

# 物联网操作系统实验

## 实验一：mbed 开发环境的建立

### 一、实验目的

1. 学习 mbed 操作系统的安装和使用
2. 学会设置路径、新建程序
3. 学习开发板驱动程序的安装和连接；
4. 掌握调试程序的方法

### 二、实验仪器与设备

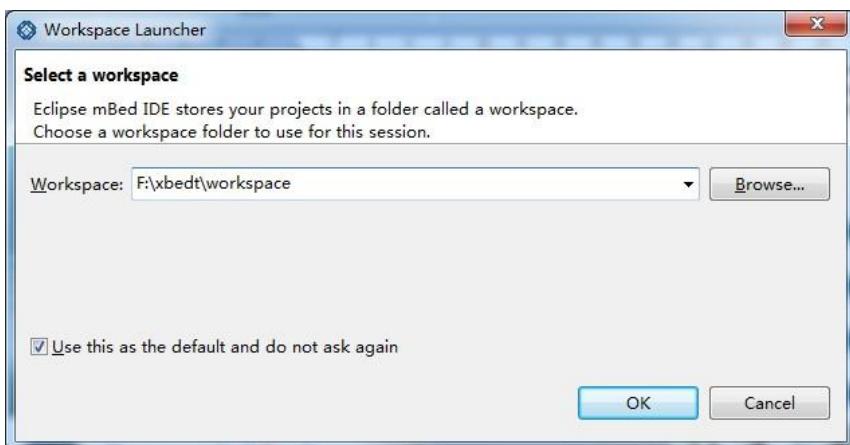
Mbed 开发板、连接线、计算机、相关软件

### 三、实验内容

#### (1) SMeshStudio 的安装和使用

SMeshStudio 是配合 xbed LPC1768 使用的离线 mbed 开发环境，当然，它也可以用来 开发任何支持 gcc 编译器的 mbed 开发板，而且它还可以用来开发 Arduino（一种和 mbed 类似的快速开发原型，建立在 AVR 架构之上）和 Contiki（当前流行的物联网操作系统之一）应用。SMeshStudio 基于 eclipse 开发，所以用户只要安装了 JDK 运行环境，直接复制 就能使用。

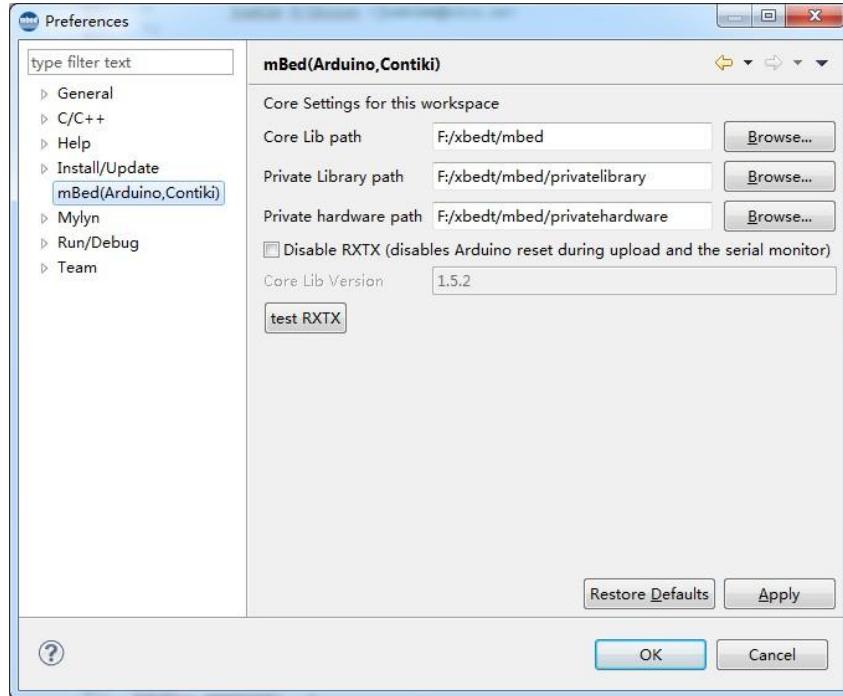
同任何 Eclipse 一样，在你首次运行 SMeshStudio 时，它会让你选择你的工作目录，此 工作只需要设置一次，你只要选择你习惯的目录即可，如果选择的目录不存在，它会自动 创建一个，日后你所有的代码都会存在在此目录中（有可能你下载的版本已经把工作目录 默认设称.. \workspace，也就是和 eclipse 平级的 workspace，所以系统会自动跳过此过程），



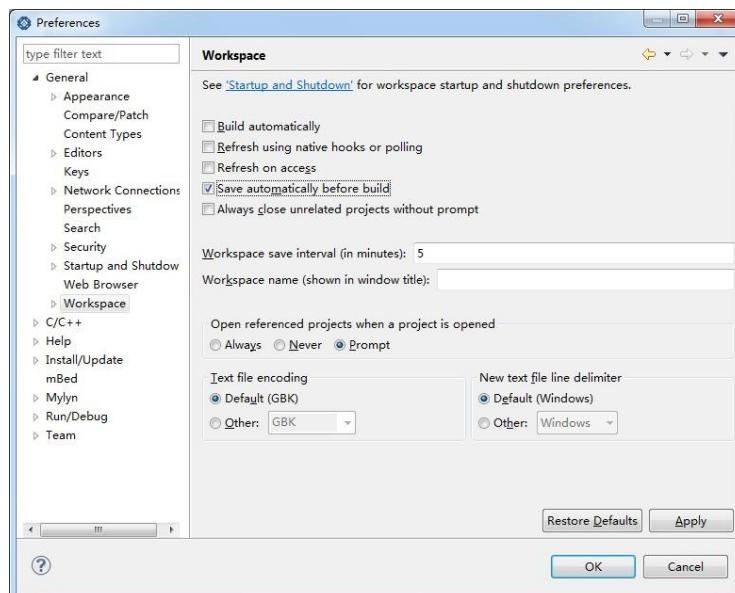
图表 1-11 设置 SMeshStudio 工作目录

进入 SMeshStudio 后，你需要先完成一些配置工作才能更好地使用，首先需要配置 mbed SDK 的安装目录，修正后的 mbed SDK 已经放在 SMeshStudio 的安装目录下，你需要 通过使用 Window——>Preferences 菜单打开配置界面

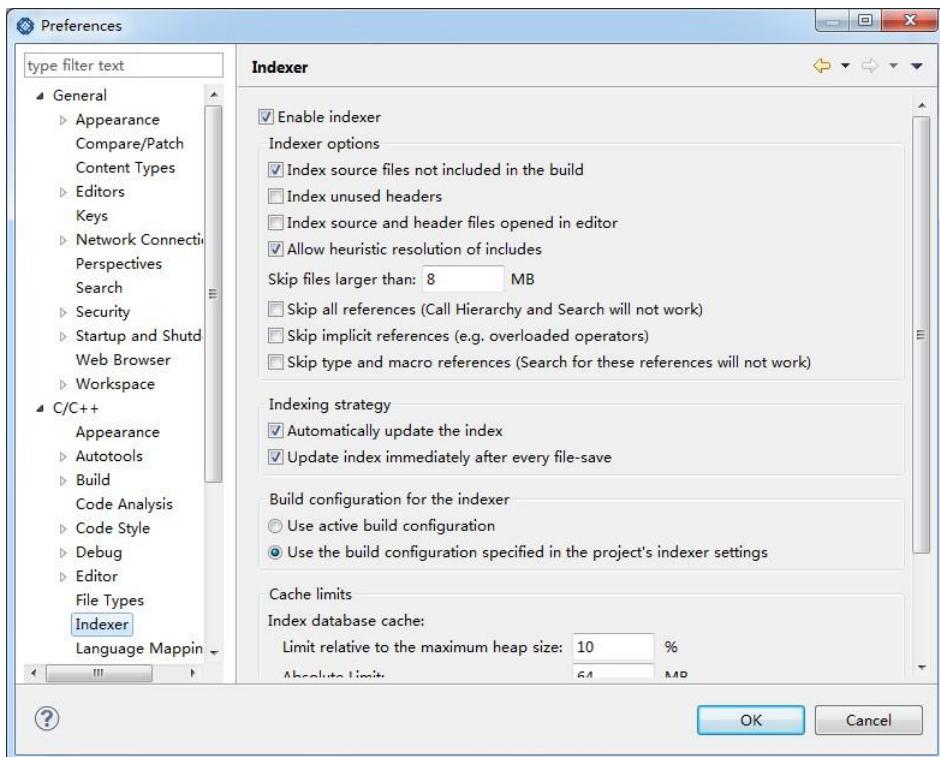
, 然后选择 mbed 配置项, 参考配置如下, 你只需要把 f:\xbedt 改成你自己的 SMeshStudio 安装目录即可(新版本的 SMeshStudio 可以完成自动配置, 所以这一步也可以省略),



图表 1-12 设置 SMeshStudio 系统目录  
另外我们还建议做以下配置 (编译器自动保存和索引更多文件),

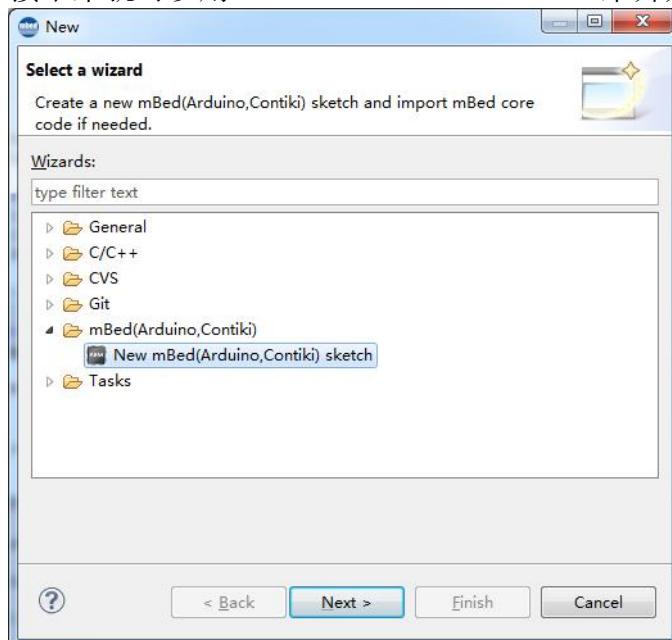


图表 1-13 设置 SMeshStudio 编译前保存

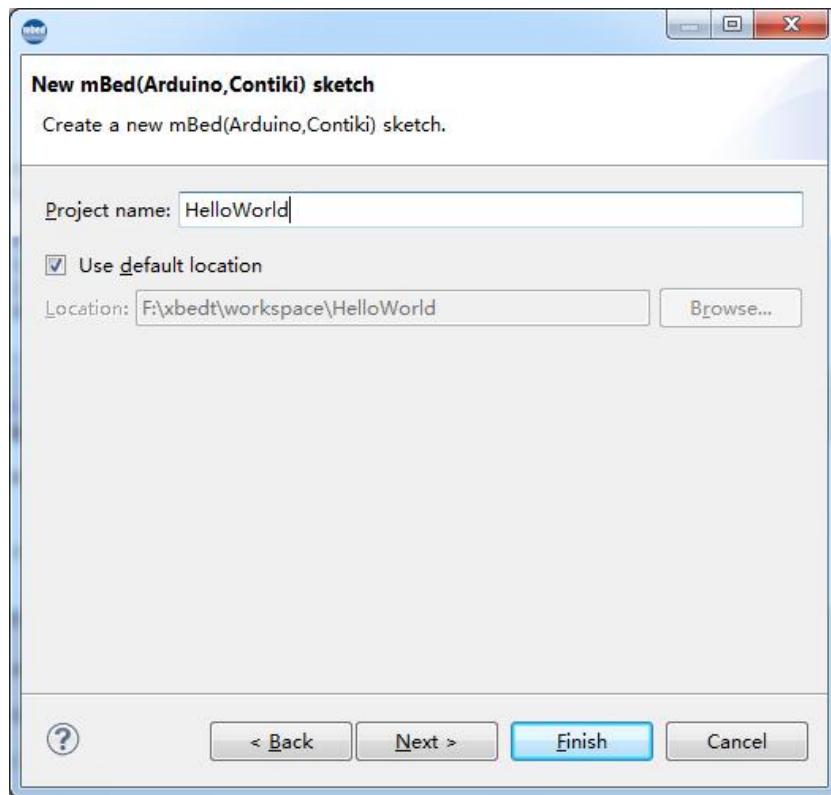


图表 1-14 设置 SMeshStudio 索引内容

接下来就可以用 File——>New——>Other 来开始创建 mbed 工程了：

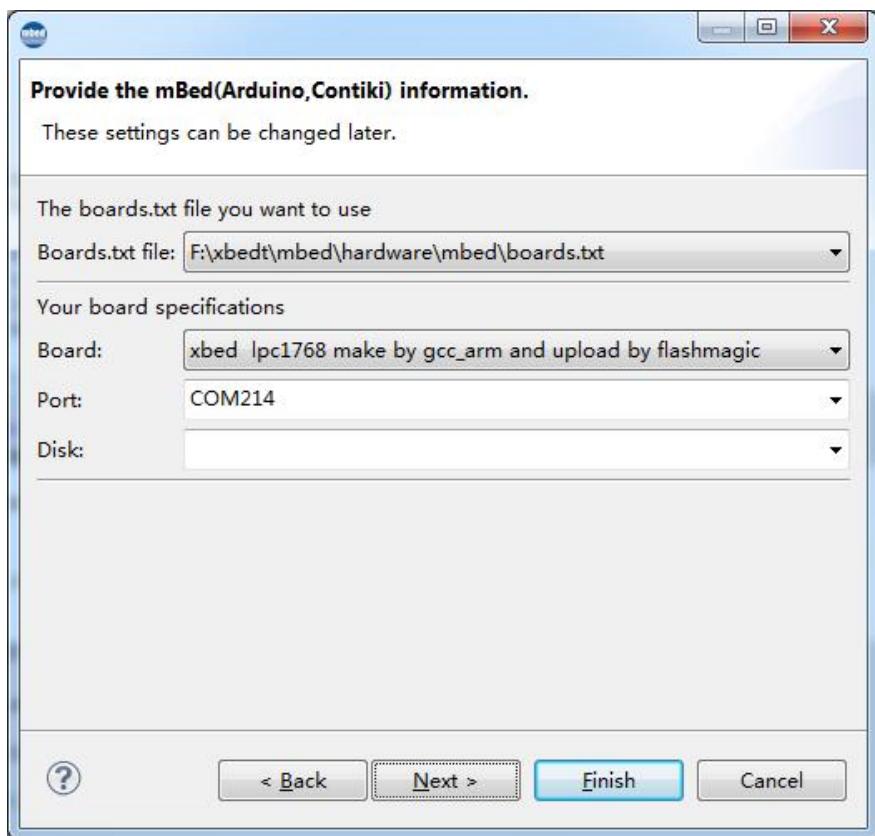


图表 1-15 SMeshStudio 工程向导——选择 Eclipse 项目类型  
选择 New mbed sketch 后点击 Next 进行下一步：

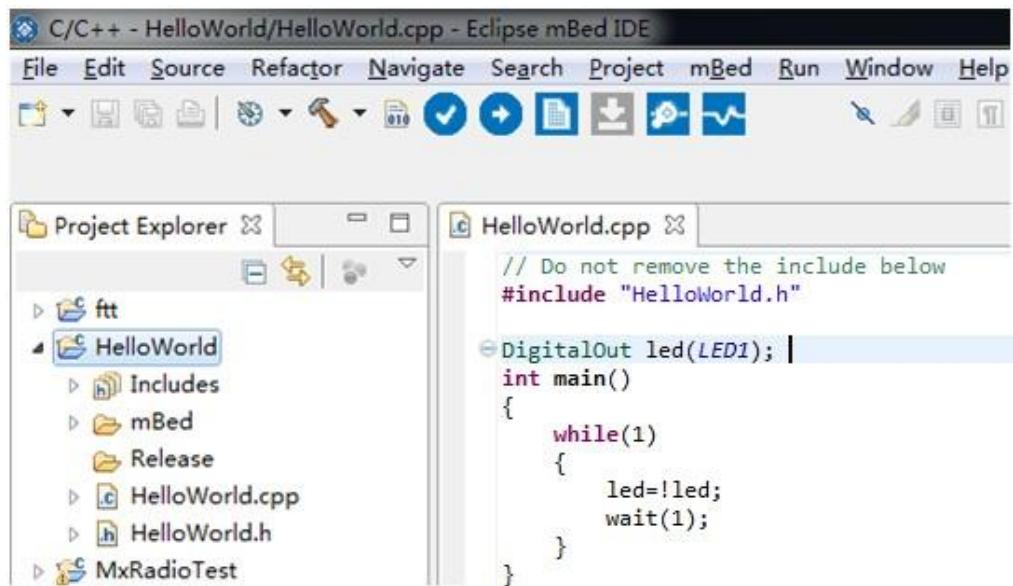


图表 1-16 SMeshStudio 工程向导——设置  
项目名

输入项目名称后点击 Next 进行开发板的选择，这步的选择非常关键。SMeshStudio 支持 Contiki, Arduino, mBed 三种软件开发类型，系统将自动根据你选择的开发板确定你的工程类型。系统的判断依据是：如果开发板名称中含有 contiki，则为 Contiki 工程，如果含有 bed，则为 mBed 工程，否则为 Arduino。在这里，你首先选择你使用的开发板文件，我们这里选择 mbed 目录下的文件，然后再选择你当前使用的 mbed 开发板，xbed 使用串口上载，只需要选择串口号即可，mbed 使用文件复制方式上载，只需要选择磁盘号即可。



图表 1-17 SMeshStudio 工程向导——设置开发板类型  
点击 Finish 完成，就可以在项目列表里面看到你刚才创建的工程了。



图表 1-18 SMeshStudio 工程向导——HelloWorld 代码

## 2、xBed LPC1768 的使用

### 安装 cp2104 驱动

xBed LPC1768 使用 cp2104 来上载程序及串口调试，所以用户首先需要从 silabs 官网下载并安装驱动程序，参考链接是（日后有可能会更新）

<http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>，安装成功后，使用 micro usb 连接 xbed LPC1768（芯片下面的 micro usb 口）和电脑，你就可以在电脑的设备管理器中找到 xbed lpc1768 虚拟出来的串口了。



图表 1-19 xbed LPC1768 虚拟出来的串口

### ● 编写 HelloWorld

按照前面描述的 IDE 使用方法新建 HelloWorld2 工程，并编写以下代码：

```
DigitalOut led1(LED1);
DigitalOut led2(LED2);
Serial pc(USBTX,USBRX);
int main()
{
    pc.baud(115200);
    while (1)
    {
        led1=0; led2=0;
        pc.printf("Hello World!\n");
        wait(1);
        led1=1; led2=1;
```

```
    wait(1);
```

```
}
```

如果用户设置没有错误的话，用户就可以使用工具栏中的  按钮进行编译 使用

 进行上载，编译成功输出如下：

```
Building target: HelloWorld Printing
size:
"f:/xbedt/mbed/hardware/mbed/tools/gcc_arm/bin/arm-none-eabi-size" -A
"F:/xbedt/workspace>HelloWorld\Release\HelloWorld.elf"
F:/xbedt/workspace>HelloWorld\Release\HelloWorld.elf      :
section          size        addr
.text            33984        0
.ARM.exidx       8        33984
.data            228      268435656
.bss             944      268435884
.heap            2048     268436832
.stack_dummy     3072     268436832
.ARM.attributes   41        0
.comment         112        0
.debug_frame     5768        0
.stabstr          118        0
Total           46323


```

```
Finished building target: HelloWorld
```

```
11:44:39 Build Finished (took 763ms)
```

上载成功的输出如下：

```
Starting upload
using mbed loader
```

```
Launching fm COM(4, 115200) DEVICE(LPC1768, 0.000000, 0)
HARDWARE(BOOTEXECRTS, 50, 100) ERASE(DEVICE, PROTECTISP)
HEXFILE(F:\xbedt\workspace\HelloWorld\Release\HelloWorld.hex, NOCHECKSUMS, NOFILL,
PROTECTISP) VERIFY(F:\xbedt\workspace\HelloWorld\Release\HelloWorld.hex,
NOCHECKSUMS)
Output:
Flash Magic Version 7.66.3343
8051/XA Driver Version 3.04.3296
```

ARM UART Driver Version 3.11.3302  
ARM Cortex UART Driver Version 4.63.3312  
ARM Ethernet Driver Version 2.05.3296  
ARM Cortex Ethernet Driver Version  
2.07.3296 ARM CAN Driver Version  
2.11.3296  
(C) Embedded Systems Academy 2000-2013 All rights reserved  
NON PRODUCTION USE ONLY  
Connection failed: comms error (Unable to communicate. (in use)) fm  
finished  
upload done

此时，你可以看到 xBed LPC1768 的两个 led 等 1 秒变换一次，同时串口调试工具有输出。