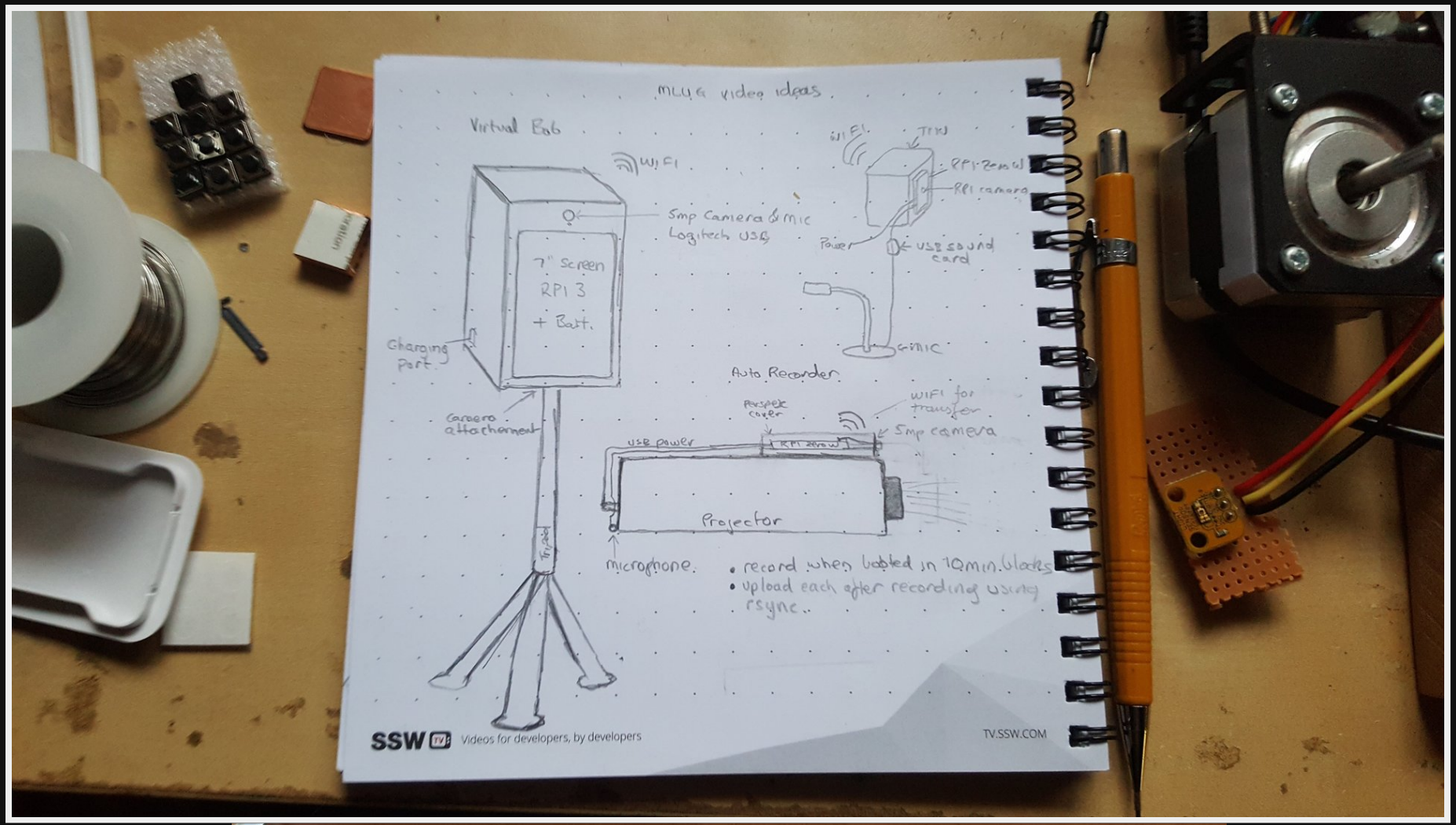
PICAM

Live stream and record using a Raspberry PI

Planning


MLH Video ideas
Virtual Bob
WiFi
5mp Camera & mic
Logitech USB
7" screen
RPI 3
+ Batt.
Charging Port
Camera attachment
WiFi
TTKJ
RPI Zero W
RPI camera
Power
USB sound card
Mic
Auto Recorder
Perspex cover
WiFi for transfer
5mp camera
USB power
RPI zero W
Projector
microphone
• record when battery in 10min. blocks
• upload each after recording using rsync..
SSW TV Videos for developers, by developers
TV.SSW.COM

Why I built it

- To record MLUG
- Stream to members who cannot make it, ie: Bob
- So others could easily create a portable recording device

Why use PICAM

- Easy to drive by touching files
- Optional encryption
- H.264/ACC encoded MPEG-TS
- Records/Streams Audio & Video

What I tried before PICAM

- The standard raspi-capture (no audio)
- RPi-Cam-Web-Interface - eLinux.org (no audio, but really good otherwise)

Hardware used

- Raspberry PI Zero W + case
- Raspberry PI camera 5MP
- USB sound card
- Analogue Microphone

Setup

- Used Raspbian Jessie Lite.
- Manually installed picam (there is a precompiled version)

Copy image

- Copy image to pi

```
sudo dd if=2017-04-10-raspbian-jessie-lite.img of=/dev/mmcblk0
```

- Boot
- Install updates

```
sudo apt update
sudo apt upgrade
```

- Setup locale

```
sudo dpkg-reconfigure locales
```

  Select en-AU.utf-8

WIFI

- Setup WIFI Add the following to /etc/network/interfaces

```
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

  Add WIFI details

```
sudo su
iwlist wlan0 scan
 wpa_passphrase <SSID> <passphrase> >> /etc/wpa_supplicant/wpa_supplicant.conf
```

  NOTE: I added a space in front of the wpa_passphrase command this is so no history is recorded here.

- Connect to WIFI

```
sudo wpa_cli reconfigure
```

- Wait a few seconds and it should connect.

```
sudo ifconfig
```

Multiple WIFI

- Store multiple WIFI connections in /etc/wpa\supplicant/wpasupplicant.conf

```
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
        ssid=mobile
        psk=password
  priority=10
}
network={
        ssid=home
        psk=welcome
  priority=1
}
network={
        ssid=work
        psk=123456
  priority=1
}
```

Compile ffmpeg

- Compile x264

```
mkdir ~/src; cd ~/src
git clone git://git.videolan.org/x264
cd x264
./configure --host=arm-unknown-linux-gnueabi --enable-static --disable-opencl
sudo make install
```

- Compile ffmpeg

```
cd ~/src
git clone git://source.ffmpeg.org/ffmpeg.git depth=1
cd ffmpeg/
sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfre
```

  This will take 4hrs

```
sudo make install
```

NOTE: Should of used Rick Miles cross compile here.

Stream on LAN

- Add the following to nginx.conf

```
rtmp {
  server {
    listen 1935;
    chunk_size 4000;
    application webcam {
      live on;

      exec_static /usr/local/bin/ffmpeg -i tcp://127.0.0.1:8181?listen
                                        -c:v copy -ar 44100 -ab 40000
                                        -f flv rtmp://localhost:1935/webcam/mystream;
    }
  }
}
```

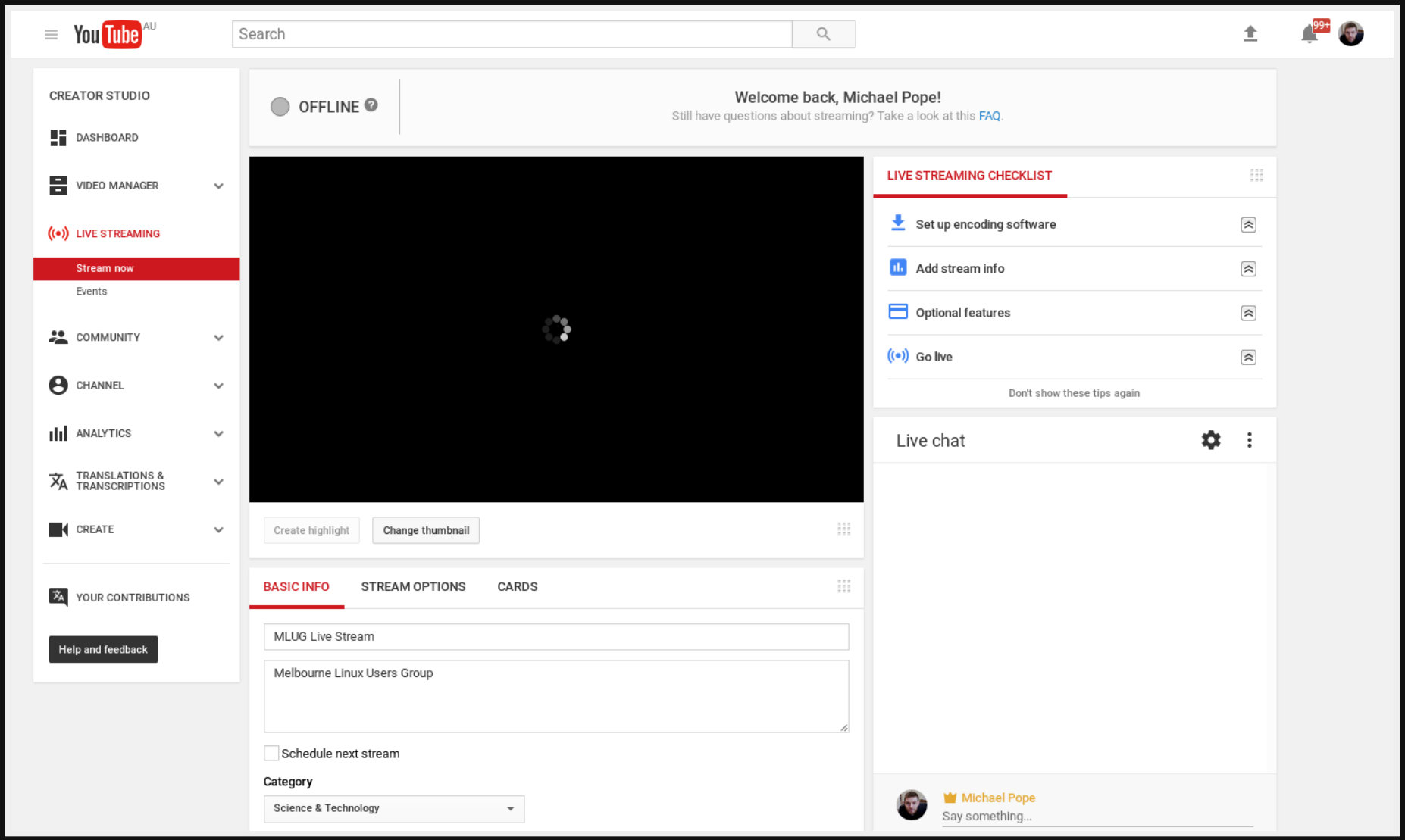
- Start picam with tcpout

```
~/picam/picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0
```

- Now you can access via LC

```
vlc rtmp://10.1.1.145/webcam/mystream
```

Youtube live setup



- Go to the youtube dash https://www.youtube.com/live_dashboard
- Verify your account with your mobile phone
- Enable live through the upload button in youtube.

Stream to Youtube through nginx

- Add the following to your nginx.conf file on the PI

```
# Stream to Youtube, 128k, video bitrate at 400k
 rtmp {
   server {
     listen 1935;
     chunk_size 4000;
     application webcam {
       live on;

      exec_static /usr/local/bin/ffmpeg -i tcp://127.0.0.1:8181?listen
         -c:v copy -ar 44100 -ab 128k -b:v 400k -b 2000000
         -f flv rtmp://a.rtmp.youtube.com/live2/<Your YOUTUBE API KEY>;
     }
   }
 }
```

- Restart nginx
- Start picam with tcpout

```
~/picam/picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0
```

- Now it should be streaming to youtube live.

Start/Stop recordings

- Start

```
touch ~/picam/hooks/start_record
```

- Stop

```
touch ~/picam/hooks/stop_record
```

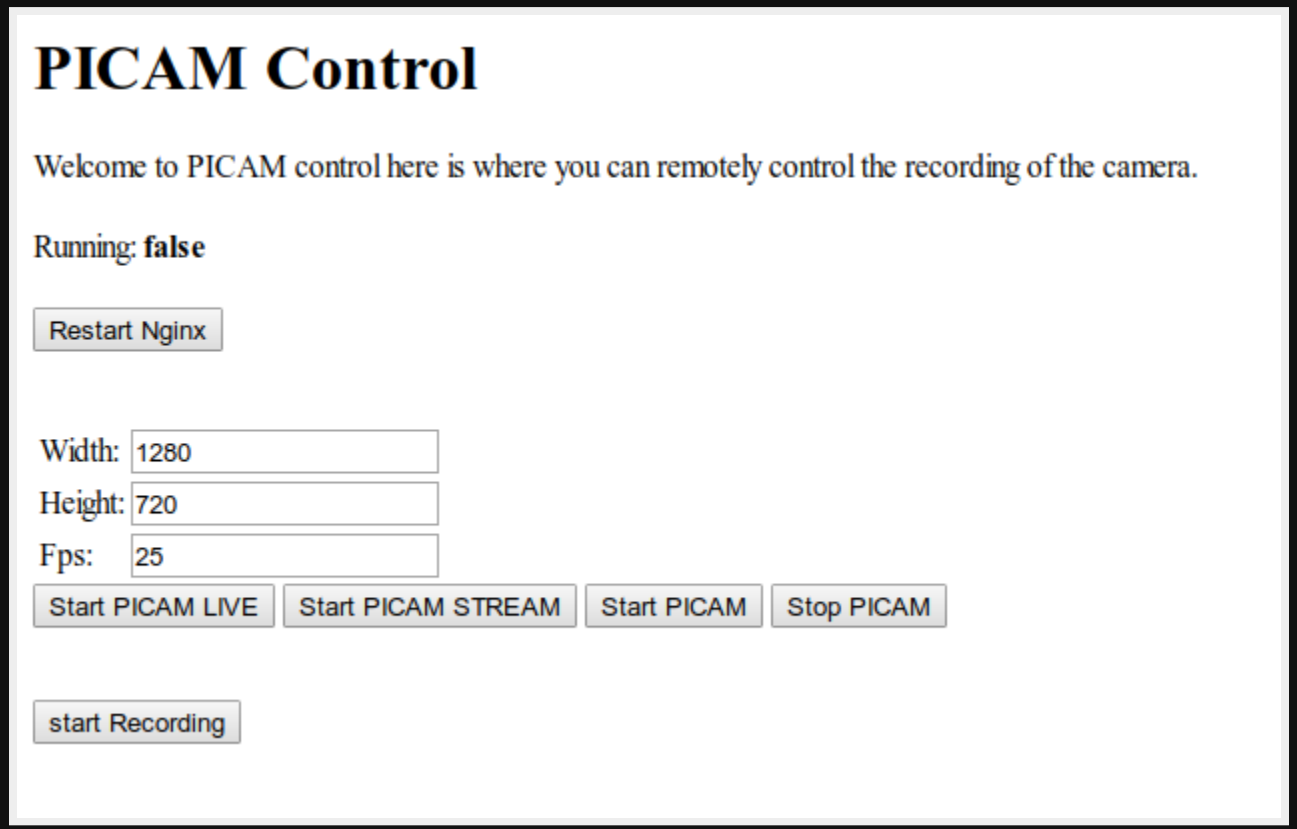
Space used tested with 1920x1080@25 fps

Length	Size	Resolution	Description
1min	18MB		Tested
60min	1080MB		Calculated

- Records into ~/picam/rec/archive as MPEG2-TS by default

Web frontend

PICAM Control



Tech

Langugage	Ruby
Framework	Sinatra
Address	http://<IP>:4567

Install requirements

- Install latest ruby Installing Ruby(2.4) on Rails(5) environment on Raspberry Pi 3 – Jean Brito
- Install sinatra

```
gem install sinatra sinatra-flash thin
```

Clone my code

- Clone my picam\control

```
cd ~/code
git clone https://github.com/map7/picam_control.git
```

Auto start web frontend

- Start picam_control on boot Add to /etc/rc.local

```
cd /home/pi/code/picam_control;ruby picam_control.rb &
```

Email internal IP on startup

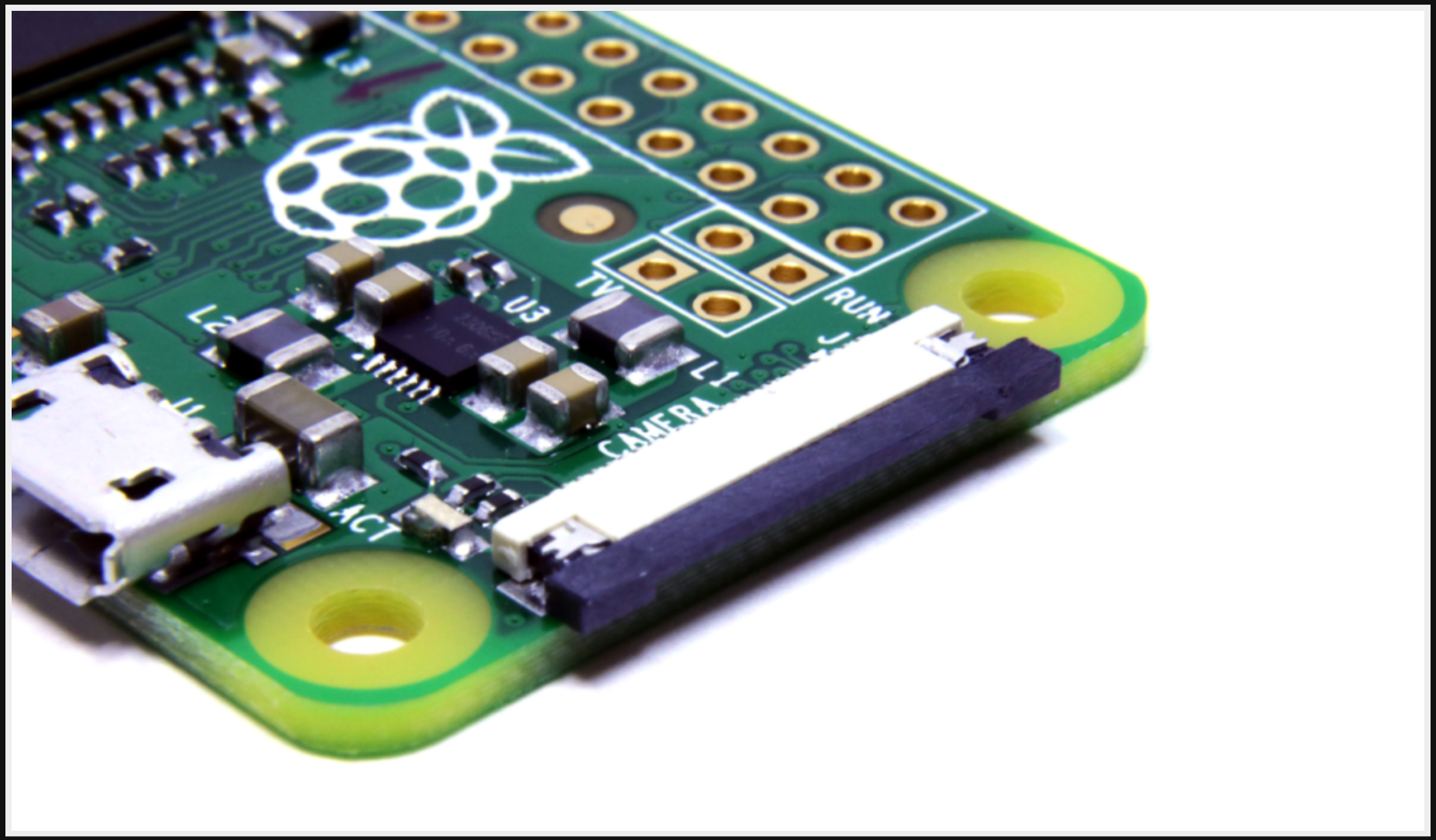
- Download startup_mailer from https://gist.github.com/johnantoni/8199088
- Add to startup to /etc/rc.local

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
    python /home/pi/code/startup_mailer.py
    cd /home/pi/code/picam_control;ruby picam_control.rb &
fi
```

Troubles

Snapped off camera connector

https://www.youtube.com/watch?v=bhaeKdUltz8



Coffee cup

Fixed connector

Test camera, thought it was broken

- Booted the camera up on a standard Raspberry PI
- Check camera exists

```
vcgencmd get_camera
```

- Take a picture

```
raspistill -o test.jpg
```

- Capture a video 10secs

```
raspivid -o video.h264 -t 10000
```

Bad internet

Had to scale back my resolution for streaming.

Tests

Setting different resolutions

```
# (240p SD) 352x240@8fps works at home on my poor internet
./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 352 -h 240 --fps 8

#    352 x 240 (240p SD)
#./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 352 -h 240 --fps 25

#    480 x 360 (360p)
#./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 480 -h 360 --fps 25

#    858 x 480 (480p)
#./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 858 -h 480 --fps 25

#    1280 x 720 (720p) (Half HD)
#./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 1280 -h 720 --fps 25

#    1920 x 1080 (1080p) (Full HD)
#./picam --tcpout tcp://127.0.0.1:8181 --alsadev hw:1,0 -w 1920 -h 1080 --fps 25
```

At home

Width	Height	FPS	Recording Result

Running with –tcpout through sinatra app to Youtube

Width	Height	FPS	Recording Result
352	240	15	Great
480	360	15	Great
858	480	15	Bad

Running without –tcpout through sinatra app

Width	Height	FPS	Recording Result
1280	720	25	Great
1920	1080	25	Great

Width	Height	FPS	Recording
			Result

Running with –tcpout through sinatra app

Width	Height	FPS	Recording
			Result
1280	720	25	Great
1920	1080	25	Great

Future

- Power off a projector
- Automatically send video files to my server once finished recording
- Replace youtube live with something open source
- Create a proper stand for the PI
- Use more compression to run at a higher resolution

References

- GitHub - map7/picam_control: Control my picam remotely
- GitHub - iizukanao/picam: Audio/video recorder for Raspberry Pi with language...
- GitHub - iizukanao/picam-streamer: Out-of-the-box SD card image for live stre...

Questions

Email	map7777@gmail.com
Twitter	@map7
Github	github: map7