

Cubieboard 智能小车 2.0 版

各位伙伴们新年快乐，很少在论坛里发帖，早就答应了师兄要把文档发出来和大家共享，怎知自己的拖延症已经严重到一种境界了。。想想事情总不该拖到明年，所以选择在这个时间发出来。没什么技术含量，和社区里的大牛比起来差得远了，只希望能够帮助那些和我一样的菜鸟，见笑了。

首先得感谢 Windland 师兄的帮助，几乎全程都是在他的指点下完成的，在这里我将完成小车的全过程分享出来，也算是弥补师兄的遗憾吧（他的那张 SD 卡烧毁了，所有代码以及环境都没了，所以就没能把更详细的过程放上来）

必需的配件：

1. 小车底盘 ¥67
2. L298N 驱动模块 ¥60
3. 7.2V 电池组（给 L298N 供电） ¥18
4. 杜邦线 ¥4
5. 移动电源 ¥60
6. 无线网卡 ¥28
7. Arduino ¥60（其实是老师发的）
8. Cubieboard ¥345

额外的配件：

1. 迷你音响（用于扩大声音，注意大小和可充电，USB 口紧张） ¥30
 2. 摄像头（带 MIC 可录音） ¥20
 3. 云台 ¥6.50+10 运费 = ¥26.5
 4. 舵机 ¥8 * 3 = ¥24
-

Cubieboard 智能小车实现的功能：

- 通过网页或是 iPhone App (自己制作完成的) 控制小车
- 小车能够快速前进、快速后退、慢速前进、慢速后退、向左转弯、向右转弯、停止
- 能够播放在网页端输入的文字

- 能够录音，并可以控制录音时长
- 网页端实时摄像头视频输出
- 能够调整摄像头角度，上、下、左、右

Cubieboard 的职责：

- 启动 mjpg-streamer 服务，控制摄像头
- 使用 espeak 播放语音
- 开启若干个端口与 Web 控制台通信
- 数据处理后通过串口与 Arduino 通信

L298N 的职责：

- 控制四台发动机
- 给 Arduino 供电

Arduino 的职责：

- 控制 L289N
- 控制摄像头云台转动

这次我的小车是购买了专用的底板，配合电机驱动板驱动。小车底板和拆卸玩具车相比有更大的空间来安放需要的配件，例如移动电源，音响等，可扩展性更强；缺点是发热量大，价格更高一些，需要自行编写代码驱动电机（不过这不算什么难事啦）

1. cubieboard（1或2都无所谓）装上Linux系统，接上usb无线网卡，配置接入wifi.
2. 通过（双方的串口）针脚，将arduino和cubieboard连接起来，串口通信。
目标：cubieboard向arduino发送一个字符，arduino收到字符后控制一个led亮起来（或是其他，what ever,反正以后就是控制小车电机）
3. 在cubieboard上搭建一个web站，或图形界面的程序，或一个没有界面的socket程序。总之，你需要通过它告诉cubieboard你想向小车发出怎样的指令。目标：你能看到cubieboard给你一个helloworld，你也能发给它一个helloworld，即人机通信了。
4. 网站后台代码（or 其他上述代码）中调用库或系统命令来向cubieboard的串口写数据，数据就会发送到arduino上。目标：你给cubieboard发helloworld时，cubieboard给arduino发helloworld，arduino的led亮起。

5. 购买一个usb摄像头，再linux上安装motion或mjpg-streamer，并启动起来。安装和配置方法可以百度。目标：通过浏览器访问能够看到图像。
6. arduino引脚和 驱动板或遥控车电路相连（根据你用驱动板还是改遥控车而对应），通过驱动板控制电机和遥控车芯片的电路图是可以百度到的，目标：通过代码控制引脚输出的电平从而使电机得到正转、反转、转向的信号，从而能够转起来。

合并以上所有，最后要考虑的就是如何把这些放在一个小车上。
你需要准备螺丝刀，电烙铁，松香，锡

以上是师兄帮助我分解的步骤，我稍微调整了一下顺序，接下来我就按照这个步骤一步步来分享我的过程。

第一步：

这步就不用多说了吧，随便什么系统都没有关系，我用的是 Cubian NAND 上的系统。

第二步：

首先需要指出做这样的小车供电和 USB 口都是问题。Cubie 提供了两个 USB 口，无线网卡占一个，摄像头占一个，因此 Arduino 和 Cubieboard 只能通过串口连接。顺带一提，如果 L298N 驱动板的电源电压足够那么可以将其作为一个稳定的电源给 Arduino 供电，否则就必须使用有两个 USB 口的移动电源，分别给 Cubieboard 和 Arduino 供电。

回归正题，Arduino 和 Cubieboard 的串口连接大家可以参考这篇教程 [\[教程\] 如何使用UART](#)。为了大家方便，我再整理出来介绍一下：

1. apt-get install build-essential make gcc g++ git-core
2. fex2bin 和 bin2fex 工具下载
git clone git://github.com/linux-sunxi/sunxi-tools.git cd sunxi-tools
make
中间可能会发生问题
apt-get install libusb* pkg-config
3. 使用 locate 或者 find 来找到 script.bin 的地址

```
updatedb  
locate script.bin
```

4. 将script.bin 复制到工具目录下
cp xxxxx/script.bin /sunxi-tools

5. 将 bin 反编译为 fex
./bin2fex ./script.bin > ./script.fex

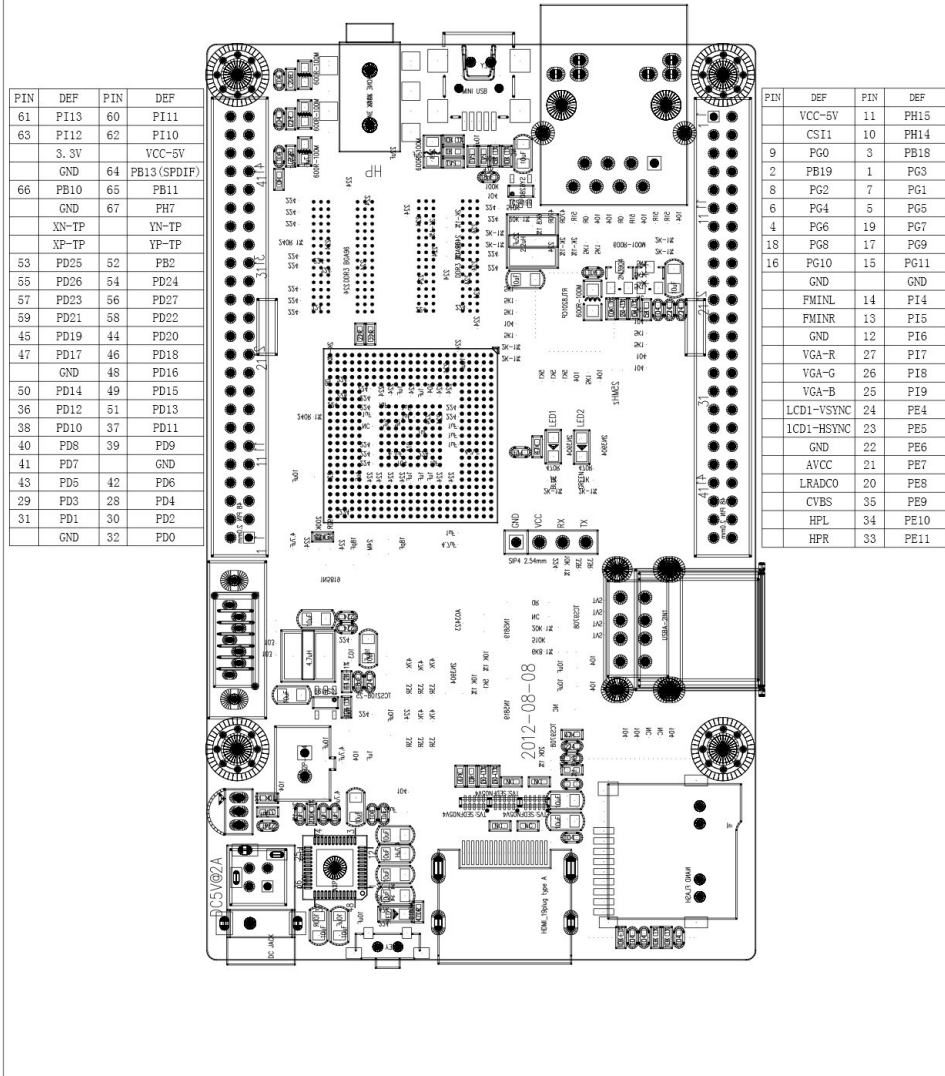
6. 修改script.fex
[uart_para4]
uart_used = 1
uart_port = 4
uart_type = 2
uart_tx = port:PG10<4><1>
uart_rx = port:PG11<4><1>

7. fex 编译回去
./fex2bin ./script.fex > ./script.bin

8. 复制回原位置
cp script.bin xxxxx/
sync
reboot

9. 完成以后将刚刚定义的 tx 针脚和 Arduino 的 rx 针脚连接，rx 和 tx 相连，
GND 与 GND 相连再通过教程中的第九步测试通信
以下是 Cubieboard A20 的 96 针脚图，找到对应的 PG10 和 PG11

Cubian GPIO Pin Defination



[A10/A20 Cubieboard Expansion Ports](#)

10. 在第九步中需要注意的是你自己的机器上未必是 /dev/ttyS3，可能是 /dev/ttyS*，可以自行逐个查看串口信息来判断哪个是对应的设备文件

11. 测试

看串口信息：stty -F /dev/ttyS*

此处的 ttyS* 可以根据串口信息判断

加入echo功能 :stty -F /dev/ttyS* -echo

echo "AAAA" > /dev/ttyS3

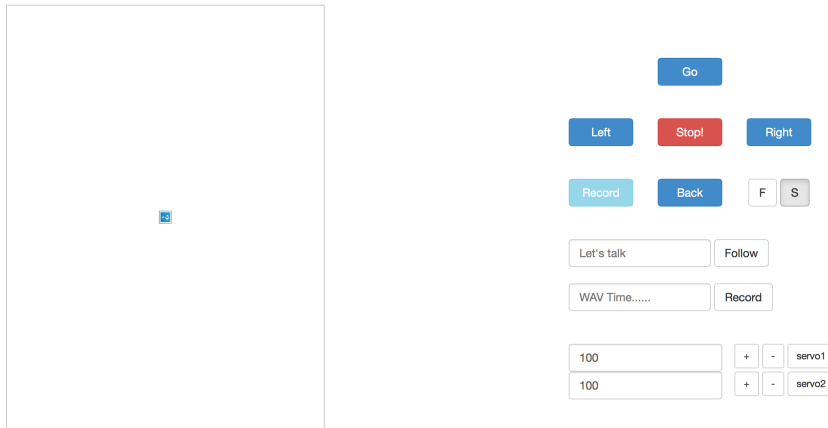


第三步：

由于种种原因，我把 web 站搭在了笔记本上（当然搭在 Cubieboard 上也是一样的，但是由于 Cubie 要被捆绑在小车上，不是那么方便，所以我选择了搭在笔记本电脑上），为了方便 socket 连接，我选择了个人较为熟悉并且编写比较快捷的 Python CGI。

网站如图：

Welcome to the Cubieboard Car Web Controller!!!



通过几个按钮来控制小车前进后退转向，F/S 是控制 Fast 还是 Slow；Talk 是让 Cubie 说出你输入的话，Record 是录音，输入框里面输入要录的时间，以秒为单位；下面两个 Servo 是控制摄像头云台，上面的是上下转，下面的是左右转；左边是摄像头输出。

具体代码请见 Web_code 文件夹

第四步：

我在 Cubie 上用 Python 开了两个 socket 监听，一个对应 Talk, Record，另一个对应其余用于控制小车的指令。总共创建了 3 个 screen，这里两个外加一个 mjpg-streamer 的服务。有需要的话可以再增加一个控制摄像头开关的按钮。

详细介绍一下，Talk 和 Record 就是直接用 Linux 的系统指令来执行接收到的命令。而另一个则是重定向到了 /dev/ttyS1 将信息传给 Arduino 执行。为了区分不同的指令，例如前进后退和舵机旋转，我的解决办法是在传输的数据前加上前缀，例如 '@' 以及 '*' 等等。

具体代码请见 Cubie_code 文件夹

第五步：

我在淘宝上花了 20 块随便淘了一个免驱的 UVC 摄像头，拆掉以后固定在云台上，并用两个舵机控制云台活动。

说起来容易，其实做起来还是遇到了很多麻烦的，可能是因为我买的云台的做工太糟糕了，刚装上去的时候舵机会带着螺丝一起转，但不会带着云台转。于是我被迫多次使用了神器——502胶水来固定螺丝、摄像头、云台等。

安装 mjpg-streamer

通过 SVN 将源码拖到 Cubie 上。选择一个合适的位置，执行：

```
svn co https://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer/ mjpg-streamer
.....
.....
cd mjpg-streamer
make
.....
.....
vi start.sh
```

把 `./mjpg_streamer -i "/input_uvc.so" -o "/output_http.so -w ./www"` 改成 `./mjpg_streamer -i "/input_uvc.so -y" -o "/output_http.so -w ./www"` 目的是兼容目前流行格式的摄像头。

再然后执行 `./start.sh` 完了以后打开 `http://127.0.0.1:8080` 就可以看到控制台了。要向远程访问只要把 127.0.0.1 换成 Cubie 的 IP 就可以了。

将视频输出到自己搭建的 Web 控制台

首先在你的网页代码的 `<head></head>` 标签中插入

```
<script type="text/javascript">
  /* Copyright (C) 2007 Richard Atterer, richard@atterer.net
  This program is free software; you can redistribute it
  and/or modify it
```

```

    under the terms of the GNU General Public License, ver
sion 2. See the file
    COPYING for details. */

    var imageNr = 0; // Serial number of current image
    var finished = new Array(); // References to img objects
which have finished downloading
    var paused = false;

    function createImageLayer() {
        var img = new Image();
        img.style.position = "absolute";
        img.style.zIndex = -1;
        img.onload = imageOnload;
        img.onclick = imageOnClick;
        img.src = "http:// (你的 IP 地址) :8080/?action=snapshot&
n=" + (++imageNr);
        img.height = "420"; // 大小可以自行调整
        img.width = "560";
        var webcam = document.getElementById("webcam");
        webcam.insertBefore(img, webcam.firstChild);
    }

    // Two layers are always present (except at the very begi
nning), to avoid flicker
    function imageOnload() {
        this.style.zIndex = imageNr; // Image finished, bring t
o front!
        while (1 < finished.length) {
            var del = finished.shift(); // Delete old image(s) fr
om document
            del.parentNode.removeChild(del);
        }
        finished.push(this);
        if (!paused) createImageLayer();
    }

    function imageOnClick() { // Clicking on the image will p
ause the stream
        paused = !paused;
        if (!paused) createImageLayer();
    }
</script>

```


在 `<body></body>` 标签中合适的位置插入：`<div id="webcam"></div>`，并且可以使用 `style="margin-left:20px;"` 来调整视频的位置。这样就成功将视频输出到网页中了！

此外由于摄像头长宽的限制，我必须将画面输出翻转 90°，因此我需要在网站的 `<head></head>` 标签中插入如下代码：

```
<style>
img
{
    /* Rotate div */
    transform:rotate(90deg);
    -ms-transform:rotate(90deg); /* Internet Explorer */
    -moz-transform:rotate(90deg); /* Firefox */
    -webkit-transform:rotate(90deg); /* Safari 和 Chrome */
    -o-transform:rotate(90deg); /* Opera */
}
</style>
```

如此即可实现视频的倒转

MIC 录音

至于开启 MIC 录音功能，首先需要 alsa 包，可以使用 apt-get 安装。alsa 是一套非常著名的音频驱动，可以使用 alsamixer 来调整音量。

arecord 指令用于录音
arecord -h 查看帮助信息

注意这几个选项

```
arecord -D, arecord --device
-D - 设备名称选择PCM的名字
arecord -t, arecord --file-type
类型的文件类型（VOC，WAV，原料或AU）
arecord -f, arecord --format
-f, - format =格式样本格式（不区分大小写）
arecord -d, arecord --duration
-d, - 时间=# # 秒后中断
```

当中属 -D 最为复杂，分为以下几步进行操作：

首先使用 `arecord -l` 查看自己设备对应的标号

```
**** List of CAPTURE Hardware Devices ****
card 0: sunxicodec [sunxi-CODEC], device 0: M1 PCM [sunxi PCM
]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: CAMERA [USB2.0 PC CAMERA], device 0: USB Audio [USB A
udio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

显然我需要使用的是 CAMERA [USB2.0 PC CAMERA] 来进行录音，注意首先它是 **card 2**，所以 hw 的第一个数字为 2，然后它是 card 2 中的第一个设备，所以第二个数字为 0，因此得到了 **hw:2,0**

注意：如果提示的是

```
arecord: main:682: audio open error: No such file or directory
```

那说明设备不对，请按照上述步骤检查尝试

但是这样以后还会提示错误：

```
arecord: set_params:1087: Channels count non available
```

经过 Google 得知需要将 **hw** 修改为 **plughw**，修改之后即可实现录音

附上最终代码：

```
arecord -d 10 -f cd -t wav -D plughw:2,0 foobar.wav
```

经测试该话筒音质不错，录了 20s 就产生了 3.4M 的文件

此外 aplay 用于播放，直接 `aplay foobar.wav` 即可

特别说明：以上仅仅是自己成功的经验之谈，并未深入研究，如有错误欢迎指出，一定虚心接受。

第六步：

我是使用 L298N 驱动模块的，在购买的时候卖家提供了文档支持，即便没有自己动手百度一下就能找到。

其实是蛮容易的，只要输入高低电平就可以控制小车运动了，再控制 ENA 针脚就

可以调整速度。

不过我在使用 L298N 的过程当中有一个插曲，我在测试过程中一不留神 L298N 下边的针脚通过杜邦线短接了，瞬间温度急剧升高，我闻到了焦味才反应过来。这次错误导致了一个元件被烧坏，因此我的其中一个轮子不能调整速度，而只能开和关。也就是说，它只能够在 PWM 为 255 的时候全速运行，而不能控制速度，因此我的快慢速实现只能被迫改为前轮驱动后轮，即关闭后轮电机，而不是我原先设计的降低所有电机的速度。

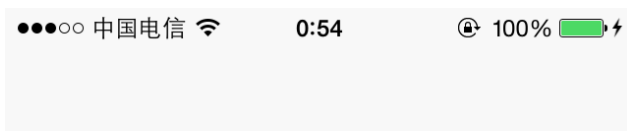
具体代码请见 Arduino_code 文件夹

最后，如何将它们全部放到一个小车上？

其实这也不是件容易的事儿，首先按照图纸安装好小车底盘，接着我遇到了第一个，需要将导线与发动机焊接起来，这对于从来没有尝试过焊接的我来说是有一定挑战的。不过幸运的是我找到了一家拥有电烙铁和锡的店，经过小心翼翼的几次尝试我发现这其实并没有我想象的那么困难，于是也是较为顺利地完成了。然后就是要解决如何安置茫茫多导线的问题，为了节省空间，我把 L298N 和 Arduino 放在了小车的第二层，并且将杜邦线折叠一次以后用线绑在一起，这样挤一挤正好能将这一层的空间完全利用。

虽然我买的小车底板上面有许多的孔，但是很遗憾我一个都没能用起来，因为这根本没有适合固定 Arduino 和 Cubieboard 的孔，甚至连它自家的 L298N 我也没能固定上去。最终我想到了使用具有一定厚度的泡沫胶来将所有的模块与底盘粘在一起，效果非常不错，基本不会有任何移动，并且在必要的时候也能够拆卸下来。

再贴一张自己写的 App 截图，纯属学习和无聊，个人的第一个 App，用了讯飞语音识别。UI 丑是丑了点，不过能用就行 = =



播放语音 识别语音



具体代码请见 iPhone_code 文件夹



总结:

自从买了 Cubieboard 就觉得把这样一块开发板仅仅用于学习 Linux 和搭建服务器真是大材小用了，学习了 Arduino 以后我终于接触到了这样一个对我来说全新的领域。虽然说把这样一台高配置（相对于其它嵌入式设备来说）的开发板只是做了这样一辆小车还是很 low 的，但经过努力能做出这样一个不算太差的作品我对自己的表现还算满意的。其实最后的作品不是最重要的，而是在这过程中我又学会了很多新的知识，obj-c 以及 iOS 开发，alsa 音频驱动包的使用，Cubieboard 针脚控制，内核修改等等，以及巩固了 Python、Web 开发的一些知识。在整个过程中不断地遇到困难，解决困难，知识搜索能力和独立思考能力都

得到了很大的提升。

另外在这个过程中真的受到了师兄很大的帮助，并且在社区上也找到了许多宝贵的资料，这也让我强烈地感受到了分享的力量，因此我也会尽我一份力去帮助那些和我一样爱折腾的人。



最后附小破车露骨照：

