第二届全国高校物联网应用创新大赛编程挑战赛决赛赛题说明文档

为了让各参赛队能更好的理解决赛赛题细节，现对赛题各方面内容和注意事项作出说明。

第一部分：赛题 A

一、关于 TinyOS 环境

1. 底层固化　　功率：DCC2420_DEF_RFPOWER = 1

2. 串口通信　　组号：AM_UART_MSG = 0x89

由于 PC 端往 0 号节点发送数据时需要选定一个组号才可触发 UARTReceive.receive( )，请参赛队使用该组号来获取图片数据包。（参考 SerialToRadio 代码）

3. TinyOS 的版本

TinyOS 为 TinyOS-2.1.1，现提供一个标准的虚拟机环境，有需求的参赛队可下载安装。

百度网盘：http://pan.baidu.com/s/1sjt4UfF　　密码：　jvyr

二、图片编码

对于 1 张 100*100 像素（即 10000 个像素点）的图片，每个像素点含有 3 个字节的 RGB 数据，分别对应红、绿、蓝三原色。大赛组委会根据图片像素点位置，有序的将颜色数据放入数据包中，其数据包格式如下：

数据包序号（2 个字节）+25 个像素点的 RGB 数据（75 个字节）

即：10000 个像素点信息被分为 400 个数据包，每个数据包携带 25 个像素点的 RGB 信息；

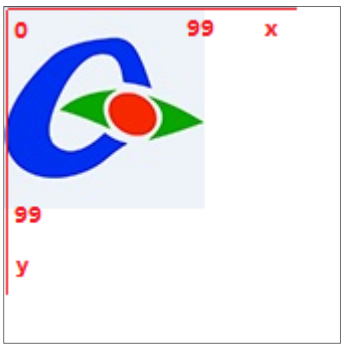


图 1 图片坐标

如上图所示，（0，0）、（1，0）……（23，0）、（24，0）这 25 个像素点数据就被放在序列号为 0 的数据包中，同样的：

（25，0）、（26，0）……（48，0）、（49，0）像素信息放入 1 号包

（50，0）、（51，0）……（73，0）、（74，0）像素信息放入 2 号包

（75，0）、（76，0）……（98，0）、（99，0）像素信息放入 3 号包

（0，1）、（1，1）……（23，1）、（24，1）像素信息放入 4 号包

（75，99）、（76，99）……（98，99）、（99，99）像素信息放入 399 号包。

三、样例代码说明

1. 作用

SendPkt：用于解析图片把图片数据放入数据包发送给 0 号点。

SerialToRadio：可从串口拿到图片数据包，并将包向外广播。

RadioToSerial：可拿到广播的包，并将包发送到串口。

参赛队可以参照 SerialToRadio 和 RadioToSerial 代码分别用于 0 号节点接收图片数据和 49 号节点输出最终结果。

但需切记一点，0 号节点在编写代码时不能存在往串口发送数据的代码块（包括 Printf），因为 0 号节点承担的是从串口接收数据，如果 0 号节点此时还往串口发送数据，那么会直接影响 PC 端的数据发送。

2. 使用

如果参赛队没有节点，请登录大赛云平台进行测试，上面会添加相应功能。

如果参赛队有节点，则可进行如下测试。

STEP1：将 SerialToRadio 烧录到 0 号点，RadioToSerial 烧录到 1 号点。

STEP2：对 1 号点进行监听（/dev/ttyUSB1 需为 1 号点对应端口）

java net.tinyos.tools.Listen –comm serial@/dev/ttyUSB1:telosb

STEP3：SendPkt.java 程序是 PC 端解码图片数据并往 0 号节点发送的 java 程序，参赛队可在安装有 TinyOS 的 Ubuntu 下运行， 查看目录：

```
root@tos211-vpc:/home/tos/Desktop# cd SendPkt/
root@tos211-vpc:/home/tos/Desktop/SendPkt# ls
aa.jpg  SendPkt.java
root@tos211-vpc:/home/tos/Desktop/SendPkt#
```

STEP4：编译 javac SendPkt.java

```
root@tos211-vpc:/home/tos/Desktop/SendPkt# javac SendPkt.java
root@tos211-vpc:/home/tos/Desktop/SendPkt#
```

STEP5：运行 java SendPkt 并输入端口数字，如 0 号点对应 /dev/ttyUSB0,则应发送到/dev/ttyUSB0，输入 0 即可。

```
root@tos211-vpc:/home/tos/Desktop/SendPkt# java SendPkt
请输入你要发送到的端口号:
0
准备往【/dev/ttyUSB0】发送数据包
serial@/dev/ttyUSB0:115200: resynchronising
第serialnum = 0包发送完成
第serialnum = 1包发送完成
```

此时再查看 1 号点的监听窗口，即可看到从 0 号点发送过来的图片信息。

```
00 FF FF 00 00 4D 00 89 01 88 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 89 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8A EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8B EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8C EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8D EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8E EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
00 FF FF 00 00 4D 00 89 01 8F EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F
3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9 EE F3 F9
```

四、最终输出格式

1. 输出格式

由于 49 号点是通过 Send.send 发送到 PC 端，故 Listen 监听到的数据存在着 00 FF FF 00 00 4D 00 89 的头信息。后面的 77 个字节才是包负载，也就是我们关心的内容。所以参赛队最终的输出信息（即在 49 号点 Listen 到信息）必须满足：

头信息（8 字节）+ 数据包负载（77 字节）

核对结果时会将第 9 和 10 个字节当作数据包序号，从而解析出最后的 75 个字节的 RGB 信息。

2. 有序输出

PC 发送给 0 号点的数据包是按照序号(即 0-399)的顺序发送的，在核对正确性时，要求参赛队按照顺序输出包的信息，即输出的包按照 0，1，2，……，398，399 的包序号输出。

a. 如输出为 0，1，3，2，8，5，6，7，9,…则只会算其中 0，1，3，8，9 为正确的输出包；

b. 重复的包只算最开始的那个；

c.不满足长度的包直接忽略；

五、其他注意事项

1. 赛题 A 提供现场测试，参赛队在正式比赛前可提前测试。
2. 赛题 A 中图片，决赛时将更换成其他图片，但像素不变。

第二部分：赛题 B

一、信息收集节点代码

Tagrev 是 Tag 信息收集节点的代码，参赛队须注意以下内容：

通信信道：CFLAGS+=-DCC2420_DEF_CHANNEL=20

无线组别：AM_Radio_MSG= 6

只有信道和无线通信组别按照上述标准节点才可进行通信，另外发送的数据包格式为：

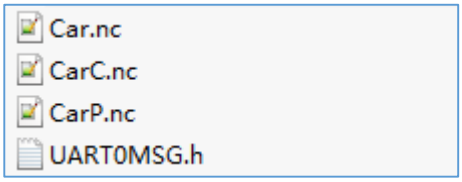
```
//数据包格式
typedef nx_struct TagMsg{
    nx_uint16_t tagid;      //标签id
    nx_uint16_t tagdata;    //标签数值
    nx_uint16_t hint;       //参赛队标志项，用于区分其他队伍
}TagMsg;
```

将按照发送到信息收集节点的数据来计算最终的正确性得分。

二、小车驱动程序和测试程序

驱动程序说明了小车驱动的实现，参赛队只需会调用小车驱动命令即可。

以下 4 个文件为小车驱动文件：



测试程序则是一个使用驱动操控小车的简单测试程序，将 CarTest 程序烧录到车载节点时，小车会启动----前进-----转动-----后退 循环。

其包含的文件为：

BlinkAppC.nc	2,243	0	NC 文件
BlinkC.nc	2,246	0	NC 文件
Car.nc	547	0	NC 文件
CarC.nc	713	0	NC 文件
CarP.nc	14,267	0	NC 文件
Makefile	184	5,734	文件
pr.h	316	0	H 文件
README.txt	678	0	文本文档
UART0MSG.h	219	0	H 文件

三、其他注意事项

1. 所有节点功率限制为 8，通信距离有限，故参赛队需要用辅助节点搭建网络来扩大节点的通信范围。

2. 参赛队需考虑小车脱离网络时，命令无法传达从而导致的小车失控情况。如加入心跳包来实时检测小车是否存在于网络中。

3. 参赛队可在正式比赛前在小车场地进行适应测试。

4. 现场会提供小车、节点等供参赛队使用。