**利用树莓派搭建的远程监控系统**

左牧

我是个业余的电子爱好者，职业是.NET程序员，2012年12月，我第一次了解到树莓派，2013年1月我拥有了第一块树莓派。由树莓派开始接触硬件，随后我又接触了Arduino，并开始彻底爱上软硬件结合的应用开发。现在为大家分享的这个应用是我在树莓派上做的第一个应用演变而来的。

**缘起小车**

刚刚接触树莓派时，我做了一辆遥控小车，当时购买了车底盘、发动机之类的部件，之后又买了个L298N电机驱动板，还买了面包板、杜邦线以及一堆发光二极管。更夸张的，我还买了个玩具遥控玩具车，然后拆开来研究原理。前前后后投入还真不少，在连续多日的努力之下，我的小车终于制作完成了（见图1）。

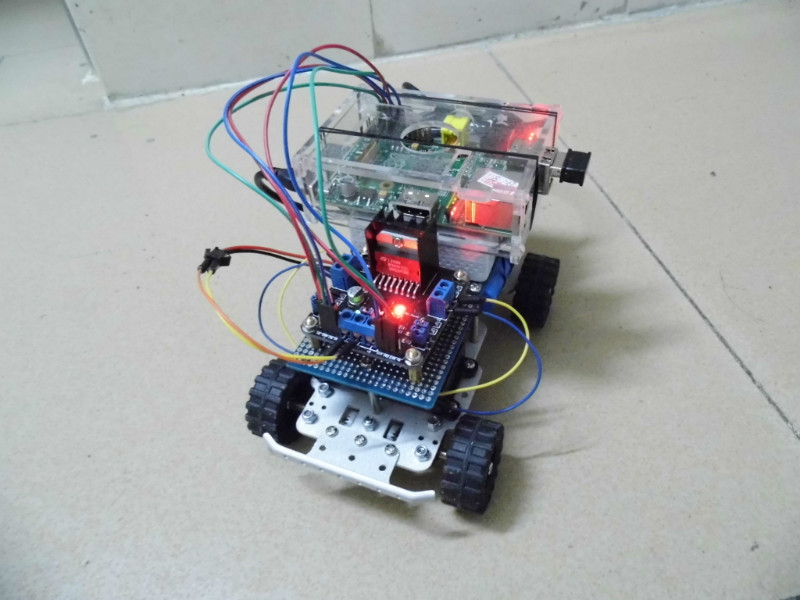


图1 基于树莓派的小车（视频地址：<http://v.youku.com/v_show/id_XNTA0MjA5MTk2.html>）

制作那辆小车时，我采用了python操作GPIO，并利用L298N驱动板去驱动电机，然后通过Wi-Fi联网，再利用VNC远程登陆到树莓派来操作程序控制，实现起来比较简单。当然要完成这个以及接下来的应用，首先需要在树莓派上装好系统，配置好Wi-Fi，安装SSH、VNC等程序，并且需要安装最关键的Python的GPIO库。

**为小车升级**

制作完成的小车操作起来很成功，不过后来我又为它进行了升级。首先为它更换了一个履带式底盘，并增加了摄像头和超声波模块，通过发射超声波来接收遇到物体返回超声波的时间差，再由于声音在空气中传播速度约等于340m/s，就可以算出距离，从而实现了自动避障。

用在这辆小车上的摄像头是一个USB免驱摄像头，属于笔记本配件中比较常见的一种。图像处理是通过python的pygame库实现，但控制程序运行在树莓派本地，所以控制仍然是需要VNC远程。制作完成的小车如图2所示。

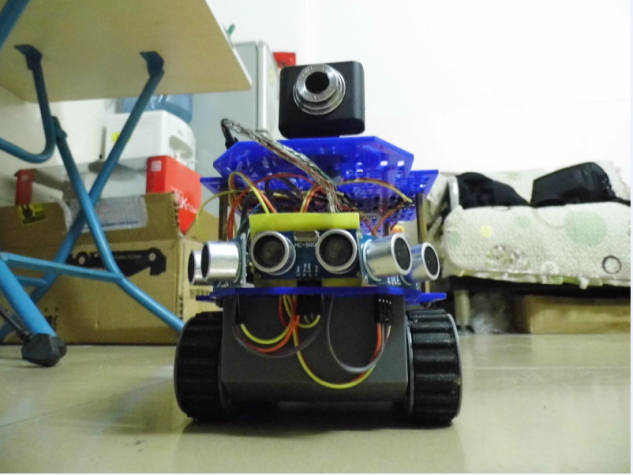


图2 升级之后的小车

后来，由于没解决在家里自动充电的问题，导致续航时间比较短，因此真正用于远程控制时，实用性很低。当时我想，如果这个小车一直连着电源，那么移动和避障就不需要了，只要加装一个云台就可以了。如果能够脱离VNC实现外网做P2P控制就更好了，这样我就能够在异地观察家里的情况（主要是想观察家里的猫）。于是就有了现在要说的这个东西。

**打造远程监控系统**

这一步升级有相对较大的改动，平台不仅仅只采用树莓派，我还采用了Arduino作为下位机，用以驱动云台和灯；树莓派做上位机，也是该控制系统的服务端，用于实现视频的采集和传输以及驱动下位机。另外我还开发了一个.NET winform客户端程序，并通过socket连接，采用TCP协议作控制指令传输，UDP协议做视频画面传输，从而实现了NAT穿透。这样就能在外网环境下控制家里的摄像头了。

该应用涉及3个平台（RPi、Arduino、.NET）、3种开发语言（Python、C、C#），以及图像、多线程和通信技术。

平台之间的连接关系如图3所示。

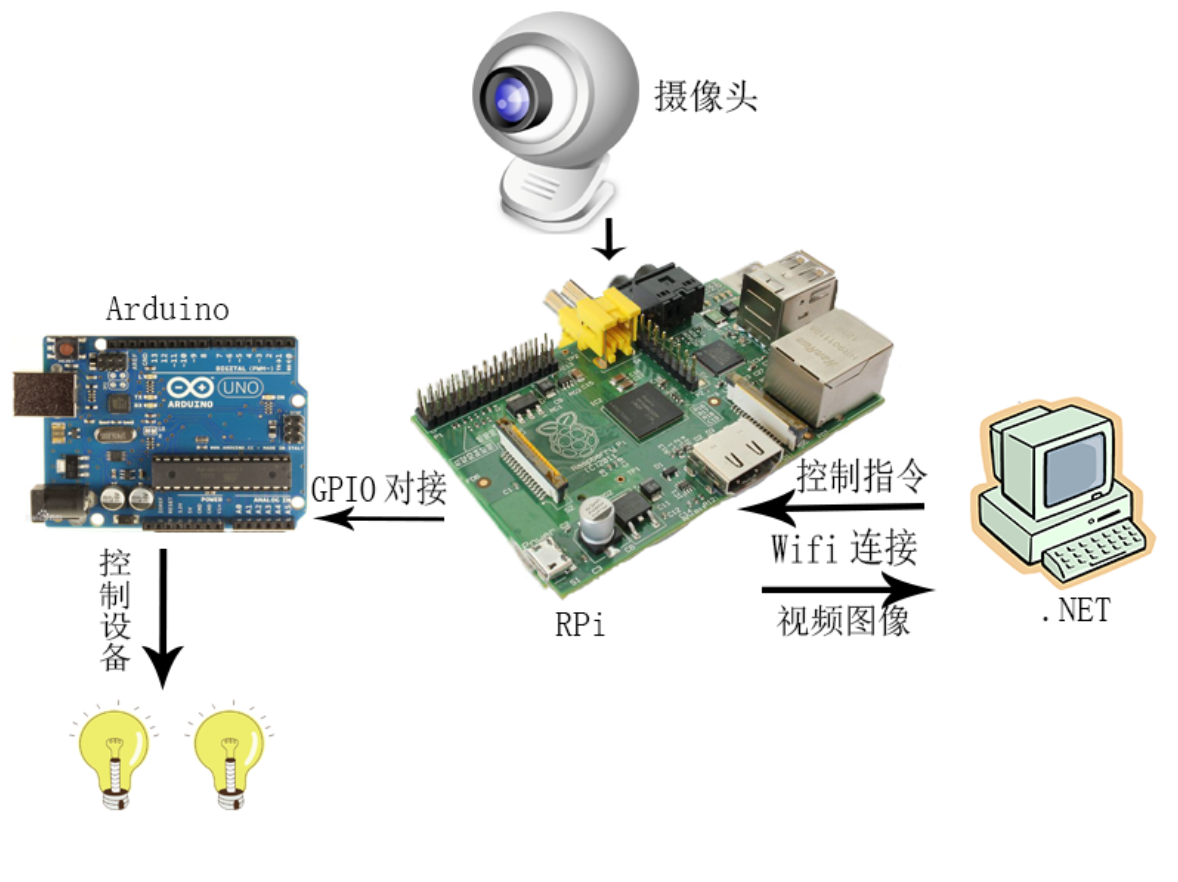


图3 平台之间的连接关系

升级之后的远程监控系统用到的所有硬件如下。

1. 树莓派×1
2. Arduino uno板×1
3. 逻辑电平转换器×1
4. 面包板×1
5. 亚克力支架×1
6. 舵机×2
7. 云台×1
8. USB免驱动摄像头×1
9. USB Wi-Fi网卡×1
10. LED灯珠×2
11. 杜邦线和跳线若干

要实现外网控制，还需要做一些准备工作，首先是一台连上外网的路由器，其次是配置路由器动态DNS。路由器代理配置我采用的是花生壳，配置方法我就以花生壳为例吧，其他类似应用的配置也都很接近，但是路由器不同，对应的路由器网关配置界面也不一样。

关于动态DNS的配置可采取如下步骤。

1. 注册花生壳（Oray）账号；
2. 用浏览器进入你的路由器网关地址（如果之前没设置过的话，默认地址通常是<http://192.168.1.1>，登录名和密码都是admin）；
3. 找到【动态DNS】配置界面，用花生壳账号和密码登录；
4. 找到端口转发配置界面，配置映射端口（例如，我树莓派的服务端程序监听的是8888端口，那么我就把8888端口转发到我树莓派的内网IP上）。

这样一来，树莓派上的端口就绑定到外网了。

**在树莓派上进行的制作**

树莓派中的程序主要包括监听端口，启动视频传输线程，并将视频画面发送到Winform控制端，接收控制端指令发送到下位机。

视频采集和传输

我目前采用的是pygame模块驱动摄像头，通过UDP协议将画面传输到 Winform控制端。在传输的时候，由于mtu最小传输单元的限制，需要将提取到的帧拆包发送。然后在Winform端进行粘包，再将画面显示到窗体。由于单帧图像是进行拆包发送的，网络环境差的话会出现丢包，从而可能导致画面局部缺失，所以目前这块还存在缺陷，待后续优化。之前也试过TCP协议，虽然能避免丢包问题，但是画面延迟比UDP协议高。

树莓派连接Arduino下位机

我采用的是GPIO对接的方式，由于树莓派和Arduino的端口电压不一致，这里需要用到逻辑电平转换模块和杜邦线、跳线若干。连接方式如图4所示。

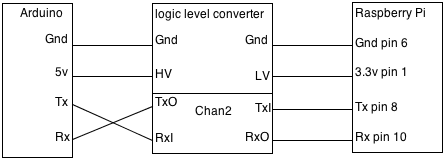


图4 树莓派连接Arduino

**在Arduino上进行的制作**

Arduino是一个开源硬件平台，很多电子爱好者对此是再熟悉不过了，我采用的是Arduino uno板。在这里主要是用来驱动硬件，包括两个舵机，两个LED。通过和树莓派GPIO的对接，接收参数，从而控制舵机的旋转位置和LED的亮灭。LED是为了便于在夜晚监控时提供光线，毕竟我的摄像头仅仅是一个20块钱的免驱摄像头而已。

**制作.NET控制端程序**

在.NET平台实现的是控制端，相当于客户端。

我本身就是.NET程序员，而且程序用的是最简便的Winform程序，对我来说这一环节是完成最快的。程序主要功能包括发送控制指令、接收图像数据、粘包、显示画面。树莓派是部署在家里，而这个控制端需要能够在外网环境使用，比如公司。通过花生壳代理，解析到家里的路由器公网IP地址，.NET连接到服务端后，在NAT上打洞，同时开启图像接收端口的监听。这里开启了两个通道，一个是从控制端到服务端的控制指令通道，另一个是从服务端到控制端的视频传输通道。为了适应不同网络环境，程序设有清晰和流畅两个模式，“清晰”模式的分辨率为320像素×240像素，“流畅”为160像素×120像素。

制作完成的作品如图5所示，利用它拍摄到的效果如图6所示。

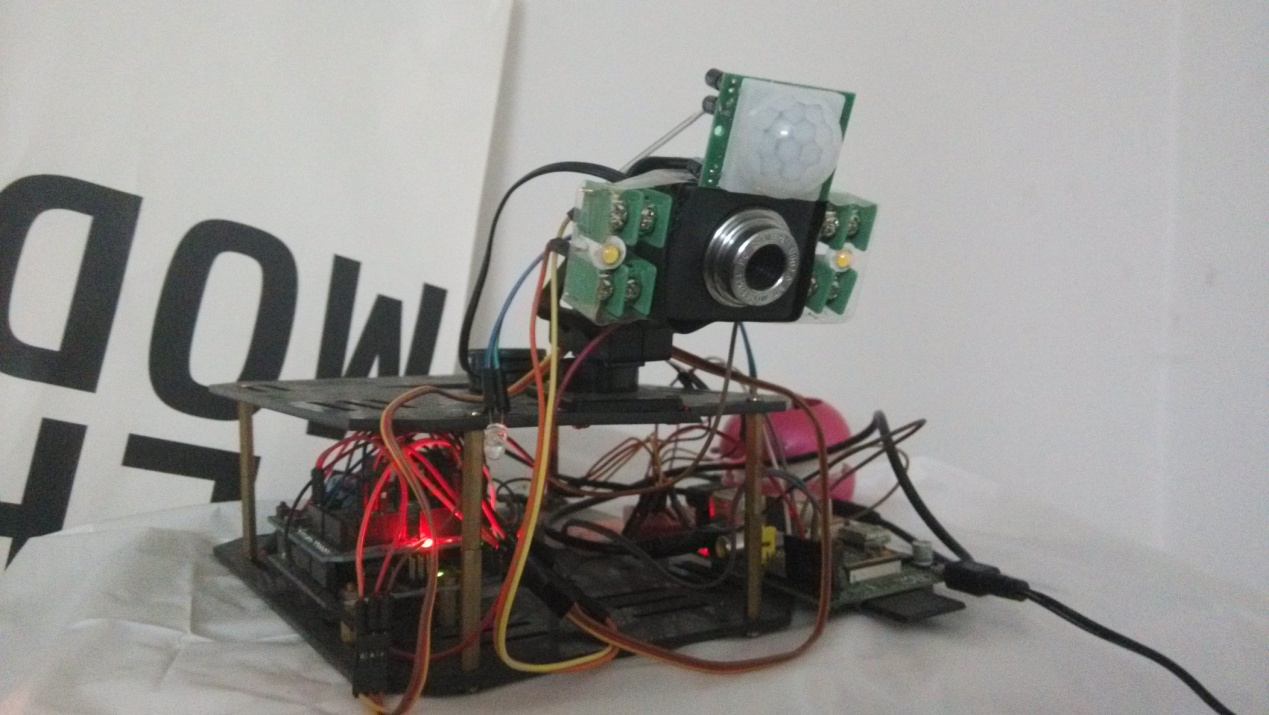


图5 制作完成的监控系统

****

图6 用摄像头拍下的照片（160像素×120像素）

关于此制作的树莓派python程序下载、Arduino 程序下载和.NET程序下载感兴趣的读者可以到无线电网站[www.radio.com.cn](http://www.radio.com.cn)上下载。