

# 一种手机平台控制的轮式探测机器人的设计与实现

李众力 谢敏

摘要:

本项目建立了一个远程实时控制系统，机器人利用摄像头将其采集到的视频信号压缩后，经过 Wi-Fi 模块通过无线网络传输技术将数据传送到控制端（手机），解压缩以后反应在控制端平台上，控制者通过这些信息发出相应的控制信号，系统再通过 Wi-Fi 模块将控制端的控制信号传输到远程终端（机器人），远程终端运用基于 ARM 的 Linux 嵌入式系统对信号进行解析并进行相应操作响应，实现控制端对与轮式探测机器人的实时操控。

关键词： 机器人 实时视频传输 无线遥控

## 引言

物联网是当下的热门话题之一，它体现了“无处不在的网络、无所不能的业务”的思想，正在改变着人们的生活方式和工作方式。目前，越来越多的工作也已经不是由人直接去进行，而是通过控制机器人去完成的，例如爆破作业、抢险救援等高危险性、高难度工作。本项目是采用移动终端去实时控制轮式探测机器人，由于视频信息采集量大，传输实时性高，并且使用方便，因此具有很大的实用价值和现实意义。

## 1 系统方案

整套系统分为控制端与被控制端，控制端为手机，被控制端为机器人。手机端接收机器人传回的视频信息并显示，同时可将用户的操作发送给机器人。被控制端接收用户传来的控制指令执行相应操作并将实时的视频信息传回控制端。系统示意图如图 1 所示：

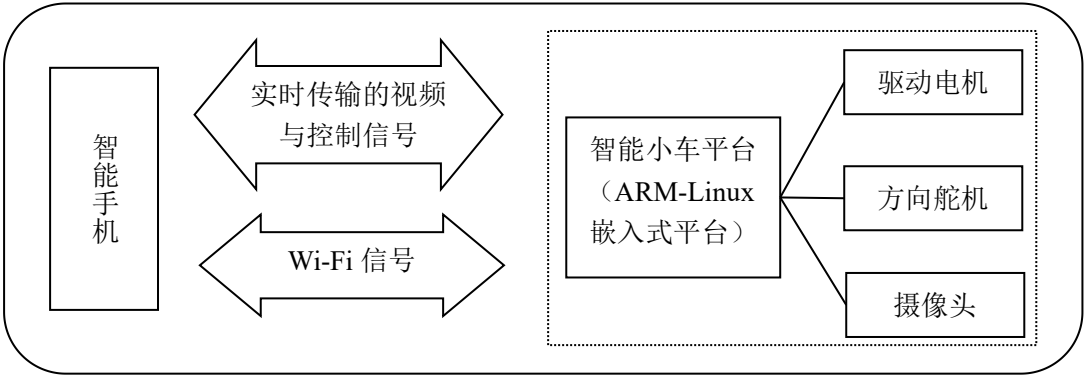


图 1 系统示意图

被控端采用两层架构。上层以 ARM 处理器 S3C6410 为核心，选用广州友善之臂公司的 Tiny6410 开发板作为上层的主板，负责视频采集、压缩，并将压缩后的视频通过 Wi-Fi 方式发送给远端的控制器（手机）。Wi-Fi 无线通信利用无线路由器进行转发。

下层是以 51 单片机为核心的控制板，由 Atmega88 主控芯片、外围电路和电机驱动模块组成。主要负责机器人移动控制，摄像头云台运动控制以及其他传感器的数据采集。51 单片机通过 I/O 口输出 PWM 信号控制直流电机的转速和转向，实现机器人的前进、停止、左转和右转。由于本项目的机器人采用的是大功率有刷电机，需要用很大的电流来驱动，所以我们采用了最大电流高达 60A 的电子调速器作为电机驱动。

机器人主要由硬件系统和软件系统构成。硬件系统包括：ARM 处理器、单片机、电子

调速器、外围电路接口、电机、舵机、摄像头、电池等。其中，由于下层的电机耗电量巨大，如果采用统一供电的方式容易造成电压不稳定，影响上层控制器工作，所以在供电方面我们将上层与下层分开供电。软件包括：嵌入式 Linux 操作系统、外围设备驱动程序、Linux 应用程序以及单片机应用程序等。

## 2 系统硬件设计

硬件系统主要由以下模块构成：图像采集模块、Wi-Fi 通信模块、控制电源模块、Tiny6410、单片机 Atmega88、摄像头云台、电子调速器以及方向舵机、直流电机模块等。具体结构框图如图 2 所示：

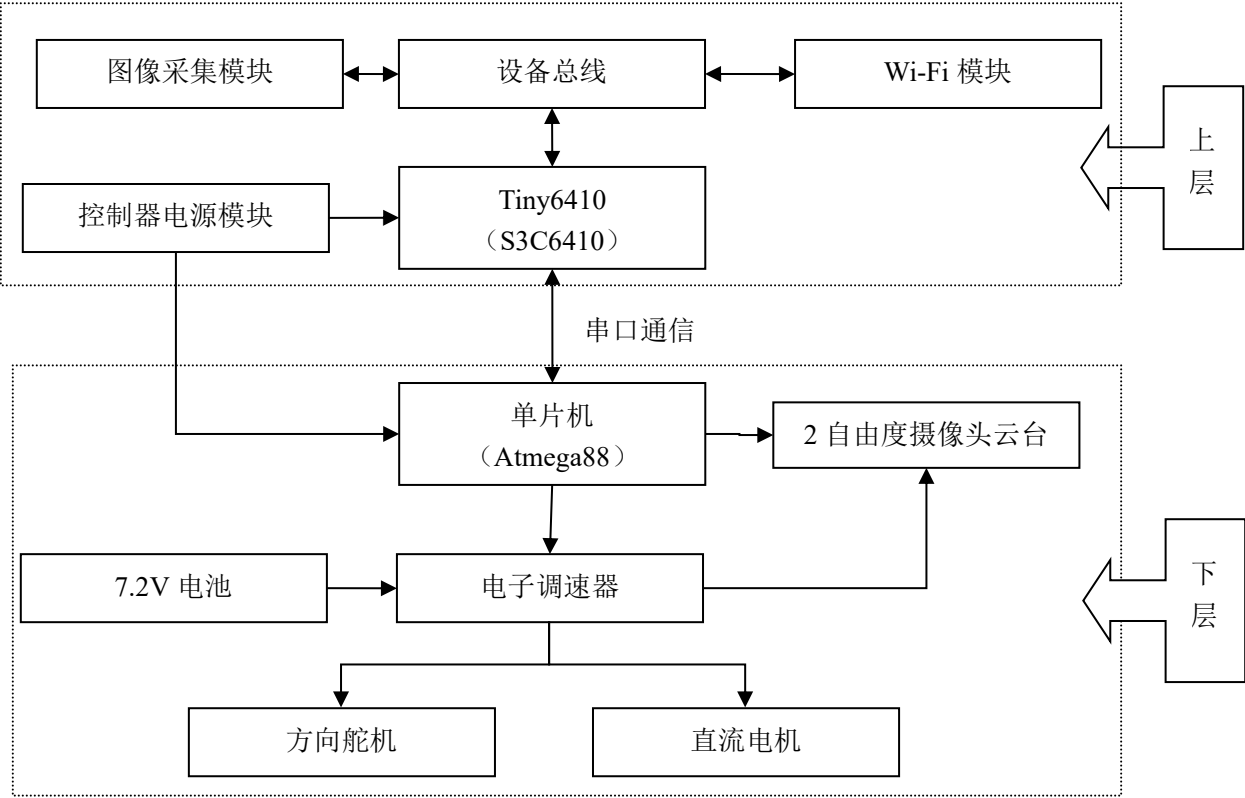


图 2 硬件系统结构框图

由于下层的电机耗电量巨大，如果采用统一供电的方式容易造成电压不稳定，影响上层控制器工作，所以在供电方面我们采用将上层与下层分开供电的方式。控制电源模块内嵌稳压模块，可同时为 Tiny6410 和单片机供电，经过稳压处理后，控制电源模块可给 Tiny6410 提供 3.3V 电压，给单片机提供 5V 电压。为满足下层电机的耗电需求，电子调速器由 7.2V 电池供电后，可自行稳压为方向舵机、直流电机以及 2 自由度摄像头云台供电，输出电压为 5V，最高电流可达 60A，保证直流电机的正常工作。

所有的控制信号都由单片机发送的 PWM 波输出，PWM 输出信号的高低可以控制直流电机的转速、方向舵机的转向以及 2 自由度摄像头云台的转动。当占空比加大时直流转速升高；占空比减小时，转速降低；当占空比为 0 时，电机停止工作。而 2 路 PWM 输出的正负顺序转换则可控制电机的正反转，从而控制小车的前进和后退。同样，不同占空比的 PWM 信号也可控制舵机的不同转向和 2 自由度摄像头云台的转向。

相比于下层系统，上层则稍微简单一些，各模块通过 I/O 接口链接到设备总线上，通过 S3c6410 微处理器统一管理。

## 3 系统软件设计

### 3.1 控制端软件设计

控制端采用的是 Android 手机设备，控制端软件则是基于 Android 平台的程序开发，采用 Java 语言通过 eclipse 编程平台进行程序的设计与编写。Android 应用软件可实现客户端与服务器之间的数据通信。一个 Android 工程一般由多个 Activity、Intent、Content Provider、Service 等程序框架组成<sup>[1]</sup>，其中最基础的是 Activity 活动类，该类相当于软件运行时的一个界面，在界面上可以添加各种控件，如 Button 按键、ImageView 图片等，所有的控件都是在布局文件 layout 中进行设计和布局的。此外，Activity 类还负责监听系统事件，启动其他的 Activity 类。本设计共有两个 Activity，一个实现控制端与被控对象的 IP 地址配对，一旦配对成功则进入另一 Activity，用于用户控制。在多个 Activity 类之间跳转则需要用到 Intent 类，Intent 类有两个重要部分：动作和动作对应的数据。

本系统的控制端软件主要可以完成以下功能：

在确保控制端与被控制端处于同一网络环境中后（即连接相同的无线路由器），搜索被控制端，得到被控制端的 IP 地址与控制端口号。一旦控制端与被控制端配对成功后则由 Intent 类跳转到控制界面。控制界面如图 3 所示：



图 3 控制端控制界面

控制界面会随时监听屏幕按键状态，一旦相应按键按下，将会通过 Wi-Fi 发送相应的指令到被控端，被控端对指令进行解析后作出相应的动作。同时，为了能让用户体验更佳，我们使用了手机的重力传感器，一旦监听到重力传感器数据的变化，就发送相应的方向控制信息到被控端，手机往左倾斜则向左转，往右倾斜则向右转。控制界面还有一个最重要的功能就是实时接收并显示被控制端传来的视频数据，以便实现远程实时监控。

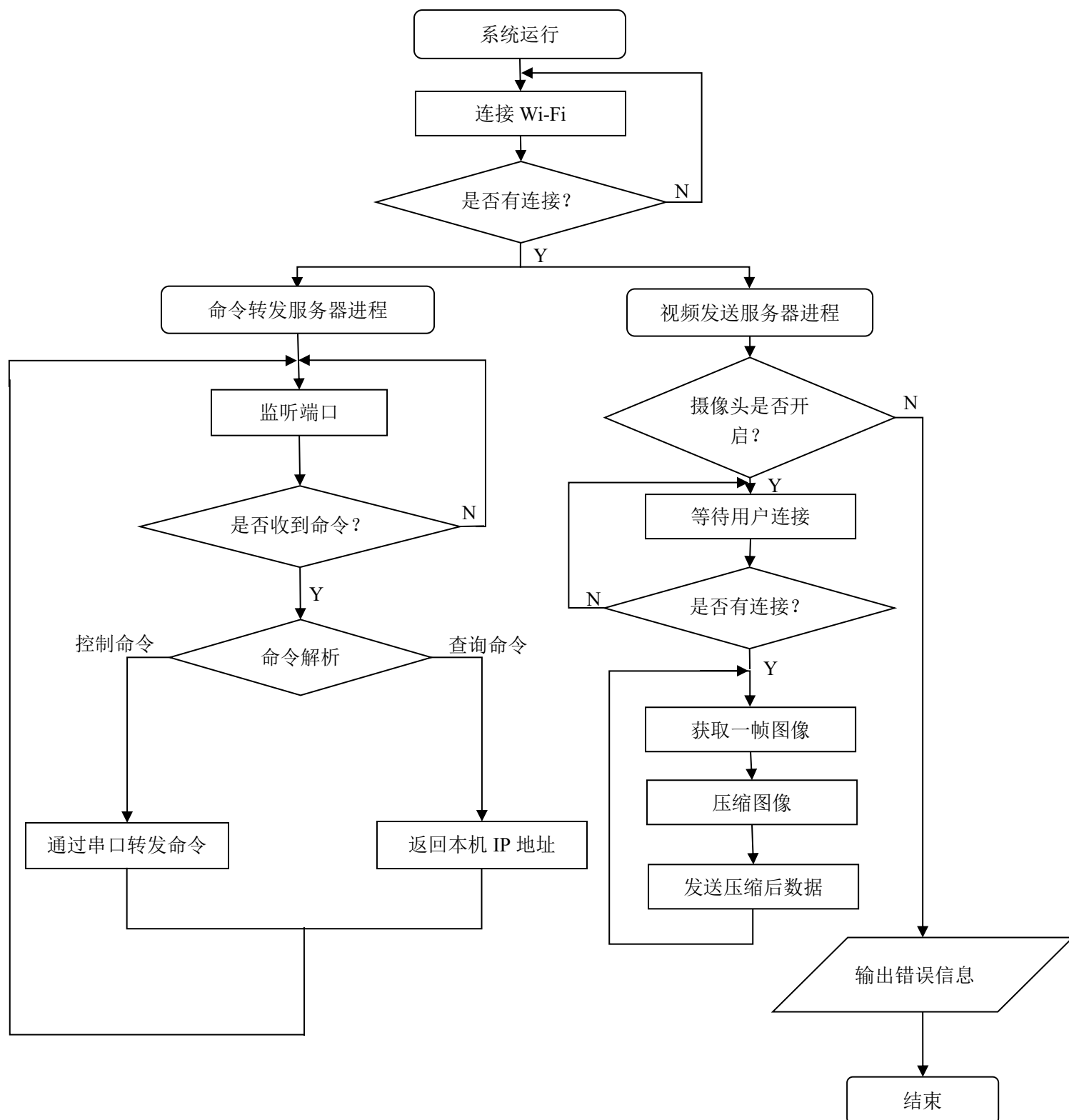
### 3.2 机器人平台软件设计

机器人平台的软件是控制轮式机器人的智能化体现，它控制着机器人的所有运行状态。本设计在 Linux 系统平台上对机器人进行编程，首先对 Linux 内核进行剪裁，加载相关驱动，再将编译好的 Linux 系统镜像烧写到 ARM11 开发板中，完成机器人平台软件运行环境的搭建。<sup>[2]</sup>

对于机器人的控制和视频的传输是本系统的关键技术之一，要完成客户端与服务器端的连接就需要通过 socket 编程实现。Socket 有 2 种通信方式：面向连接的方式和无连接的方式，

本系统采用面向连接的方式，从而提高了系统的可靠性。<sup>[3]</sup>

机器人平台的软件设计主要包含以下几个部分：网络连接脚步程序、命令转发服务器程序、视频转发服务器程序。机器人平台的软件设计成开机自动运行，以便系统能正在工作，其工作流程如图 4 所示：



机器人开机运行后会自动搜索并连接 Wi-Fi 网络，通过网络连接脚本程序可获得 IP 地

址及控制端口号等信息。连接到 Wi-Fi 网络后，机器人可处理多进程，一个是命令转发服务器进程，另一个是视频发送服务器进程。

在命令转发服务器进程中，将实时监听端口，一旦收到命令将对命令进行解析，若为查询命令，返回本机 IP 地址；若为控制命令，则通过串口将命令发送给下层单片机。发送给单片机的指令格式如下：

0xFF	cmd	ID	channel	Data1	Data2	CRC
------	-----	----	---------	-------	-------	-----

其中，0xFF 为起始码，cmd 为控制字，ID 为单片机的 ID 号（在仅使用 1 片驱动控制板时，ID 号出厂是设置为 1，可以通过上位机搜索获得当前板子的 ID 号），ID 号为本机的 ID 时才执行操作，channel 为通道号，CRC 为前五个字节(0xFF 除外)的校验和。

单片机可根据指令执行相应操作，本小车利用舵机控制信号控制电机，cmd 的第一位即 cmd[0]表示控制信号类型。cmd[0]==m 代表移动命令，若 cmd[1]==1，则为直流电机控制命令，其后的数字范围为-100~100，代表小车的移动速度，若 cmd[1]==2，则为舵机控制命令，其后的数字范围为 0~180，代表小车的转动角度。cmd[0]==c 代表控制摄像头云台命令，若 cmd[1]==1，代表云台左右转动，若 cmd[1]==2，代表云台上下转动。控制机器人的指令部分代码如下：

```
void col_car(char cmd[MAXLINE]) {
    printf("%c\n", cmd[0]);
    int tmp;
    //第一位是 m 代表控制电机命令
    if(cmd[0] == 'm' || cmd[0] == 'M') {
        printf("ColCar\n");
        switch (cmd[1]) {
            //1 为控制电机命令，读取后面的数字范围为（-100~100）
            case '1': //控制电机,读取后面的数字范围为（-100~100）

                cmd_buffer[1] = 1;
                tmp = (cmd[3] - 48) * 100 + (cmd[4] - 48) * 10 + (cmd[5] - 48);
                if(tmp > 100) {
                    printf("Speed cmd Error,Spd>100\n");
                    break;
                }
                printf("Spd=%d\n", tmp);
                //将数据转换为角度
                if(cmd[2] == '-') {
                    tmp = 90 - ((tmp * 90) / 100);
                    printf("back+%d\n", tmp);
                    //向串口发送控制指令
                    Send2Com(1, (unsigned char) tmp);
                } else if (cmd[2] == '+') {

                    tmp = 90 + ((tmp * 90) / 100);
                    printf("go+%d\n", tmp);
                    //向串口发送控制指令
                    Send2Com(1, (unsigned char) tmp);
                }
            }
        }
    }
}
```

```

        } else {
            printf("Spd cmd Error\n");
            break;
        }

        break;
        //2 为控制方向命令，后面的数字范围为(0~180)
    case '2':
        tmp = (cmd[2] - 48) * 100 + (cmd[3] - 48) * 10 + (cmd[4] - 48);
        if (tmp < 0 || tmp > 180) {
            printf("Cmd error\n");
            break;
        }
        //向串口发送数据
        Send2Com(2, (unsigned char) tmp);
        break;
    default:
        break;
    }
} else {
    printf("Not for Cars\n");
}
}

```

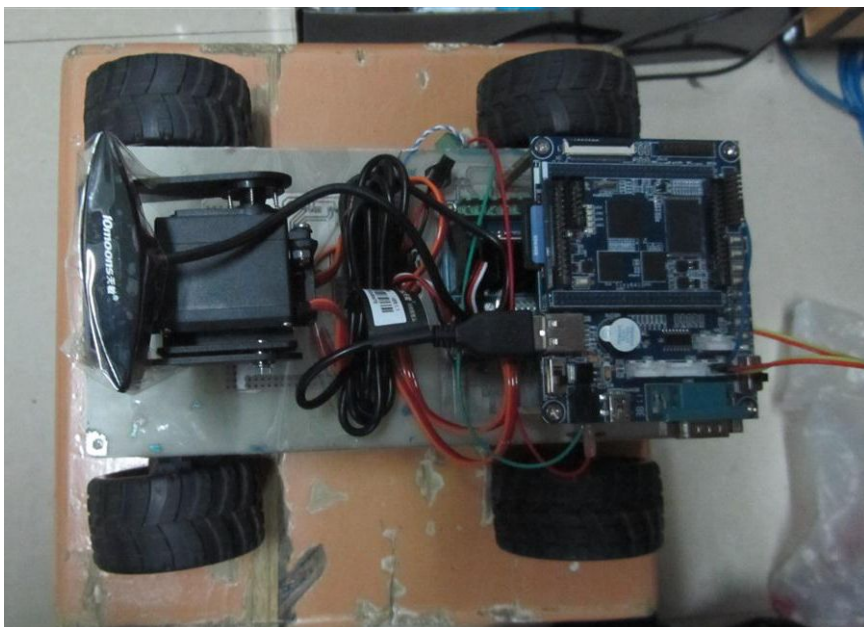


图 4：机器人实体图

## 结语

本设计通过 Android 软件开发平台编程，结合 LinuxARM11 系列开发板的设计与编程，设计并实现了一个利用 Wi-Fi 进行通信的能实时传输视频信号的由智能手机控制的轮式机

器人系统。经过调试运行，智能手机能对该轮式机器人实现良好的实时控制，且视频流传输较为流畅，为以后研究远距离实时控制机器人实现复杂动作的技术打下了良好的基础。

## 参考文献

- [1] 杨丰盛, Android 应用开发解密[M]. 北京: 机械工业出版社. 2010
- [2] Neil Matthew, Richard Stones. Linux 程序设计[M]. 陈健, 宋健健 译. 第 4 版. 北京: 人民邮电出版社, 2010
- [3] Andrew S. Tanenbaum, David J. Wetherall 计算机网络. 严伟, 潘爱民 译. 第 5 版. 北京: 清华大学出版社, 2012