

# 科创 3C 实验指导书

声明：本实验指导书基于 Windows 7 操作系统，事实上，android 的开发也可以在 MAC OS 上进行，只有安装软件过程不同。MSP430 我目前无法在 MAC OS 上开发，当然用虚拟机可以解决。

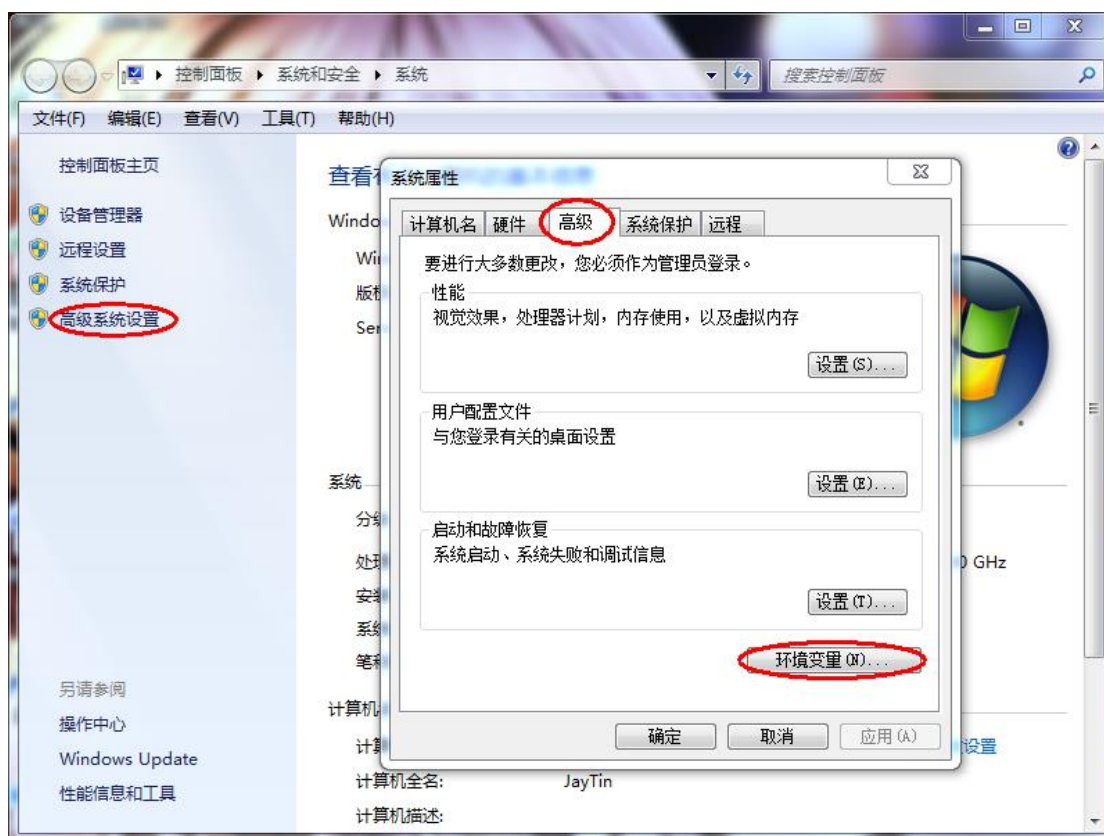
## 一、android 开发

众所周知，android 应用的开发使用的是 Java 语言，一般结合 Eclipse 的开发环境和其 ADT 插件，使用 Google 公司提供的 android SDK 进行开发。

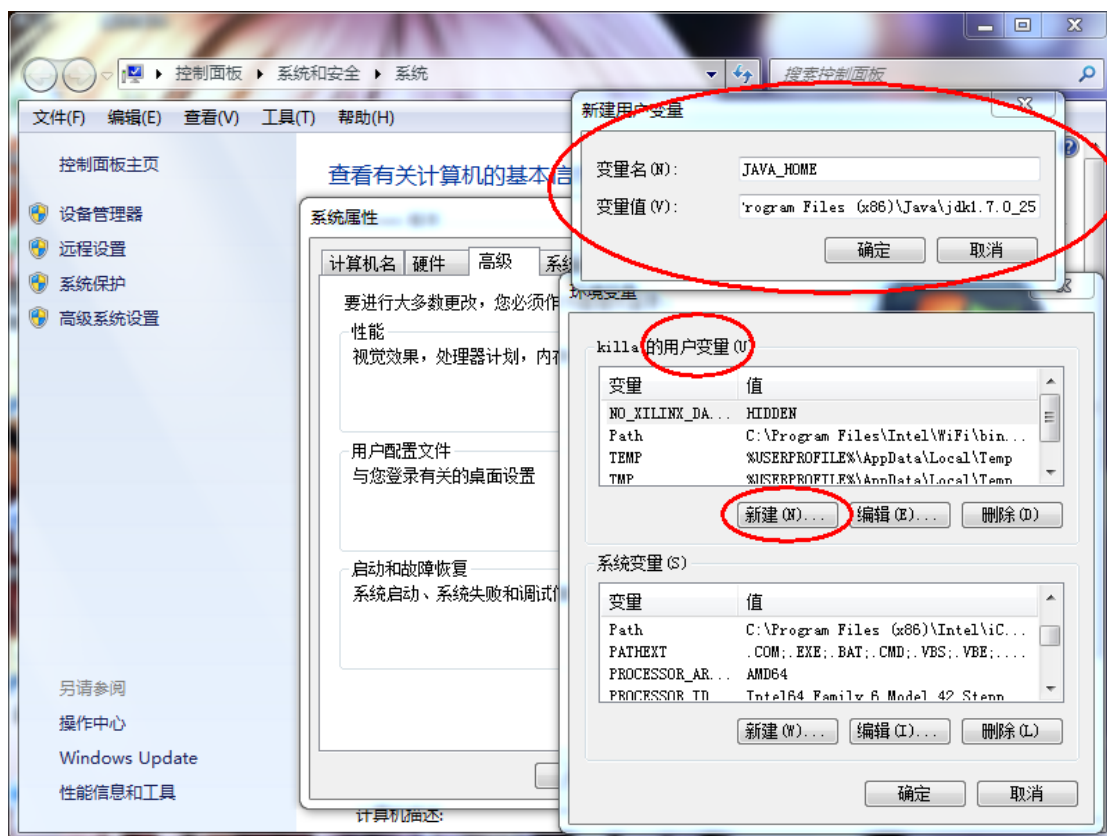
### 1、Java 的安装

Java 是什么这里就不多说了，但是请记住，windows 是要安装 JDK 后才能使用 Java 的，这点与 Python 相同。具体安装什么呢？其实是 Java 的编译器、Java 虚拟机等（Java 实际上是运行在 Java 虚拟机上的）。JDK 的官方下载地址：[http://www.java.com/zh\\_CN/download/](http://www.java.com/zh_CN/download/)，下载得到的安装包按照正常程序的安装方式就行，安装完后还要进行环境变量的配置，这是为了让操作系统得知 Java 的存在位置以便调用。环境变量的设置方法如下：

- 1、用鼠标右击“计算机”，选择“高级系统设置”->“高级”->“环境变量”。  
如下图：



- 2、在用户变量中点击“新建”然后新建如下变量：



变量名: JAVA\_HOME, 变量值: 你的安装路径\Java\jdk 版本号。

3、接下来查看系统变量，如果其中有 classpath 和 path 变量，则在其值后面分别添加：

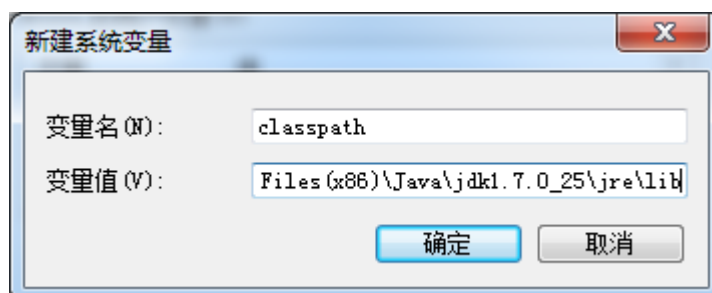
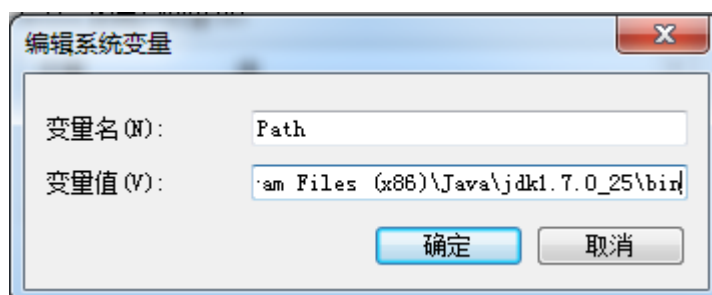
;你的安装路径\Java\jdk 版本号\jre\lib

和

;你的安装路径\Java\jdk 版本号\bin

如果没有则新建，值为去掉分号的上述值。

如图：



4、至此，Java 就安装好了，如果想检验 Java 是否安装成功，则在命令提示符里面输入如下命令：`java -version`，如果出现 java 的版本信息，则说明安装成功。

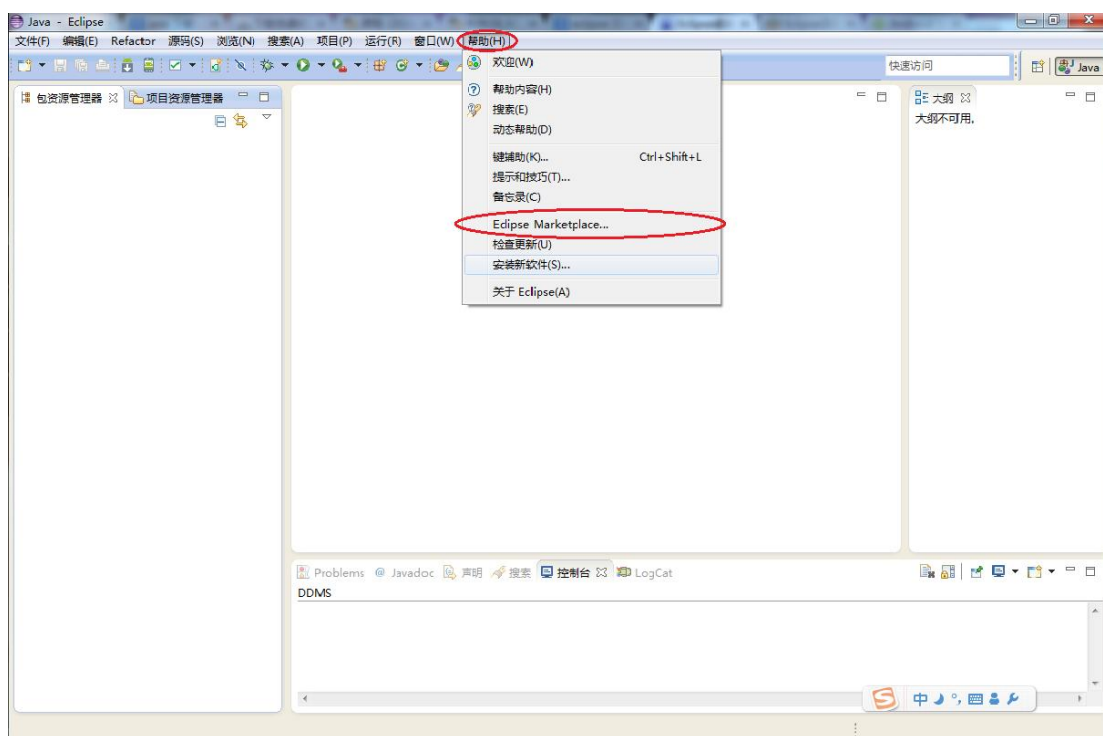
## 2、Android SDK 的安装

Android SDK 是开发 android 应用必不可少的工具，它里面包含了 android 操作系统的 jar 包等一些工具。下载地址：<http://developer.android.com/sdk/index.html>，下载完成后只需要把解压后的文件夹放在任何你想的位置，然后记住地址就好。

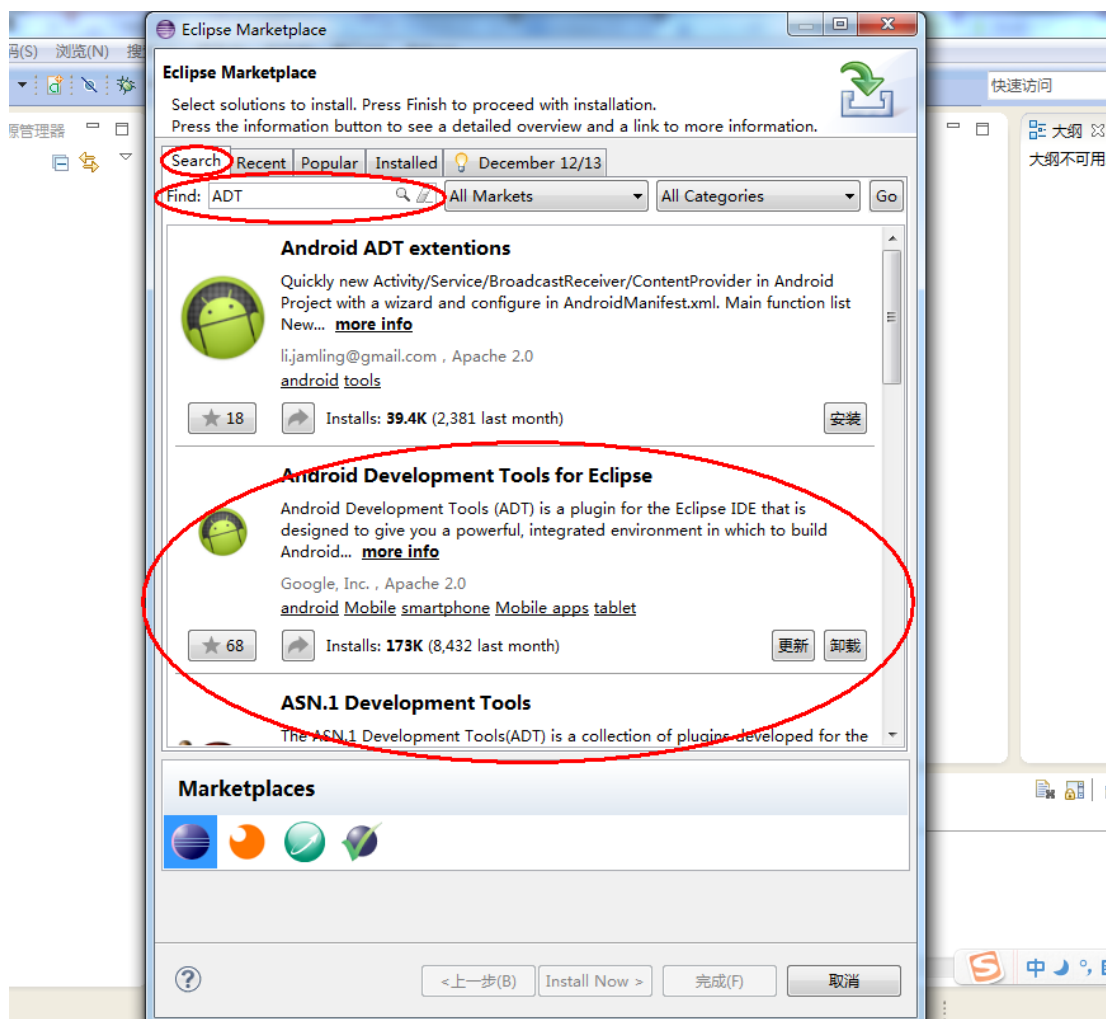
## 3、Eclipse 和 ADT 的安装

Eclipse 是一款非常强大的 IDE，主要用来编写 Java 代码，它的自动完成和可扩展性非常强大。下载地址：<http://www.eclipse.org/downloads/>，下载后无需安装，直接解压到你想要的安装目录，运行里面的可执行文件即可。如果想要汉化，参考百度经验 <http://jingyan.baidu.com/article/e75057f28401a8ebc91a899e.html> 官方给出的汉化包非常完美。

ADT 是 Eclipse 上的一个插件，用来方便程序员建立、导入 android 工程等，安装方法如下：打开 Eclipse 的“帮助”->“Eclipse Marketplace”如图：



然后搜索安装 ADT 如图：

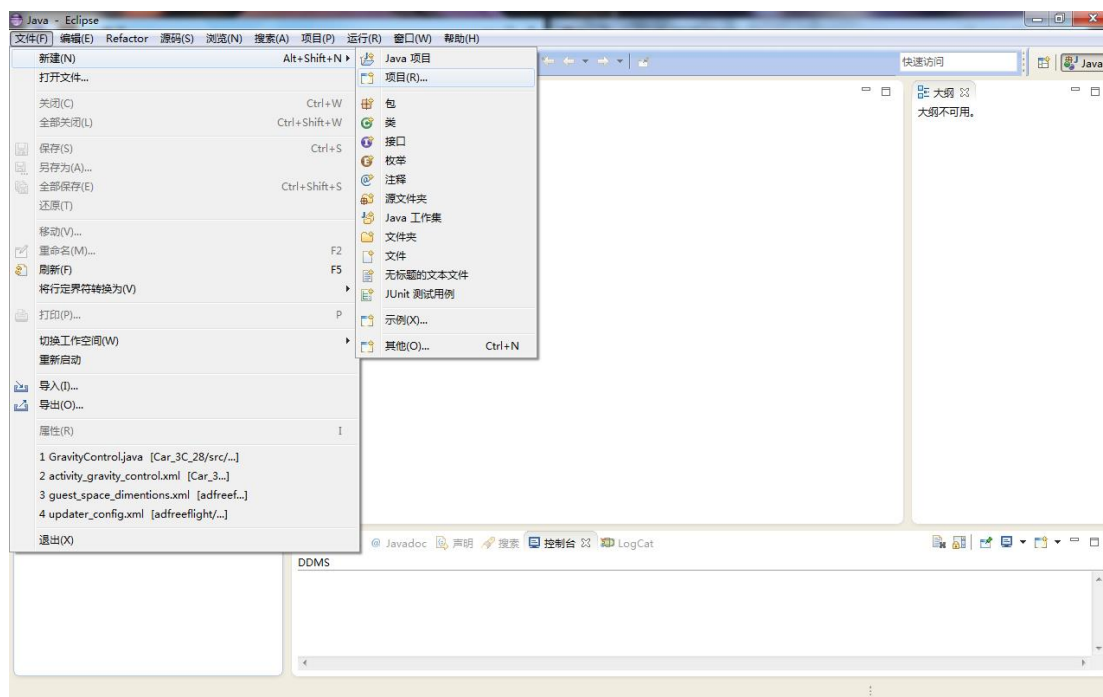


至此 Eclipse 的安装就完成了。

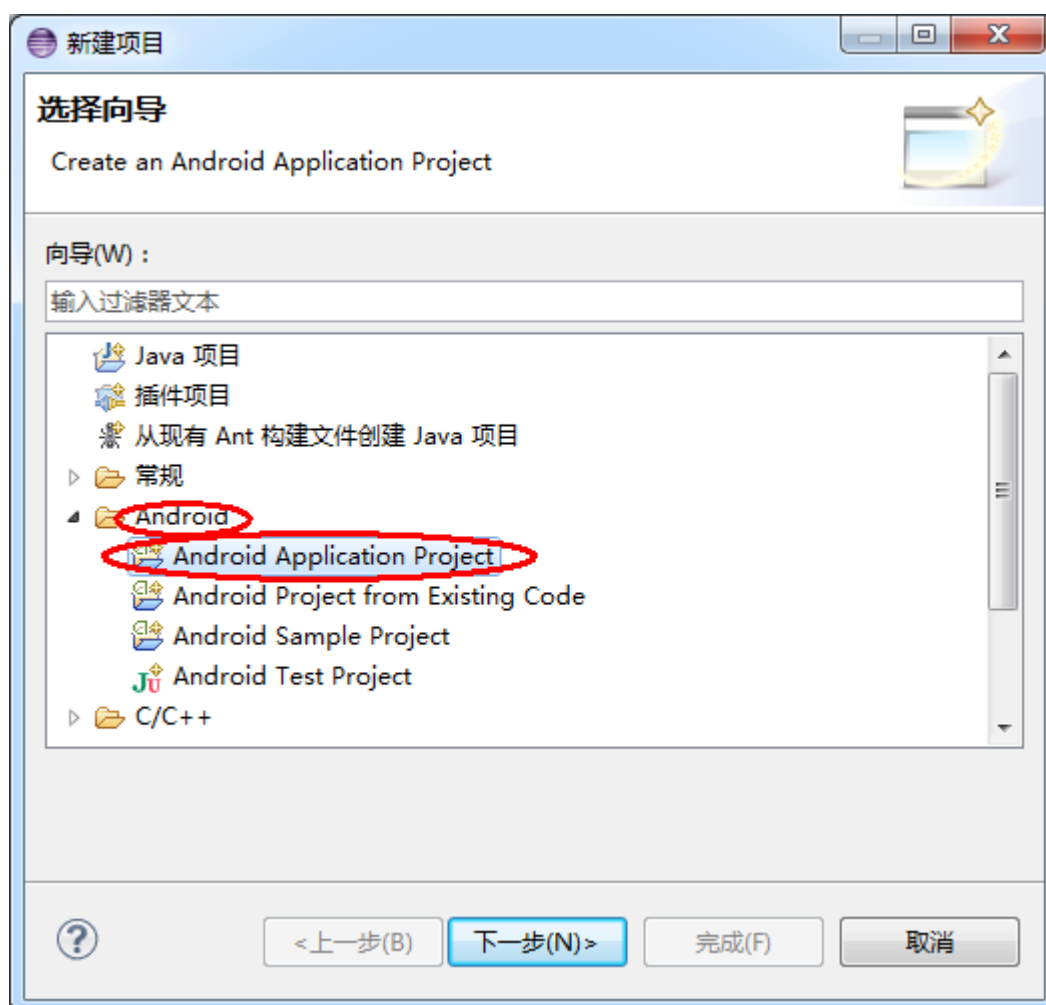
开发 android 应用不一定非要使用 Eclipse，还有其他 IDE 可选。Google 官方也给出了自己修改后的专门用来写 android 应用的 Eclipse，里面自带 ADT。嫌这种方法麻烦的同学可以去搜一下。

#### 4、第一个 android 程序的编写

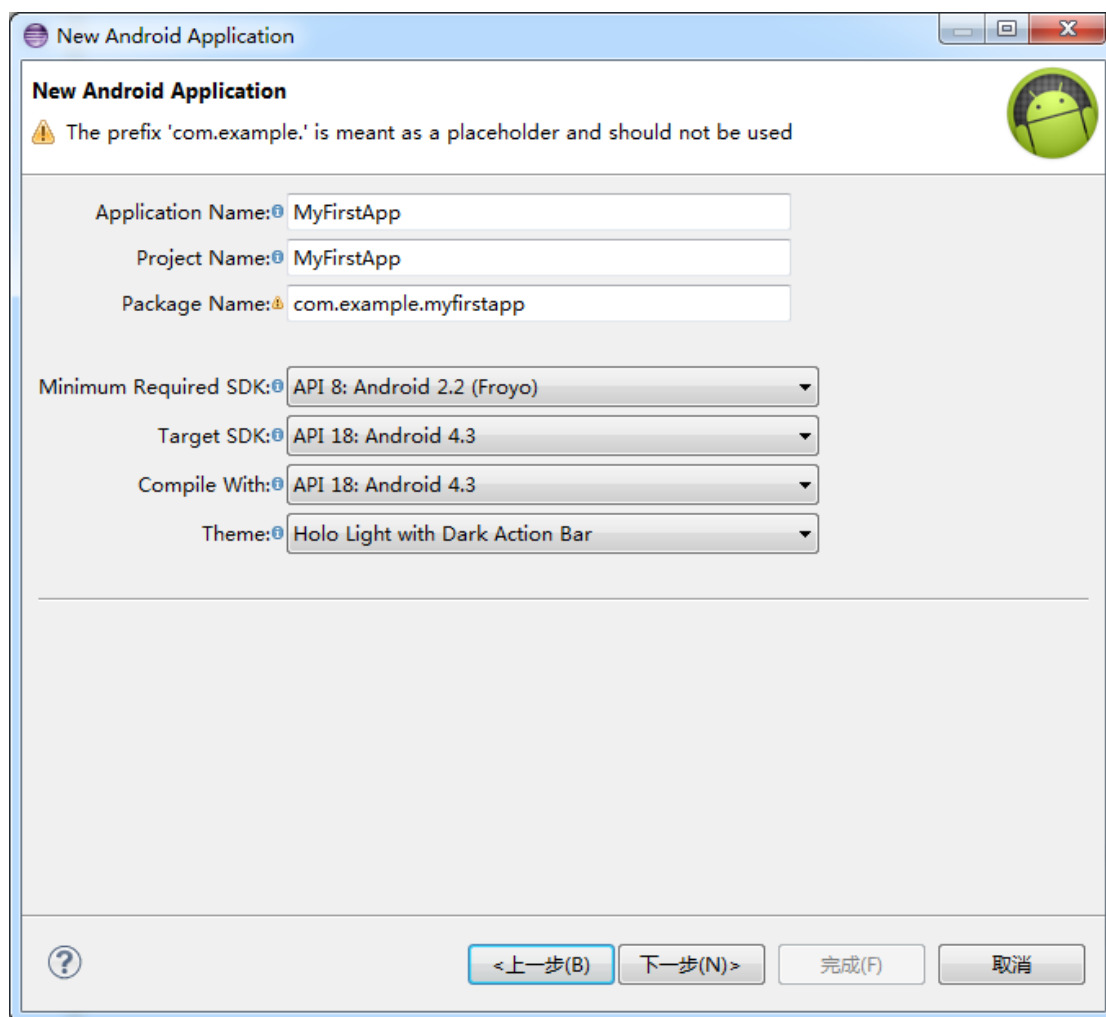
点击“文件”->“新建”->“项目”如图：



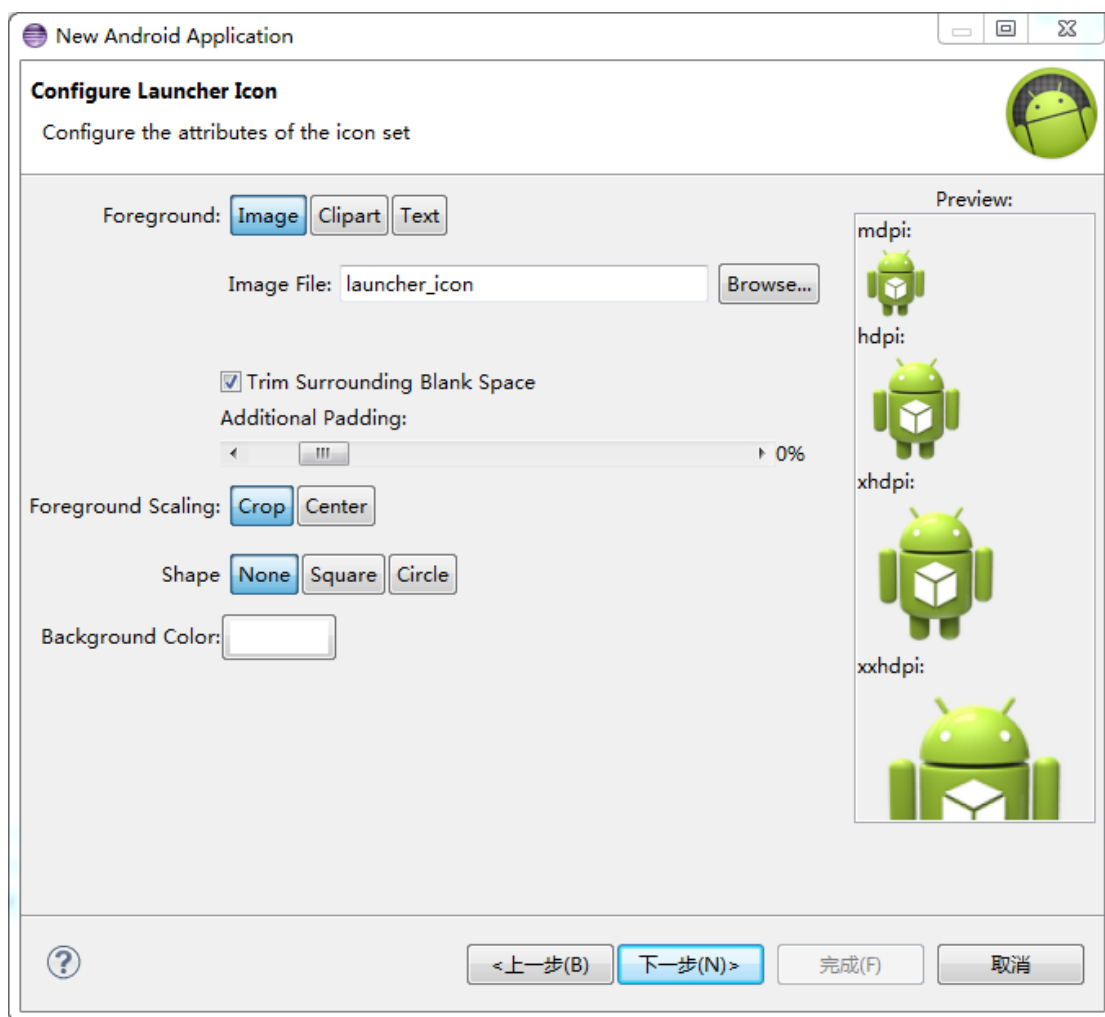
点击 android 中的 android application project 然后下一步，如图：



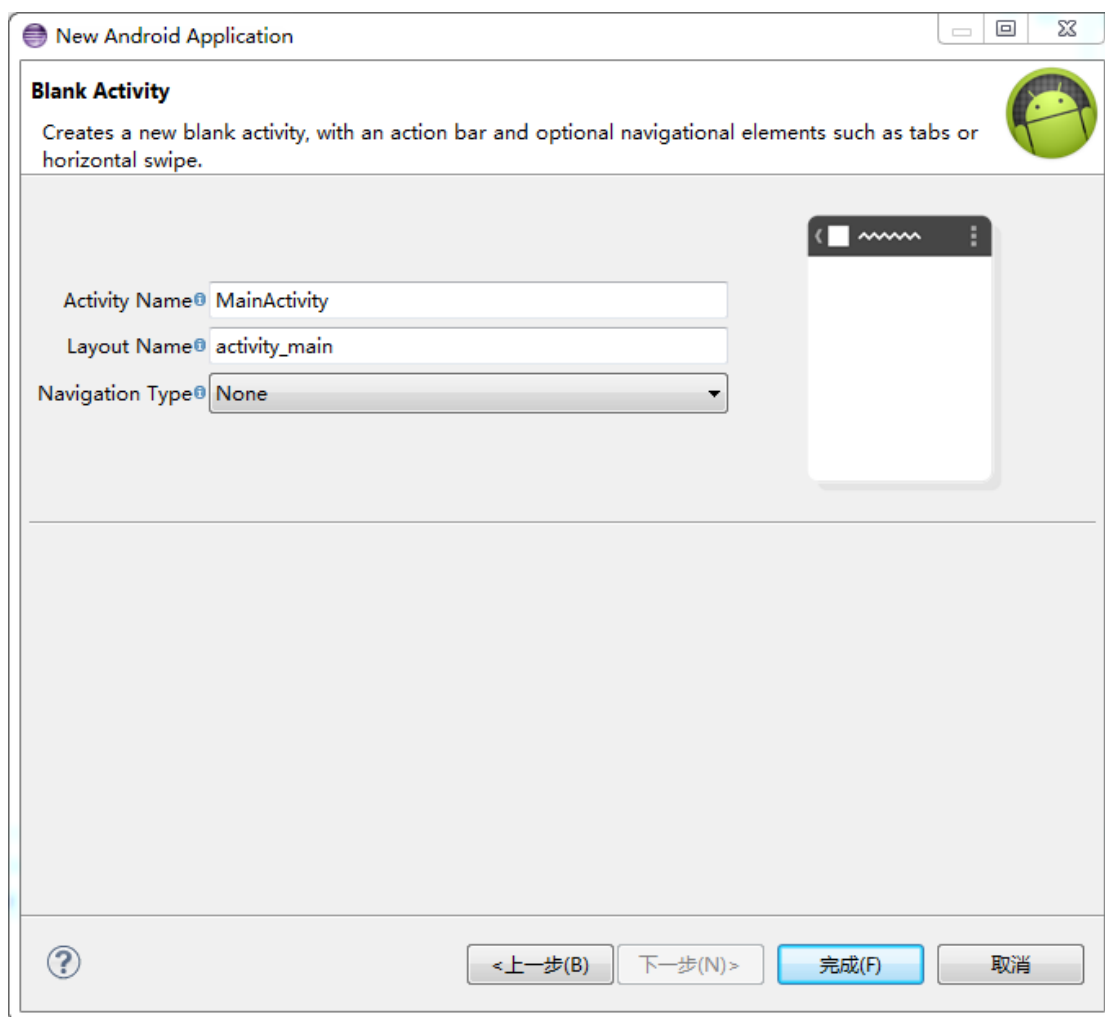
填写应用的名称、工程名称、包名等信息，应用名称将显示在你的手机上。如图：



一直点下一步到如图页面，这里你可以选择你应用的图标：



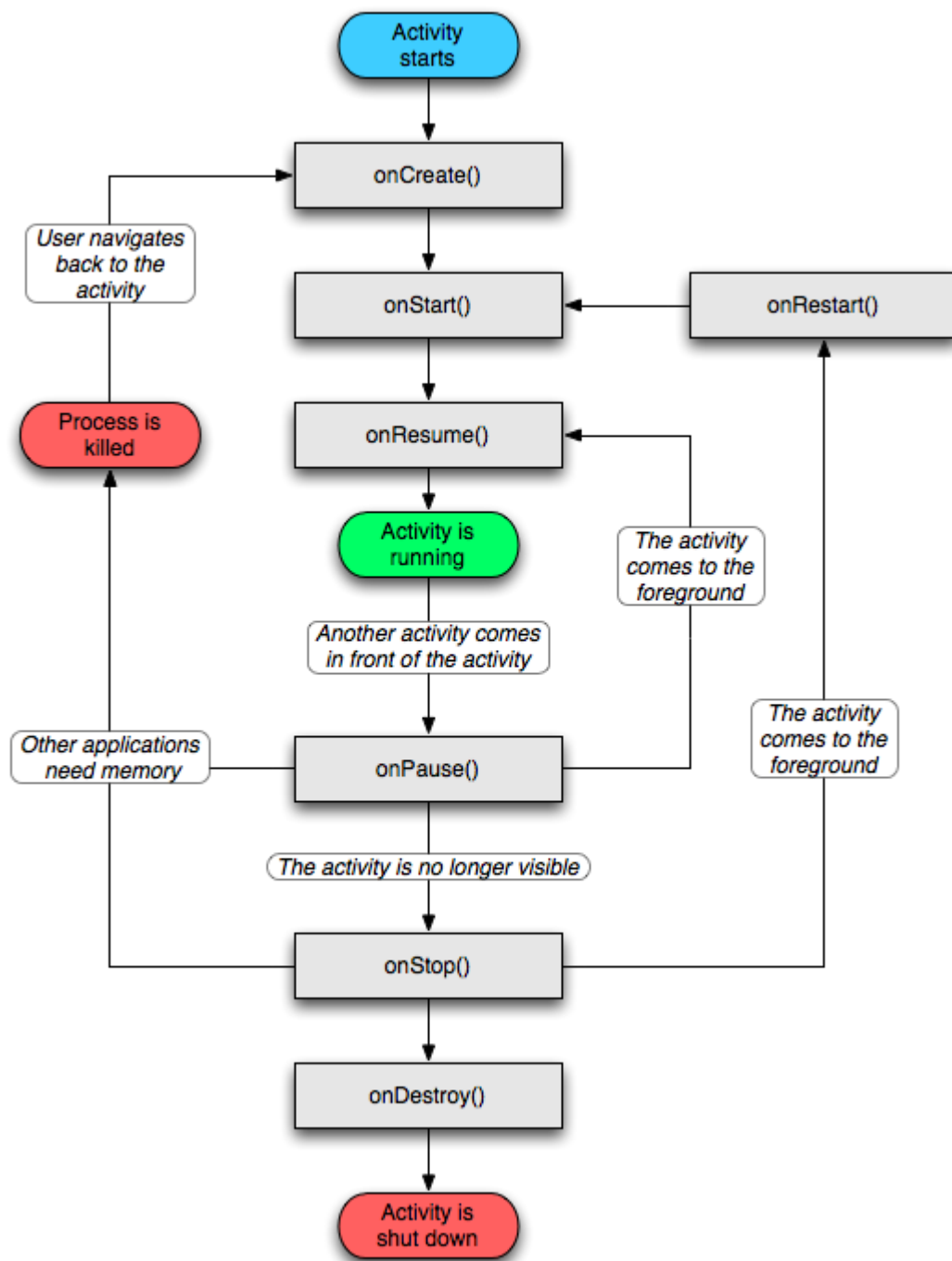
继续点下一步，直到遇到如图的界面，一个 **ACTIVITY** 你可以理解为一个窗口，**LAYOUT** 指的是窗口的界面，这里可以修改窗口的名称和对应的界面 XML 文件的名称。



点击“完成”，我们就建立了一个 Android 工程

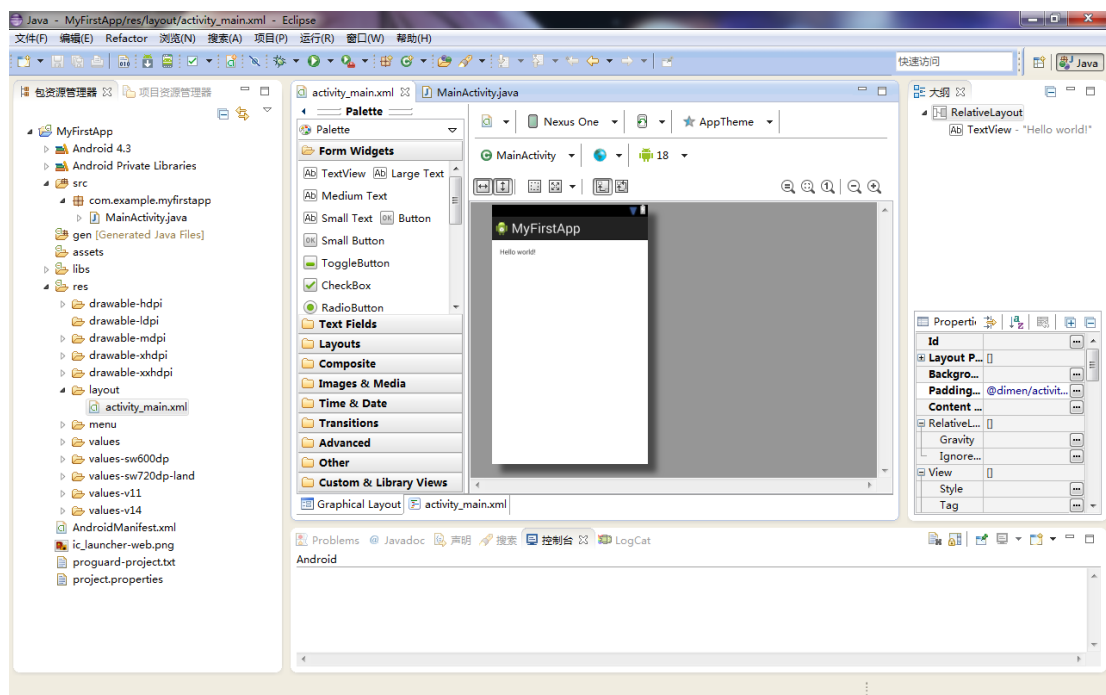
Android 的程序没有 main 函数，每个窗口是通过几个流程来执行的，流程图如下：





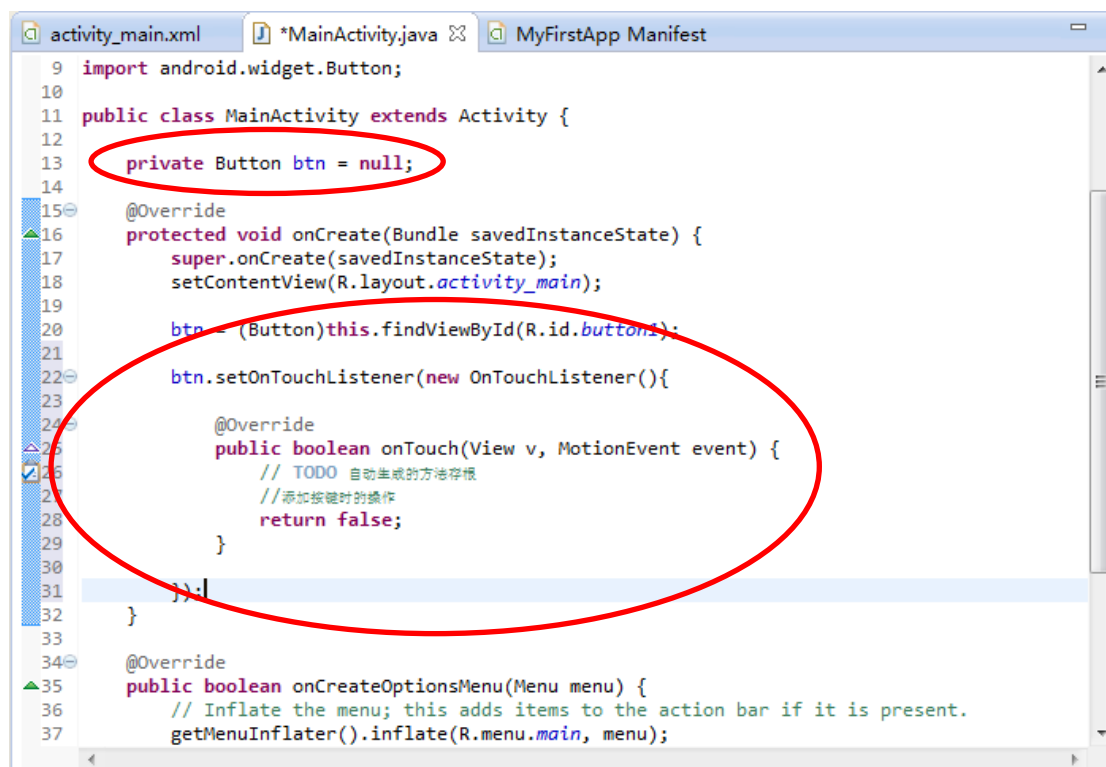
每个流程对应一个函数，我们可以重载这些函数并定义在这个流程中要做哪些工作。

Android 程序的界面是由 xml 格式的文件来定义的，我们可以通过改写 xml 文件或者使用 ADT 提供的 xml 解析器对界面进行图形化的修改，如图：

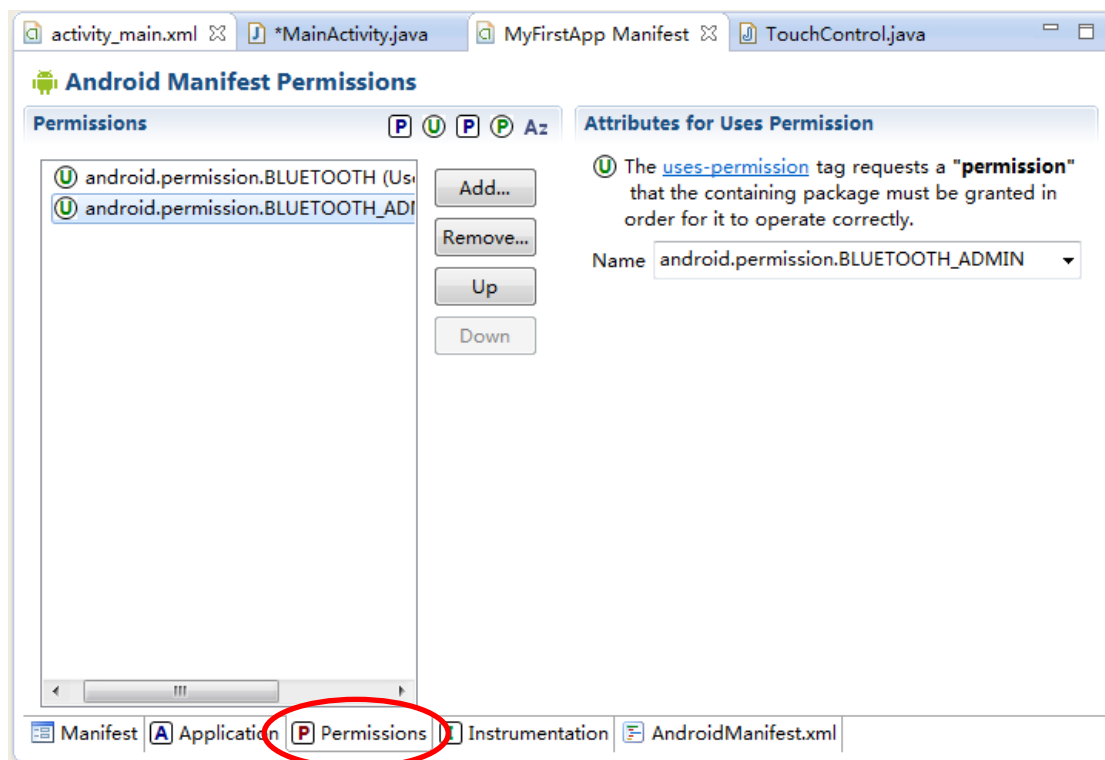


可以通过拖动左边的控件到窗口上来添加控件,通过右边的对话框进行控件的参数设置,本课程主要用到的控件有 `Button`、`ImageView` 等。

在添加控件后要在 `java` 文件中添加控件对象,再对对象进行操作。如图:



手机蓝牙模块的使用必须先声明权限,声明在 `AndroidManifest.xml` 中,需要添加两条 `Uses permission` 声明。如图:



代码中使用蓝牙传输信息需要定义 `BluetoothAdapter` 对象，使用此对象获得手机蓝牙适配器，然后从已经配对的设备中找到需要的设备（HC-06 要先用手机自带的蓝牙和手机进行配对），赋给 `BluetoothDevice`，然后获得套接字 `BluetoothSocket`，才能传输信息。代码如下：

```
private BluetoothAdapter bta = null;
private BluetoothSocket bts = null;
private BluetoothDevice btd = null;
private static final UUID MY_UUID =
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
bta = BluetoothAdapter.getDefaultAdapter();
if(bta == null){
    finish();
    return;
}
if(!bta.isEnabled()){
    finish();
    return;
}
Set<BluetoothDevice> devices = bta.getBondedDevices();
for(int i = 0 ; i < devices.size() ; i ++){
    BluetoothDevice device = (BluetoothDevice)
        devices.toArray()[i];
    if(device.getName().equals(Carname)){
        btd = device;
        break;
    }
}
```

```

    }
}
if(btd == null){
    finish();
    return;
}
try {
    bts = btd.createRfcommSocketToServiceRecord(MY_UUID);
} catch (IOException e) {
}
bta.cancelDiscovery();
try{
    bts.connect();
} catch (IOException e){
    try {
        bts.close();
    } catch (IOException e1) {}
}
}

```

至此，就可以利用BluetoothSocket获取输出流进行输出了。代码如下：

```

try {
    OutputStream ops = bts.getOutputStream();
} catch (IOException e1) {
    e1.printStackTrace();
}
try {
    ops.write("你想发送的东西".getBytes());
} catch (IOException e) {
    e.printStackTrace();
}
}

```

如想使用重力感应模块，则需要导入Sensor包，寻找到想要使用的Sensor然后调用相应的函数即可，此过程与蓝牙的使用相似，故不再赘述，同学们可以自己去网上查阅资料。

## 5、调试建议

运行你的程序的时候可以将你的手机接在电脑上，然后点击“运行”



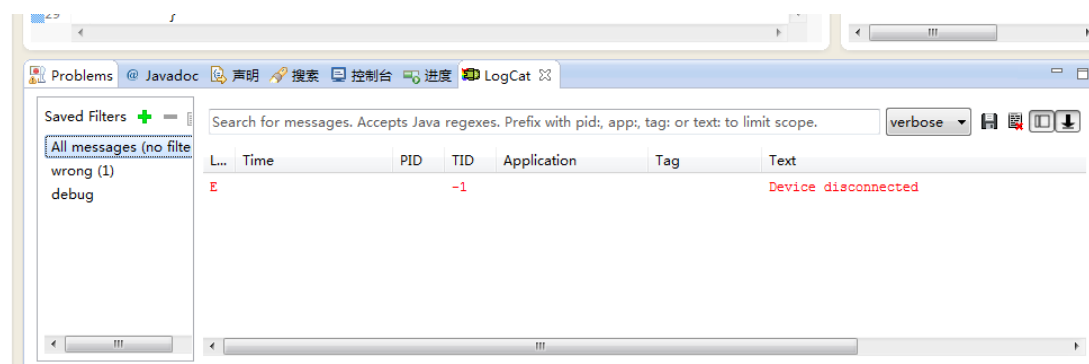
你的程序会自动安装在你的手机上运行。

如果不连接手机，则会在android模拟器上运行你的程序，如图：



第一次运行需要创建一个AVM（Android Virtual Machine），可根据自己的喜好按提示创建。

Android程序调试时无法输出信息到显示屏上，故习惯于在屏幕上cout信息进行调试的同学可能会很痛苦，这里建议大家使用Log函数输出日志，使用方法为Log.i(“一个任何你希望的字符串标签”，你要输出的内容如变量的值);这条语句可以添加在程序的任何地方。要查看输出的信息，可在下面的logcat窗口根据字符串标签进行筛选查看。如图：



## 二、MSP430 单片机开发

相信参加过科创 2A 或 2B 的同学们对 MSP430 并不陌生了，开发环境可以是官方的 CCS 或是更加轻便的 IAR。安装过程很简单，如果安装 CCS9 及以上版本则无需另外安装驱动，安装 IAR 还需再安装 MSP430 单片机的驱动。

这里不再详细介绍 430 的开发，只要大家仔细看看小车原理图，弄懂每个管脚是用来干什么的即可。小车的行动控制主要通过 P2.1、P2.2、P2.4、P2.5 控制电机，P1.6 控制舵机实现。舵机的控制是通过 PWM 波来实现的，给定一个固定占空比的 PWM 波足够长的时间，舵机便会转到一个固定的位置不再变化。旋转的角度和方向由占空比决定，一般以占空比为 50% 的波作为初始正方向的基准波。**舵机使用需注意给定足够长时间的 PWM 波，小车方向旋转到位后应停止 PWM 的供给以免烧坏舵机。**

430 单片机 PWM 波的生成是由 TimerA 来完成的，使用之前需将 TimerA 的功能设置为 PWM 模式，具体请看 430 说明书。管脚的开关可通过设置 P1DIR 和 P2DIR 控制管脚的输入/输出模式来实现。另外控制电机的管脚也可以设置为 PWM 输出模式，想要添加控制速度功能的同学可以参考 430 说明书的设置方法。通信建议写成 UART 的中断模式，以便修改。

下面给出一些详细注释的代码供参考：

```
void Send(char data){
    while(IFG2 & UCA0TXIFG == 0);
    UCA0TXBUF = data;
} //UCA0TXBUF 寄存器是 UART 的发射寄存器，写在这个寄存器中的内容会被发射出去。

void UART_Init(){
    P1SEL |= BIT1 + BIT2;
    P1SEL2 |= BIT1 + BIT2; //P1SEL 和 P1SEL2 两个寄存器用来设置 UART 发射口和接收口的功能。
    UCA0CTL1 |= UCSWRST;
    UCA0CTL1 |= UCSSEL_2;
    UCA0CTL0 &= (~UC7BIT);
    UCA0BR0 = 0x06;
    UCA0BR1 = 0;
    UCA0MCTL = UCBRS2 + UCBRF3 + UCOS16;
    UCA0CTL1 &= (~UCSWRST);
    UCOIE |= UCA0TXIE + UCA0RXIE; //使能 UART 接收和发射中断
} //UART 的初始化过程。

void TimerA_Init(){
    TA0CCR0 = 3000; //PWM 波的占空比是 TA0CCR1/TA0CCR0 的值
    TA0CCR1 = 1500; //设置 PWM 的占空比为 1500/3000 = 50%
    TA0CCTL1 = OUTMOD_7;
    TA0CTL = TASSEL_2 + MC_1; //以上两句设置 TimerA_0 工作在 PWM 模式

    __bis_SR_register(GIE); //设置 TimerA 的功耗
} //TimerA 的初始化过程
```

```
void IO_Init(){
    P2DIR |= (BIT0 | BIT3); //P2.0 和 P2.3 设置为输出模式
    P1SEL |= BIT6;
    P1SEL2 &= ~BIT6; //以上两句设置管脚 P1.6 输出 PWM 波

    P2DIR |= (BIT1 | BIT2 | BIT4 | BIT5); //设置 P2.1、P2.2、P2.4 和 P2.5 为输出模式
} //管脚的设置
```

```
#pragma vector = USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void){
    unsigned char data = 0;
    data = UCA0RXBUF;
    //可在这里添加对接受信息的处理
} //UART 的接收中断函数
```

```
#pragma vector = USCIAB0TX_VECTOR
__interrupt void USCI0TX_ISR(void){
    IFG2 = 0;
} //UART 的发射中断函数
```

在 main 函数中需要添加以下语句：

```
BCSCTL1 = CALBC1_1MHZ;
DCOCTL = CALDCO_1MHZ; //用来设置时钟来源的代码。
WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
_EINT(); //打开所有中断
```

控制电机时使用如下语句：

```
P2OUT = (BIT1 | BIT4); //P2.1 和 P2.4 输出高电平
//如果想要控制速度，则可以设置 TimerA_1,使其工作在 PWM 模式下，设置方法与
//TimerA_0 相同，然后修改 P2SEL 和 P2SEL2 使要用到的 4 个管脚输出 PWM，通过
//调节 TA1CCR0 和 TA1CCR1 来控制占空比从而控制小车速度。
```