我通过翻阅相关书籍、查阅资料,加深了对实验功能的理解,将实验与理论有机结合,并在同学和老师的帮助下最终完成了实验。这次实验给我的最深刻印象是严谨求实的重要性:努力弄懂原理的同时也要勇于创新,能做好的事就应该全力以赴,把生活工作学习视为雕刻艺术品一般,全身心投入其中,只有这样,命运自然会为你的努力买单。

最后,感谢老师的辛苦指导和同学的热心帮助。