

任务2：找到你的pcd

1.查询 PCDuino 网卡的物理 MAC 地址；

minicom 串口连接树莓派，网络共享给树莓派

指令：ifconfig 可看到 HWaddr

```
pi@raspberrypi:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr b8:27:eb:7a:11:51
          inet addr:192.168.2.7  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5078 (4.9 KiB)  TX bytes:4942 (4.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:828 (828.0 B)  TX bytes:828 (828.0 B)
```

2. 写一个 PC 程序，Unix 或 windows，能够能通过对局域网内所有 IP 地址的 ping，找到 arp 表里的 MAC 地址和 IP 地址 对应关系，根据 PCDuino 的 MAC 段找到局域网内的 RPi 的 IP 地址 。

```
(192.168.2.7)
macs-MacBook-Pro:嵌入式 mac$ python ip.py
raspberrypi IP address is(192.168.2.7)
macs-MacBook-Pro:嵌入式 mac$ █
```

通过 arp -a 得到 arp 表，然后提取出对应的 mac 和 ip，然后匹配树莓派的 mac 地址，就可得到对应的 ip 地址

代码如下：

```
import os
arp = os.popen('arp -a').readlines()

iplist = []

maclist = []

for each in arp:
    each = each.split(' ')
    for item in each:
        if item.find(':') >= 0:
            maclist.append(item)
        elif item.find('.') >= 0:
            iplist.append(item)

pi_mac = 'b8:27:eb:7a:11:51'

i = 0
for mac in maclist:
    if mac == pi_mac:
        break
    i = i + 1

print 'raspberrypi IP address is' + iplist[i]
```