

树莓派开发

11 用树莓派做视频监控

1、给树莓派装上摄像头

□ 第一步：将摄像头连接到树莓派上

现在的摄像头都是支持USB口的，因此，将摄像头的USB头，插到树莓派上，就完成了摄像头与树莓派的物理连接

1、给树莓派装上摄像头

- ❑ 第2步，让树莓派识别摄像头
- ❑ 树莓派上的操作系统是Raspbian “wheezy”——一个基于Linux的Debian完整操作系统，早前的版本可能不能自动识别摄像头，但我们现在用的版本（2014-06-20-wheezy-raspbian.img）是可以自动识别的，所以不要做任何事情。
- ❑ 将USB摄像头接到Pi上后，可能需要安装摄像头驱动，但我们现在使用的版本(2014-06-20-wheezy-raspbian.img)已经包含了V4L2驱动，所以不要再安装驱动了。
- ❑ 验证一下：

pi@raspberrypi ~ \$ ls -l /dev/video0

- ❑ 结果如图：

```
root@raspberrypi: #  
root@raspberrypi:~# ls -l /dev/video0  
crw-rw---T+ 1 root video 81, 0 1月 1 1970 /dev/video0  
root@raspberrypi:~#
```

1、给树莓派装上摄像头

- ❑ 第三步：在**raspi-config**中使能摄像头
sudo raspi-config
- ❑ 移动到第5项——“Enable Camera”选项，按下回车键，进入camera使能设置页，选择Enable，回车。
- ❑ 选择树莓派“重启”
- ❑ 一般系统Camera都是使能的，但为了保险起见，此步做以下。

2、视频监控软件

□ 所谓“视频监控”有不同的概念：

- 本地、远程（内网、外网）
- 视频、图像

□ 一般理解的视频监控是：

- 1、通过摄像头获取图像
- 2、通过网络将图像上传到服务器上（解决固定IP，不是必须的）
- 3、用手机通过APP从服务器获取图像

□ 树莓派的作用是：采集图像和将图像上传到网络。

□ 有很多种视频监控软件，其实差别还是比较大的：

□ Fsw webcam

□ Motion

□ mjpg-streamer

2、安装监控软件

- ❑ Motion是比较有名的监控软件
- ❑ Motion的操作过程是：
 - ❑ 首先初始化设备和创建管理线程，探测并打开视频设备（如果是USB 摄像头，则设备名称是/dev/video0。
 - ❑ 如果打开设备成功，则需要读取并设置视频设备的一些参数信息，如摄像头类型，帧大小，和图像数据读取方法等。
 - ❑ 之后是创建图像采集线程Thread0，打开socket等待客户端建立连接，连接成功后创建线程service向客户端传输图像。
 - ❑ 当图像采集线程退出时，程序结束。
- ❑ Motion还具有侦探动态图像、传输图片、甚至支持MySQL操作等功能。

3、修改配置项

❑ 安装软件motion

```
sudo apt-get install motion
```

❑ 配置motion，具体方法为：

❑ 修改/etc/motion/motion.conf文件：

- ❑ Daemon = off 改为on（始终在后台运行，用top查进程杀掉）

- ❑ webcam_localhost = on改为off（不限制只在局域网内访问motion的监控结果）。

- ❑ 默认的监控端口为8081，改不改都可。

❑ 为了方便，设置开机自动运行，方法为更改/etc/default/motion文件，把 start_motion_daemon=no改为yes。

❑ 重启树莓派，重启后就可以看到效果了，在浏览器输入树莓派IP：端口就可以了。

4、关键配置项说明

- /etc/motion/motion.conf文件的主要内容是
- **daemon off**
- **setup_mode off**
- videodevice /dev/video0
- v4l2_palette 2
- width 640
- height 480
- framerate 2
- threshold 3500
- threshold_tune off
- # Noise threshold for the motion detection (default: 32)
- noise_level 64
- gap 10
- quality 50

4、关键配置项说明

- # Use ffmpeg to encode mpeg movies in realtime (default: off)
- ffmpeg_cap_new on
- ffmpeg_cap_motion on
- ffmpeg_bps 500000
- ffmpeg_video_codec msmpeg4
- snapshot_interval 1
- locate on
- target_dir /mnt/flash_8G/motion/snapshots
- snapshot_filename %v-%Y%m%d%H%M%S-snapshot
- jpeg_filename %v-%Y%m%d%H%M%S-%q
- movie_filename %v-%Y%m%d%H%M%S
- webcam_port 8081
- webcam_quality 60
- webcam_motion off
- webcam_maxrate 8
- webcam_localhost off

4、关键配置项说明

- # TCP/IP port for the http server to listen on (default: 0 = disabled)
- control_port 8080
- control_localhost off
- control_html_output on
- on_event_start /home/pi/motion/on_motion_detected.sh
- on_event_end /home/pi/motion/on_motion_end.sh

4、关键配置项说明

- **daemon off**: 关掉daemon模式。否则运行motion后，会在后台直接运行，需要用top命令查看出motion的进程号，才能kill掉motion进程。
- **locate on**: 设置当探测到图像中有运动时，把运动区域用矩形框起来。
- **videodevice /dev/video0**: 设置加载USB摄像头的设备，一般为video0，当使用network webcam，需要设置netcam_url，此时，videodevice选项自动失效。
- **v4l2_palette 2**: 设置图像的编码格式为MJPEG，可选值为0-8，分别对应的不同的图像编码格式。
- **width和height**: 设置图像、视频的长度和宽度，值越大，图像越清晰，但所需的网络流量越大，为降低占用的宽带流量，这里设置为640和480。
- **threshold_tune off**: 设置是否使用motion detection阈值自动调节。当设置为on时，下一个设置threshold 4500自动失效。设置off时，可以由threshold指定当探测到多少像素变化时，判断为图像中有运动。
- **ffmpeg_cap_new on**: 指在侦测到运动时，用视频纪录下来。
- **ffmpeg_video_codec msmpeg4**: 设定视频的编码器。
- **/target_dir /mnt/tmp_motion**: 当探测到运动时，图片和视频的保存路径，这里设为保存在U盘下。默认为/var/lib/motion/snapshots。

4、关键配置项说明

- **snapshot_interval 1**: 设定自动采集图片的周期，当有运动被检测到时，采集频率会自动变高。
- **webcam_port 8081**: 设置视频输出流的访问端口，默认8081端口。当需要外网访问motion监控的过程时，需要设置webcam_localhost为off，即不限制只在局域网内访问motion的监控结果。
- **control_port 8080**: 设置motion的网络控制端口，通过该端口可查看和修改motion.conf里的配置，如要外网访问，需设置control_localhost为关闭。
- **on_event_start /home/pi/motion/on_motion_detected.sh**: 设置当探测到运动时，执行指定目录里的文件，这里设定为执行/home/pi/motion/目录下的on_motion_detected.sh脚本命令。
- **on_event_end /home/pi/motion/on_motion_end.sh**: 当on_event_start事件开始后，即检测到运动之后，若连续10秒不能再检测到运动时，执行指定目录下的文件，这里设置为执行/home/pi/moiton/目录下的on_motion_end.sh脚本命令。10秒参数由gap 10选项设定。

5、事件脚本

- ❑ 当开启motion视频监控进程时，motion会以一定帧速拍摄JPEG的监控图像
- ❑ 当侦测到画面有指定大小的像素变化时，即监控范围内有物体运动时，motion将触发on_event_start事件，执行用户设置的on_event_start事件的脚本命令。
- ❑ 当检测到运动之后，如果有连续10秒钟像素的变化的值小于阈值，即不能再检测到运动时，触发on_event_end事件，执行用户设置的on_event_end事件的脚本命令。

5、事件脚本

- ❑ on_motion_detected.sh脚本作用是，记录下探测到运动时的时间，即拍摄的视频文件的文件名的一部分。
- ❑ 把这个时间值存到/home/pi/motion/tmp/videotime文件中。该脚本内容如下：

```
#!/bin/bash
```

```
echo "on_motion_detected start!"
```

```
DATE=$(date +"%Y%m%d%H%M%S")
```

```
ALARM_TIME="/home/pi/motion/tmp/videotime"
```

```
echo "$DATE"> $ALARM_TIME
```

5、事件脚本

- 而on_motion_end.sh脚本的作用是用于查找指定的视频文件，并将视频文件以邮件形式发送到用户指定的邮箱，当邮件发送失败时，将视频保存到大容量的存储盘中，减少对SD卡的写操作。

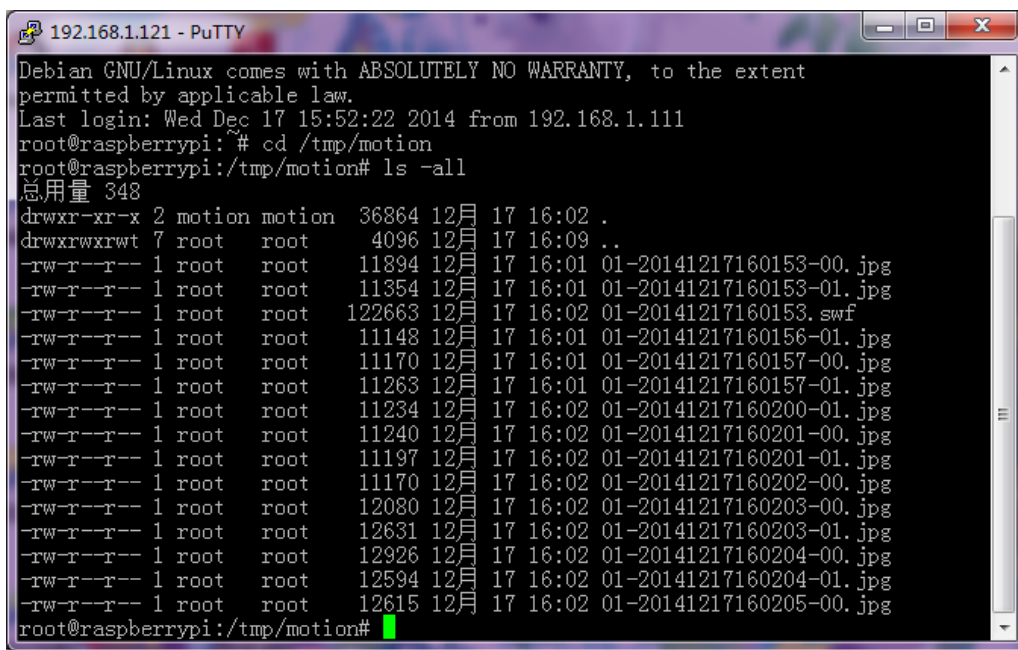
5、事件脚本

- ❑ 当motion检测到的运动结束后，将拍摄的运动视频文件（.avi文件）发送到指定邮箱里。
- ❑ avi视频的文件名为一个序号+检测到运动的时间+“.avi”，而检测到运动的时间，根据on_motion_detected脚本，保存在/home/pi/motion/tmp/videotime文件里，理论上只要从文件里读出时间，然后补全文件名，该序号由*号替代，便能发出邮件。
- ❑ 但是，由于程序运动效率原因，有时会出现记录的时间同开始录制avi视频的时间差1秒的情况，虽然只有一秒，但足以导致脚本找不到avi文件，无法正确发出监控视频。
- ❑ 因设置了gap为10，即10秒内最多只有一个视频。所以，可以去查找，以videotime中所记录的时间及其上一秒的时间命名的视频文件。

6、运行结果1

- 根据设置，在/tmp/motion下，保存了大量的视频和图像文件（依摄像头下物体运动的多少和时间的长短）。
- 其中，视频文件约为10秒长度。

- 系统关机之后，这些文件将会被删除（因为是/tmp目录），因此，motion可以在删除之前，把这些文件发送到你的信箱中。



```
192.168.1.121 - PuTTY
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 17 15:52:22 2014 from 192.168.1.111
root@raspberrypi:~# cd /tmp/motion
root@raspberrypi:/tmp/motion# ls -all
总用量 348
drwxr-xr-x 2 motion motion 36864 12月 17 16:02 .
drwxrwxrwt 7 root root 4096 12月 17 16:09 ..
-rw-r--r-- 1 root root 11894 12月 17 16:01 01-20141217160153-00.jpg
-rw-r--r-- 1 root root 11354 12月 17 16:01 01-20141217160153-01.jpg
-rw-r--r-- 1 root root 122663 12月 17 16:02 01-20141217160153.swf
-rw-r--r-- 1 root root 11148 12月 17 16:01 01-20141217160156-01.jpg
-rw-r--r-- 1 root root 11170 12月 17 16:01 01-20141217160157-00.jpg
-rw-r--r-- 1 root root 11263 12月 17 16:01 01-20141217160157-01.jpg
-rw-r--r-- 1 root root 11234 12月 17 16:02 01-20141217160200-01.jpg
-rw-r--r-- 1 root root 11240 12月 17 16:02 01-20141217160201-00.jpg
-rw-r--r-- 1 root root 11197 12月 17 16:02 01-20141217160201-01.jpg
-rw-r--r-- 1 root root 11170 12月 17 16:02 01-20141217160202-00.jpg
-rw-r--r-- 1 root root 12080 12月 17 16:02 01-20141217160203-00.jpg
-rw-r--r-- 1 root root 12631 12月 17 16:02 01-20141217160203-01.jpg
-rw-r--r-- 1 root root 12926 12月 17 16:02 01-20141217160204-00.jpg
-rw-r--r-- 1 root root 12594 12月 17 16:02 01-20141217160204-01.jpg
-rw-r--r-- 1 root root 12615 12月 17 16:02 01-20141217160205-00.jpg
root@raspberrypi:/tmp/motion#
```

- 可以试试在conf中，把保存地址改在别的目录下。

6、运行结果1

□ 看一下视频swf文件及图形jpg文件，图中方框，就是运动的部分。



6、运行结果2

- ❑ 在PC浏览器上输入：树莓派IP：8081（不行？）
- ❑ 在电脑端，无论是Windows，Linux还是OSX，或者安卓机器，需要安装VLC（一种媒体播放器软件），现在就可以打开VLC，然后打开媒体-》网络串流-》输入**http://PI的IP地址:8081**查看实时不卡的网络监控了。



6、运行结果2

❑ 在树莓派上同样可以看到此视频流

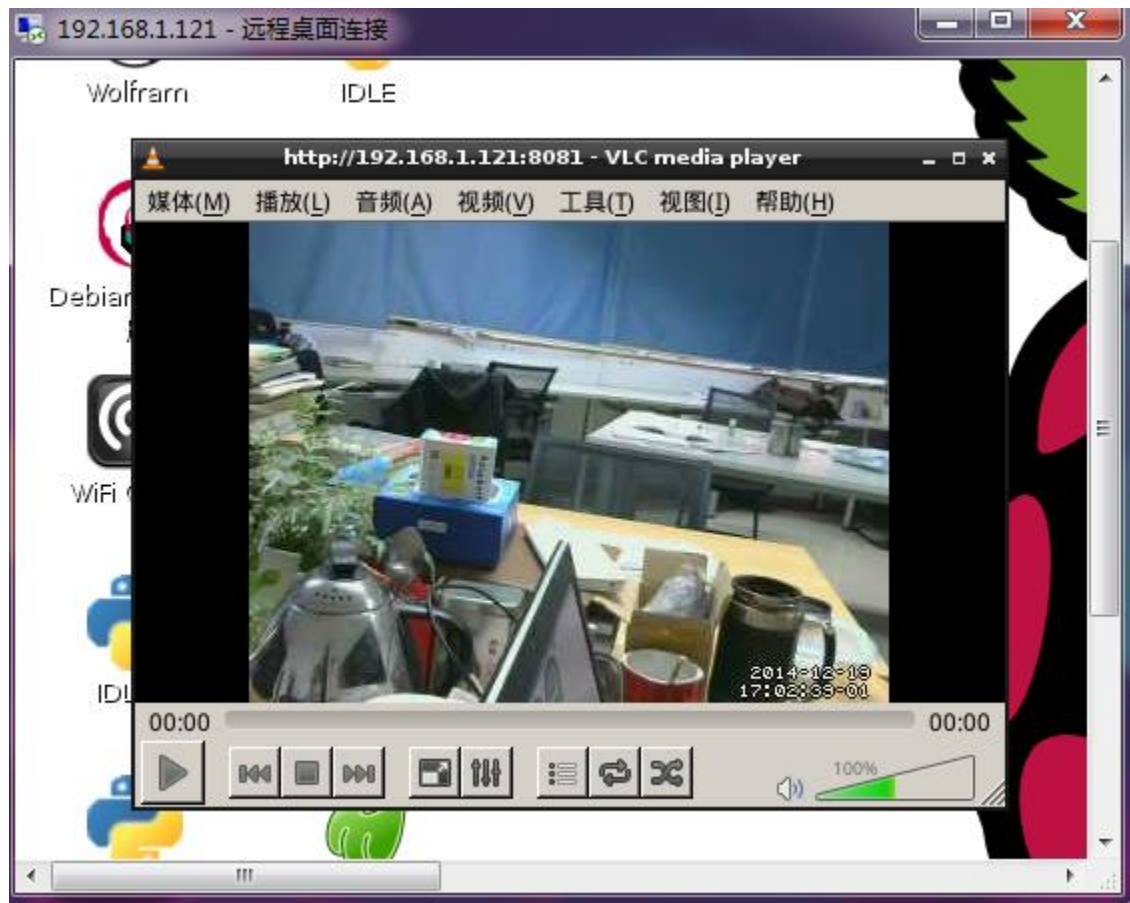
❑ 安装VLC

apt-get install vlc

❑ 系统告诉我们，
VLC是预装的。

❑ 直接在菜单（其他）上找到VLC，运行，填IP地址，结果与在PC上是一样的。

❑ 在手机上，下载VLC，一样！



7、另一种网络监控 (yeelink)

□ 1、安装抓图软件fswebcam

```
sudo apt-get install fswebcam
```

□ 2、登录

Yeelink(<http://www.yeelink.net/>), 完成注册过程。



欢迎使用 Yeelink

—— 免费、开放的平台，快速开启您的物联网时代

管理设备

⚠ 您还没有任何设备，请使用 [增加新设备](#) 来添加。

© Copyright 2013 青岛亿联客信息技术有限公司 | Powered by [Yeelink.net](#)

8、在yeelink上添加设备

□ 3、在用户中心增加一个设备——树莓派摄像头

□ 4、再为此设备增加一个图像传感器——摄像头。

注销

用户中心 首页

帐户

我的设备

增加新设备

管理设备

管理动作

API 文档

管理设备

树莓派摄像头

我的树莓派网络摄像头

设备类型

自备设备

设备图片

设备图片将在前台devices界面展示

上传图片

传感器

管理该设备的传感器并为它们增加触发动作, 以便在状态改变时通知您.

+ 增加一个传感器



摄像头

编辑 删除

摄像头

图像URL

URL: <http://api.yeelink.net/v1.0/device/165>

--	--	--



1993-2013
专注IT教育二十年

8、在yeelink上添加设备

□ 在添加了摄像头后，一个重要的信息要记下来：

传感器

管理该设备的传感器并为它们增加触发动作，以便在状态改变时通知您。

+ 增加一个传感器



摄像头

编辑 删除

摄像头



摄像头

图像URL

URL: <http://api.yeelink.net/v1.0/device/165>

图像URL

URL: <http://api.yeelink.net/v1.0/device/16587/sensor/28686/photos>

- 这是后面传输时要用到的设备ID和传感器ID
- 在本例中设备ID是：16587，传感器ID是：28686

9、树莓派上的命令脚本

- ❑ 5、在/home/pi下创建脚本yeelink.sh, 内容为以下两行命令, 其中的ApiKey和URL请改为自己设备对应的信息。

```
sudo fswebcam -d /dev/video0 -r 320x240 --bottom-  
banner --title "RaspberryPi @ Yeelink" --no-timestamp  
/home/pi/yeelink.jpg
```

- ❑ 这行是拍摄一张照片, 保存在本地: /home/pi/yeelink.jpg
curl --request POST --data-binary @"/home/pi/yeelink.jpg"
--header "U-ApiKey: xxxxxxxxxxxxxxxxxxxxxxxx"
http://api.yeelink.net/v1.0/device/1022/sensor/1294/photos

- ❑ 为脚本增加可执行权限

```
pi@raspberrypi ~ $ chmod +x yeelink.sh
```


10、POST命令解释

- ❑ 在curl --request POST --data-binary
@"/home/pi/yeelink.jpg" --header "U-ApiKey:
XXXXXXXXXXXXXXXXXXXXX"
<http://api.yeelink.net/v1.0/device/1022/sensor/1294/photos>
- ❑ 命令中，使用了POST上传图像命令。所谓：图像数据由key和value组成的键值对，图像内容则是二进制图像文件。
- ❑ Yeelink给出了上传和下载的API列表
- ❑ 上传图像
[/v1.0/device/<device_id>/sensor/<sensor_id>/photos/v1.0/device/<device_id>/sensor/<sensor_id>/photo/info/<key>](#)
- ❑ 获取图像（信息）
[/v1.0/device/<device_id>/sensor/<sensor_id>/photo/content/<key>](#)
- ❑ 获取图像（内容）
[/v1.0/device/<device_id>/sensor/<sensor_id>/photo/content/<key>](#)

10、POST命令解释

□ 上传图像:

- 对该URL的一个HTTP POST请求会为指定的图像传感器上传一幅新的图像, 使用此API来为图像传感器存储图像数据, 目前只支持上传jpg, png, gif类型的图像。

□ URL

http://api.yeelink.net/v1.0/device/<device_id>/sensor/<sensor_id>/photos

□ 请求参数

参数名	必需	类型	说明
device_id	true	string	设备ID
sensor_id	true	string	传感器ID

- 在本例中设备ID是: 16587, 传感器ID是: 28686

10、POST命令解释

□ 数据格式

binary

- 注意：此处格式与数值型传感器等不同，不再使用JSON格式，而是直接上传图片的原始内容

□ Method

POST

□ 返回值

HTTP Headers only

□ 访问授权

需要在HTTP Header中增加API Key来授权写入操作，具体请参考[API Key](#)，支持一次传送一幅图像数据。

10、POST命令解释

□ 请求实例

```
curl --request POST --data-binary @datafile.jpg --header "U-  
ApiKey: YOUR_API_KEY_HERE"
```

<http://api.yeelink.net/v1.0/device/12/sensor/3/photos>Note:
需要在HTTP Header中增加API Key来授权写入操作, 具体
请参照API Key. 支持一次传送一幅图像数据.

□ Note:目前限定相邻图像数据上传间隔须大于等于10s, 过于
频繁的请求会收到406 Response.

□ 其他参数请参看

<http://www.yeelink.net/developer/apidoc/12>

11、执行与结果

□ 手动执行一次

./yeelink.sh

□ 得到如下运行结果:

```
root@raspberrypi: #  
root@raspberrypi: ~# nano yeelink.sh  
root@raspberrypi: # ./yeelink.sh  
— Opening /dev/video0...  
Trying source module v4l2...  
/dev/video0 opened.  
No input was specified, using the first.  
— Capturing frame...  
Corrupt JPEG data: 1 extraneous bytes before marker 0xd1  
Captured frame in 0.00 seconds.  
— Processing captured image...  
Putting banner at the bottom.  
Setting title "Yeelink".  
Clearing timestamp.  
Writing JPEG image to '/root/yeelink.jpg'.  
root@raspberrypi: # █
```

12、设置为定时执行

- ❑ 6、将脚本加入cronjob定时任务

`crontab -e`

- ❑ 加入下面一行, 让脚本每分钟运行一次, 保存退出。

`*/1 * * * * /home/pi/yeelink.sh`

- ❑ `crontab`是一个Linux下的定时执行工具, 可以在无需人工干预的情况下运行作业。它是内置程序, 任何版本的Linux都带有这个程序。

13、crontab命令解释

❑ crontab命令选项:

- u 指定一个用户。
- l 列出某个用户的任务计划。
- r 删除某个用户的任务。
- e 编辑某个用户的任务。

❑ 可使用crontab -e编辑定时任务，crontab定时任务格式如下

Minute Hour Day Month Dayofweek Command

❑ 参数范围:

0-59 0-23 1-31 1-12 0-6 Command

❑ 例子:

30, 8-11 */1 * * ls

❑ 上述命令表示每天的8-11点的第30分钟执行一次ls命令。

14、实现网络监控

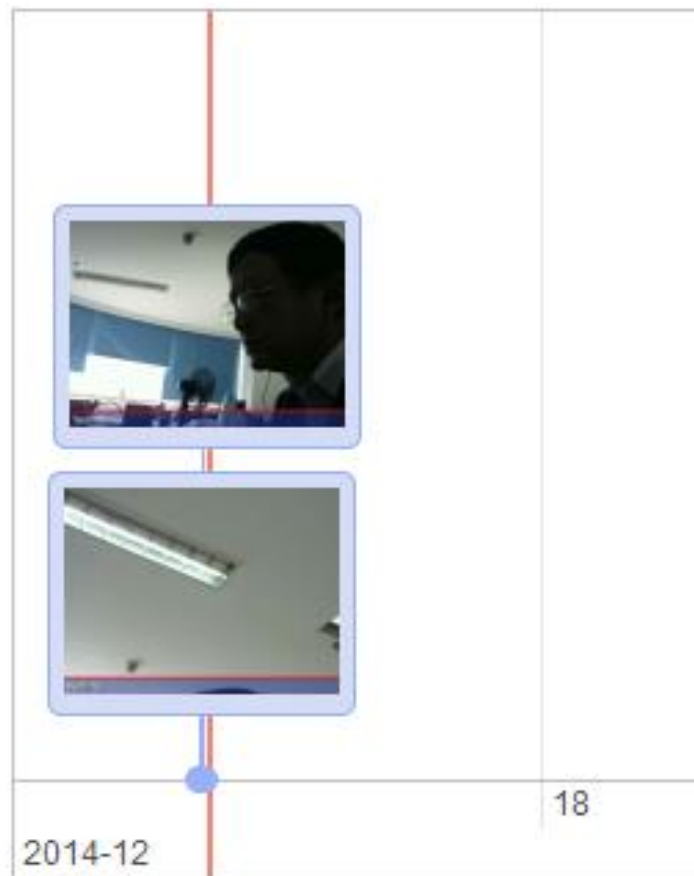
- 至此树莓派端的搭建已经完成，然后插上网线，连接电源，就可以自动进行传输了。
- 到Yeelink个人中心看看你的成功吧，如下图所示。为了保证图片传输速率，我在树莓派端搭建的时候将图片设置成了 320×240 像素，一般使用已经足够了，间隔拍摄时间是1分钟，从Yeelink时间线中可以看到，基本每分钟都会传输一张图片。

14、实现网络监控

- ❑ 在yeelink上，点击右边这样的图像URL
- ❑ 在设备/传感器下面，就可以看到已经有一张图传来了。
- ❑ 再运行一次，又来一张
- ❑ 这样把摄像头放在任何地方，自己坐在世界的另一个任何地方，都可以看到每分钟一张的实时监控照片了。
- ❑ Yeelink是否会被你的图像“撑”爆了？

图像URL

URL: <http://api.yeelink.net/v1.0/device/16587/s>



请注意! 图像未完全显示，请放大查看更多

14、实现网络监控

- 在你自己的手机上安装 yeelink 客户端APP
- 也可以看到上述内容。
- 在 yeelink 上的活跃应用有：

活跃设备



switch
on/off



电源开关
远程控制电源开关



控制中心



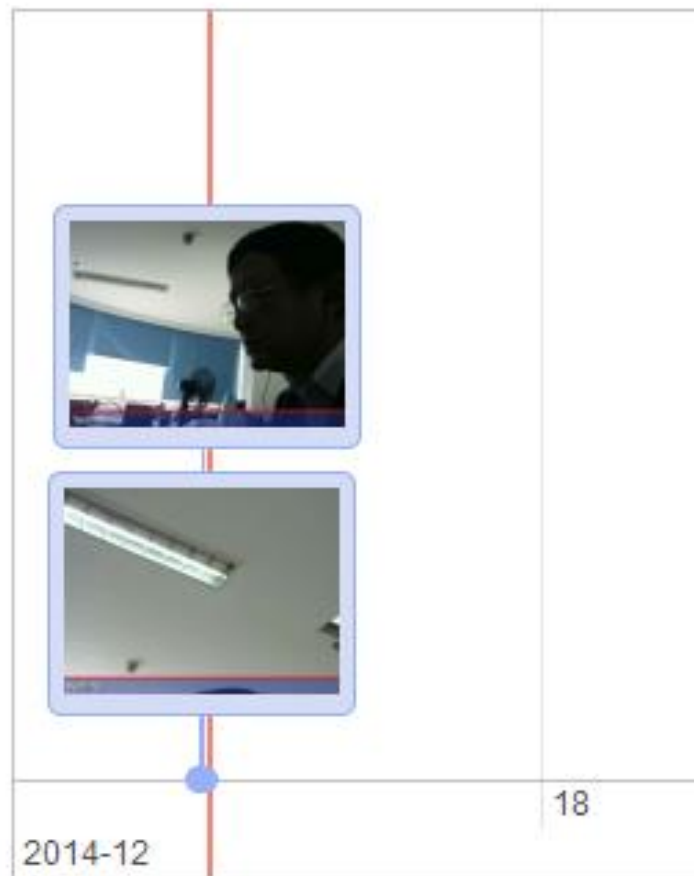
Troch
fisher's Troch



树莓派摄像头
我的树莓派网络摄像头

图像URL

URL: <http://api.yeelink.net/v1.0/device/16587/s>



请注意! 图像未完全显示, 请放大查看更多