**实验9： Zstack的雨滴和执行器控制闭环实验**

**一、 实验目的**

1.了解基于 Z-Stack 协议栈的 SappWsn 应用程序框架的工作机制

2.掌握在 SappWsn 应用程序框架下将执行节点与雨滴传感器组网的方法

**二、 实验环境**

1. 装有 IAR 开发工具的PC 机一台
2. 实验箱一台
3. CCDebugger（以及USB A-B 延长线）一个
4. USB Mini 延长线一根

**三、 实验原理**

在 雨滴传感器发送状态数据给协调器的同时，发送数据给执行器，以改变继电器的状态

**四、 实验要求**

利用 Z-Stack APP 应用程序框架，同时添加执行节点和雨滴传感器的驱动程序，使得节点可以周期性发送雨滴传感器的状态数据给协调器，同时根据余地传感器的状态数据发送控制指令给执行节点，控制继电器的状态发生变化。

**五、 实验步骤与现象**

**5.1实验步骤**

1.本实验需要用到三个节点，一个作为协调器，另一个作为执行节点；

首先使用 Mini USB 延长线将 ZigBee 网关的 Mini USB 接口连接至PC 机的 USB 接口，如图 2.80所示；

将调试器一端使用 USB A-B 延长线连接至 PC 的 USB 接口，另一端的 10pin 排线连接到实验箱

JTAG 调试接口，如图 2.81 所示；

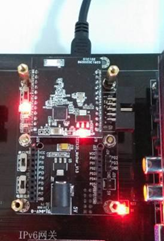
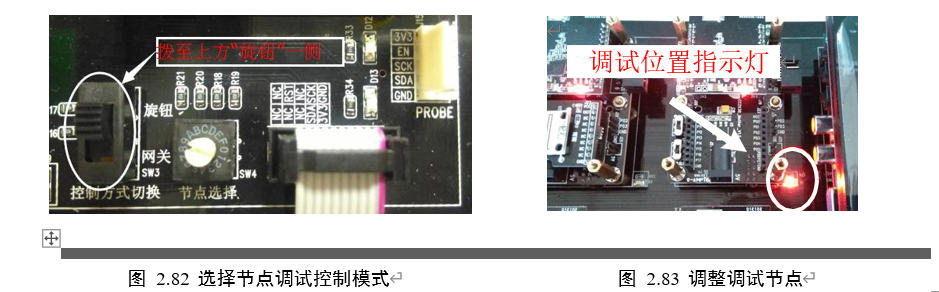


图 2.80 将 ZigBee 网关的串口连接至PC 机 图 2.81 程序下载硬件连接图

将实验箱右“控制方式切换”开关拨至“旋钮”一侧，如图 2.82 所示；

转动实验箱“旋钮节点选择”旋钮，使得 ZigBee 网关旁边的 LED 灯被点亮，如图 2.83 所示；



打开配套代码中的ZStack-CC2530\Projects\SappWsn\SappWsn.eww 工程文件，如图 2.84 所示

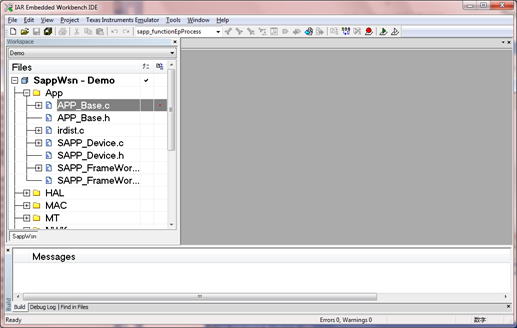
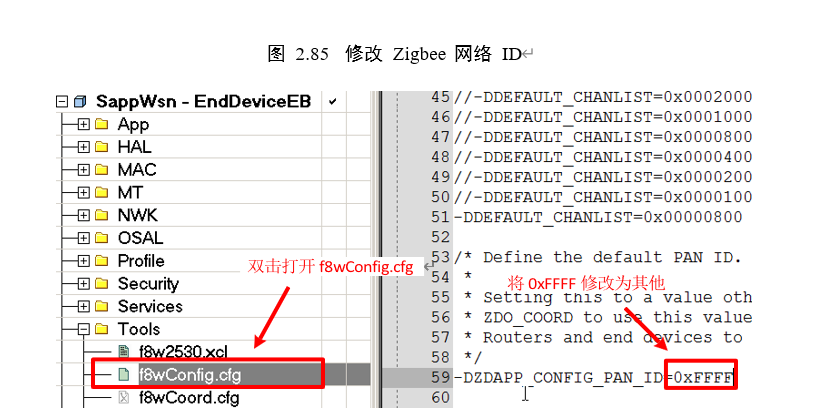


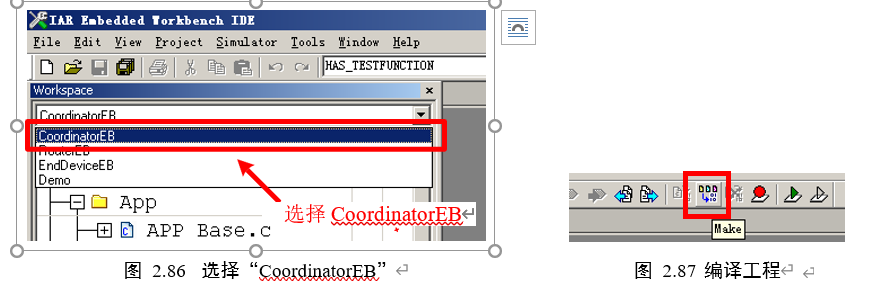
图 2.84  打开 SappWsn 工程

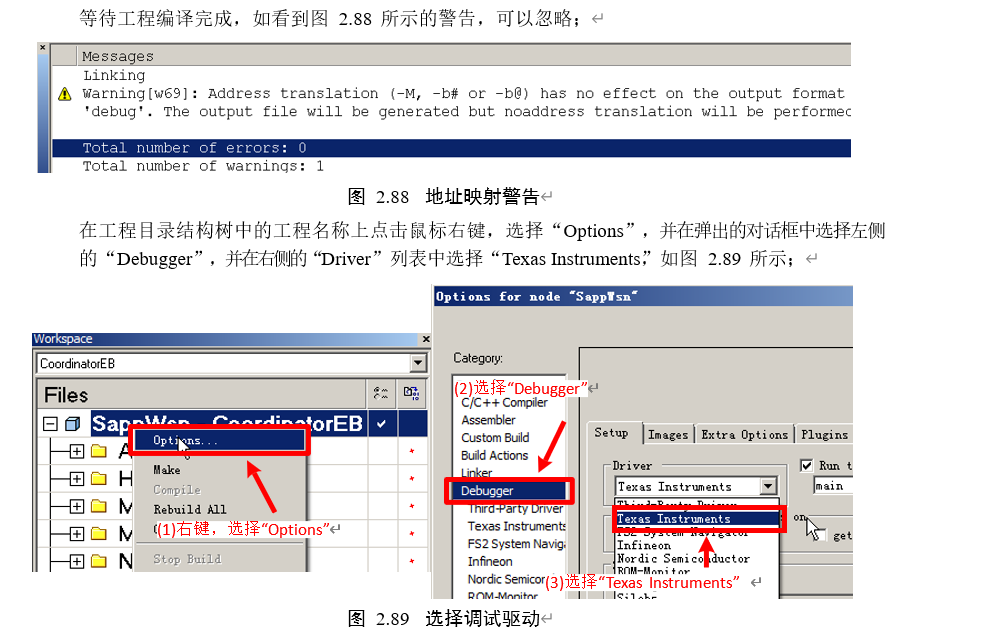
在“ Tools ” 组中， 找到“ f8wConfig.cfg ” 文件， 双击打开， 并找到大概第 59 行的“-DZAPP\_CONFIG\_PAN\_ID=0xFFFF”，将其中的“0xFFFF”修改为其他值，例如“0x0010"



在工程目录结构树上方的下拉列表中，选择“CoordinatorEB”，如图 2.86 所示；

点击工具栏中的“Make”按钮，编译工程，如图 2.87 所示；





点击“Download and Debug”按钮，如图 2.90 所示；

待程序下载完毕后，点击“Go”按钮，使程序开始运行，如图 2.91 所示；



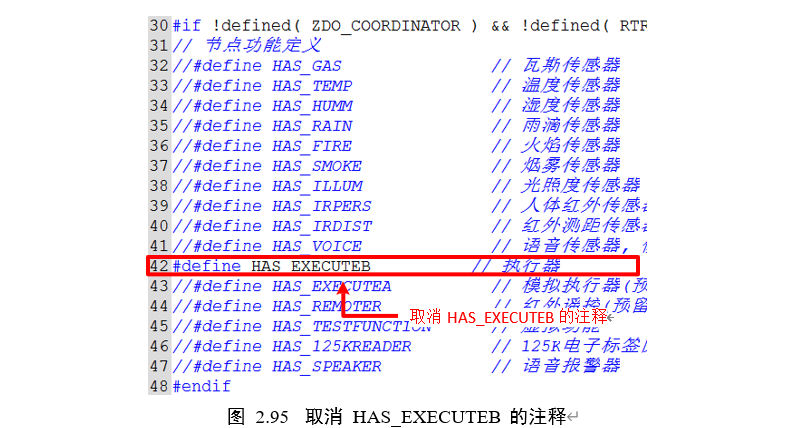
图 2.90 下载并进入调试状态 图 2.91 运行程序

点击工具栏中的“Stop Debugging”，退出调试模式，如图 2.92 所示；

转动实验箱“旋钮节点选择”旋钮，使得执行节点旁边的 LED 灯被点亮，如图 2.93 所示；



在“SAPP\_Device.h”文件中，取消“HAS\_EXECUTEB”的注释，并保证其他的功能均被注释， 如图 2.95 所示；



点击工具栏中的“Make”按钮，编译工程，如图 2.96 所示；

等待工程编译完成，如看到图 2.97 所示的警告，可以忽略；

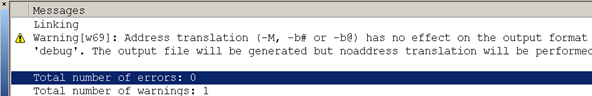


图 2.97  地址映射警告

在工程目录结构树中的工程名称上点击鼠标右键，选择“Options”，并在弹出的对话框中选择左侧的“Debugger”，并在右侧的“Driver”列表中选择“Texas Instruments”，如图 2.98 所示；

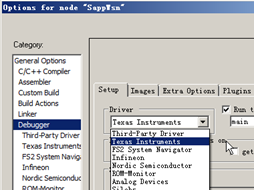
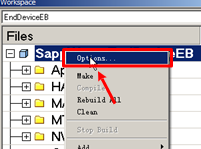


图 2.98  选择调试驱动

点击“Download and Debug”按钮，如图 2.99 所示；

待程序下载完毕后，点击“Go”按钮，使程序开始运行，如图 2.100 所示；



图 2.99 下载并进入调试状态 图 2.100 运行程序

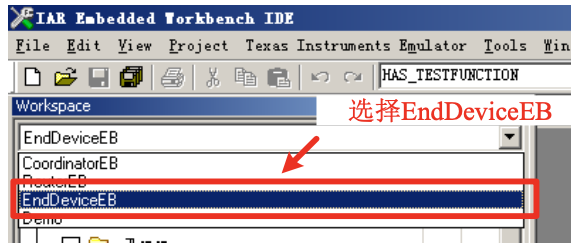
稍等片刻，可以看到执行节点的两个 LED 灯同时闪烁，表示加入到协调器组建的 Zigbee 网络；

点击工具栏中的“Stop Debugging”，退出调试模式，如图 2.101 所示;

转动实验箱“旋钮节点选择”旋钮，使得雨滴传感器节点旁边的 LED 灯被点亮，如图 2.102 所示;

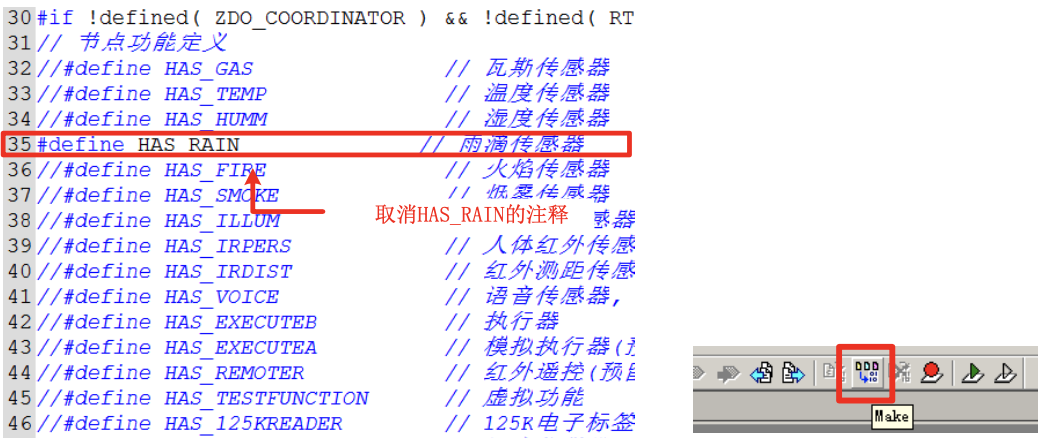


在工程目录结构树上方的下拉列表中，选择“EndDeviceEB”，如图 2.103 所示;



在“SAPP\_Device.h”文件中，取消“HAS\_RAIN”的注释，并保证其他的功能均被注释，如图 2.104 所示;

点击工具栏中的“Make”按钮，编译工程，如图 2.105 所示;



点击“Download and Debug”按钮，如图 2.106 所示;

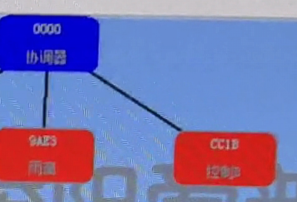
待程序下载完毕后，点击“Go”按钮，使程序开始运行，如图 2.107 所示;



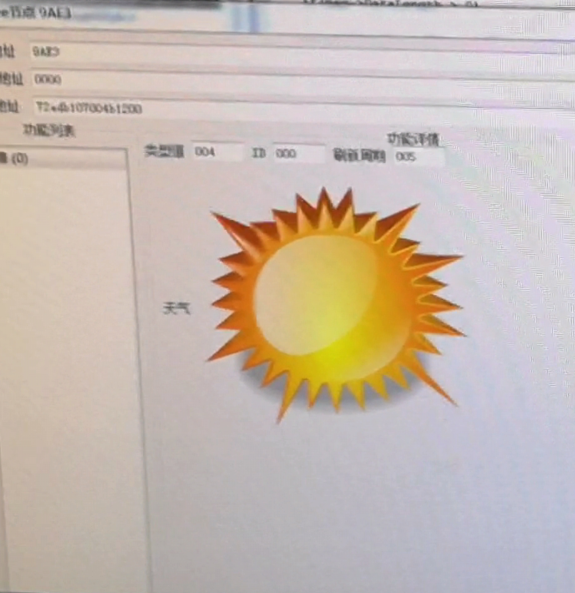
稍等片刻，可以看到雨滴传感器节点的两个 LED 灯同时闪烁，表示正确加入到协调器组建的 Zigbee 网络;

打开“Tools\CC25XX\Zigbee 调试助手”文件夹下的 ZSAPP Assistant.exe 程序， 在“串口”列表中选择协调器使用的串口号，在本例中，使用的是 COM7；

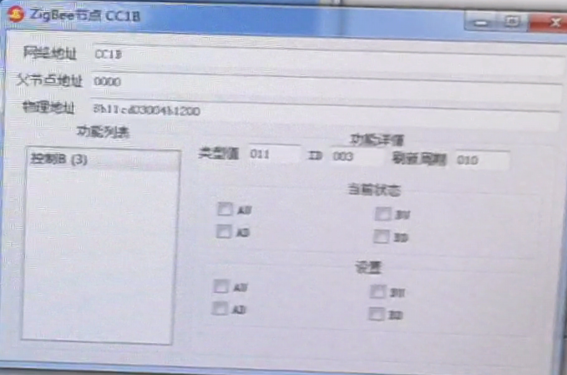
点击“打开”按钮，启动 Zigbee 网络助手，等待片刻，应当可以看到类似于图 2.108 所示的界面



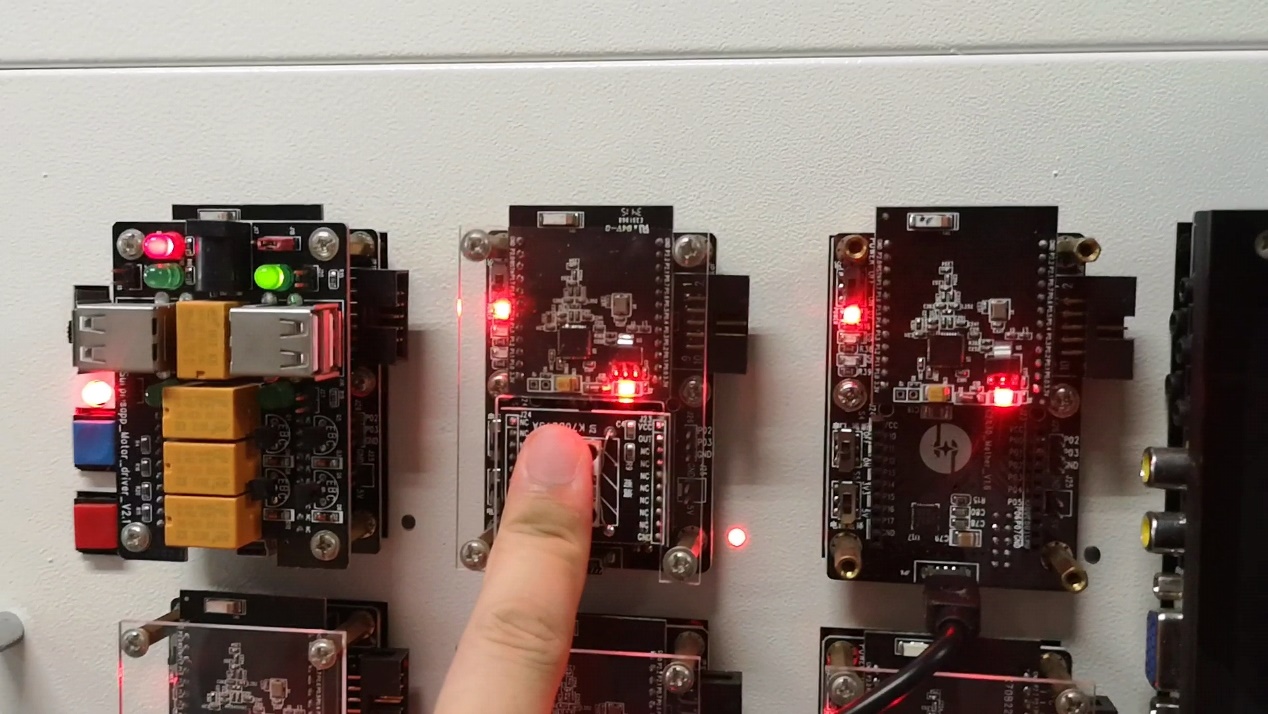
主界面中显示的是 Zigbee 网络的拓扑结构，其中，标有“雨滴”的红色节点即为雨滴传感器节点，点击雨滴传感器节点，可以打开所示的节点详细信息页面，如图 2.109 所示



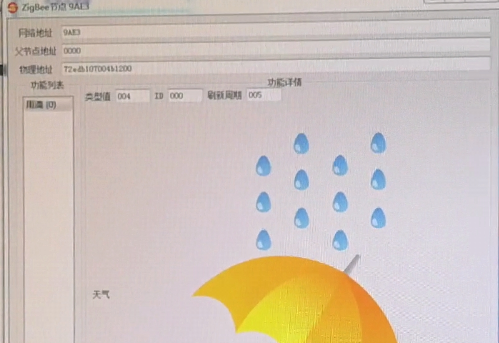
标有“控制 B”的红色节点即为执行节点，点击执行节点，可以打开图 2.110 所示的节点详细信息页面；在“当前状态”栏，显示的是四个继电器的当前状态；可以修改“设置状态”中的四个复选框的状态，并点击“应用”按钮，来控制四个继电器。



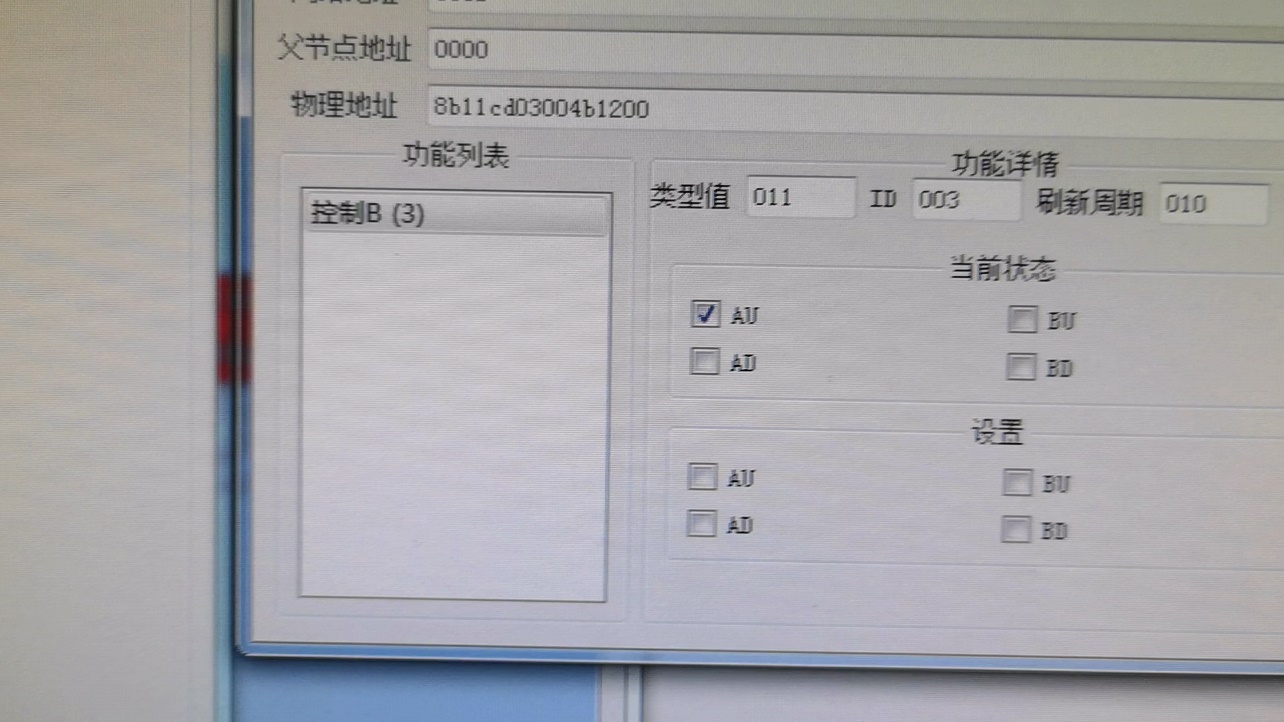
如图2.111用湿润的手按雨滴传感器后



雨滴传感器的状态发生改变，如图2.112所示



同时继电器的状态发生了改变



**5.2主要实现代码**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 雨滴传感器 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#if defined(HAS\_RAIN)

#define RAIN\_IO\_GROUP 0

#define RAIN\_IO\_BIT 0

void sensorRainResAvailable(struct ep\_info\_t \*ep, RES\_TYPE type, void \*res);

void sensorRainResAvailable(struct ep\_info\_t \*ep, RES\_TYPE type, void \*res)

{

if(type == ResInit)

{

HalIOSetInput(RAIN\_IO\_GROUP, RAIN\_IO\_BIT, Pull\_None);

HalIOIntSet(ep->ep, RAIN\_IO\_GROUP, RAIN\_IO\_BIT, IOInt\_Rising, 0);

}

//IO端口中断触发，中断源检测

if(type == ResIOInt)

{

uint8 RainValue = 1;

SendData(ep->ep, &RainValue, 0x0000, TRANSFER\_ENDPOINT, sizeof(RainValue));

}

}

void sensorRainTimeout(struct ep\_info\_t \*ep);

void sensorRainTimeout(struct ep\_info\_t \*ep)

{

uint8 value = HalIOGetLevel(RAIN\_IO\_GROUP, RAIN\_IO\_BIT);

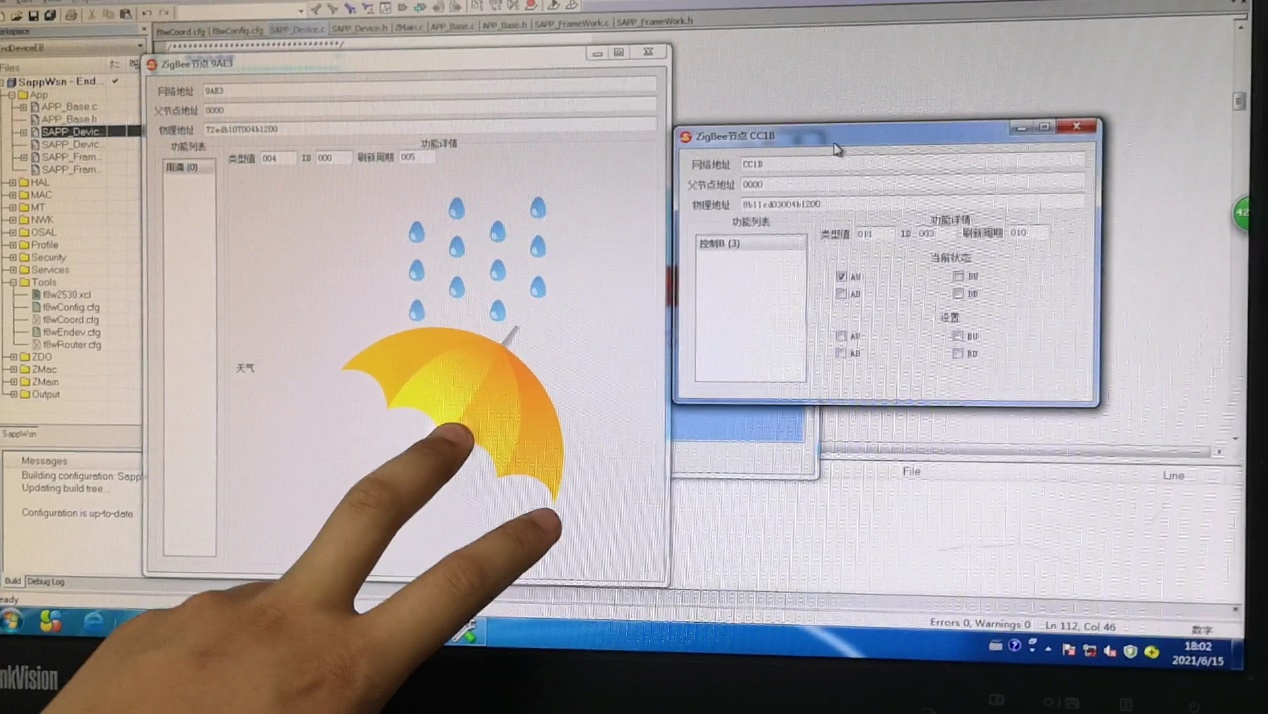
SendData(ep->ep, &value, 0x0000, TRANSFER\_ENDPOINT, sizeof(value));

SendData(ep->ep, &value, 0xcc1B, TRANSFER\_ENDPOINT, sizeof(value));

}

#endif

**5.3实验结果截图**



**5.4实验现象说明**

在“当前状态”栏，显示的是四个继电器的当前状态；用湿润的手按雨滴传感器后，雨滴传感器的状态由晴天变为雨天状态，同时通过控制执行器节点，修改“设置状态”中的一个复选框的状态，AU由未勾选变为勾选，完成了闭环控制