

Simple Time-lapse Photography with Camera or NoIR Camera and Raspberry Pi:

By Mitch Neilsen, neilsen@ksu.edu

If you want to do everything from Windows, and not bother with making the changes on your own Raspberry Pi, you can skip ahead to the Quick Start for Windows Users Only at the end of this document.

To understand everything I did to simplify data collection for time-lapse photography, read on:

On Raspberry Pi, install default Raspbian GNU/Linux 7 distribution and enable camera.

Then, modify file `/etc/rc.local`, which is executed when the system is booted up, so that the system mounts the USB thumb drive under `/mnt/usb` and executes the commands in `raspistill.txt` to capture (.jpg) images and generate a video file (timelapse.avi) from the images, and then shut the system down. The original rc.local file looks like:

`/etc/rc.local:`

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

exit 0
```

We modify the file, as shown below in red, to tell the system to execute the commands listed on the thumb drive in `raspistill.txt`. Any editor can be used to edit the file; e.g., `vi`, `pico`, etc. Note that commands you type are shown in red:

```
pi@raspberrypi /etc $ cd /etc
pi@raspberrypi /etc $ sudo vi rc.local
```

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

#
# Added for time-lapse photography
#
mount -o uid=pi,gid=pi /dev/sda1 /mnt/usb
#
# Copy raspistill.txt to raspistill.cmd, convert to Unix format, and execute
#
cp /mnt/usb/raspistill.txt /mnt/usb/raspistill.cmd
dos2unix /mnt/usb/raspistill.cmd
sh /mnt/usb/raspistill.cmd
#
# End of commands added for time-lapse photography
#
exit 0

```

Some of the applications used are not installed by default, we can install them from the Internet using apt-get:

```

pi@raspberrypi /etc $ sudo apt-get install dos2unix
pi@raspberrypi /etc $ sudo apt-get install mencoder

```

Then, create a directory to be used to mount the USB drive, and set "pi" to be the owner of the directory:

```

pi@raspberrypi /etc $ sudo mkdir /mnt/usb
pi@raspberrypi /etc $ sudo chown pi /mnt/usb
pi@raspberrypi /etc $ sudo chgrp pi /mnt/usb
pi@raspberrypi /etc $ ls -l /mnt/usb

total 0
pi@raspberrypi /etc $ ls -l /mnt
total 4
drwxr-xr-x 2 pi pi 4096 Jan 14 19:30 usb

```

Of course, if you aren't able to connect your Raspberry Pi to the Internet, you can just download the image and flash it onto your SD card using Win32DiskImager, see below for Quick Start for Windows Users Only:

Then, we use any old Windows editor to create a file called raspistill.txt in the top-level directory on our USB thumb drive:

```
sleep 10
mkdir -p /mnt/usb/pictures
cd /mnt/usb/pictures
rm a*.jpg
rm timelapse.avi
raspistill -tl 5000 -o a%04d.jpg -t 30000
ls *.jpg > stills.txt
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf
scale=1920:1080 -o timelapse.avi -mf type=jpeg:fps=4 mf://@stills.txt
sync
sudo shutdown now
```

Here is what each command is for:

```
sleep 10 -- sleep for 10 seconds while the system finishes booting up
mkdir -p /mnt/usb/pictures -- create a pictures subdirectory if it doesn't exist
cd /mnt/usb/pictures -- jump to the pictures subdirectory
rm a*.jpg -- remove any old pictures
rm timelapse.avi -- remove any old video file
raspistill -tl 10000 -o a%04d.jpg -t 600000 -- take pictures every 10 seconds for 600 secs.
ls *.jpg > stills.txt -- create a list of image files in "stills.txt"
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf
scale=1920:1080 -o timelapse.avi -mf type=jpeg:fps=24 mf://@stills.txt
-- create video file from the images with 24 frames per sec.
-- the mencoder command is on one line
sync -- flush data being written out to the thumb drive
sudo shutdown now -- shut the system down when we're done
```

Note that the first example actually takes a picture every 5 seconds for 30 seconds. It's probably good to test a short running sample before you let it run for hours and hours. After 30 seconds, it will still take about 30 more seconds to create the video. Then, you can just turn off the Raspberry Pi, remove the thumb drive, and enjoy your video. If your time-lapse video includes hundreds of frames, it will take a bit longer to create the video. If the video is corrupt, you can construct the time-lapse in Windows from the image files in the pictures folder.

Quick Start for Windows Users Only:

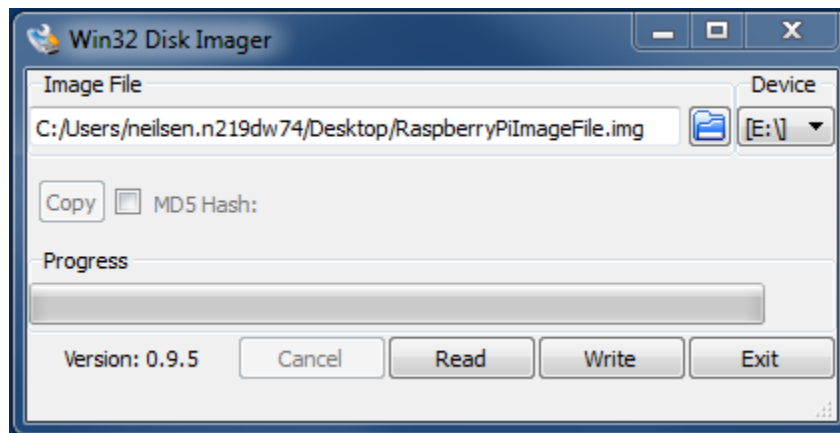
1. Download and install Win32DiskImager:

<http://sourceforge.net/projects/win32diskimager/files/latest/download>

2. Download the updated Raspberry Pi image (created from the Raspbian distribution as described above – added dos2unix, mencoder, and updated /etc/rc.local) – note it is around 8GB:

<http://www.cis.ksu.edu/~neilsen/RaspberryPi/RaspberryPiImageFile.img>

3. Use Win32DiskImager to create an image of the new Raspberry Pi distribution on the Raspberry Pi SD Card (assuming your SD card is in drive E, just click on the Write button to write the image onto your SD card):



4. Create a text file called raspistill.txt and save to your USB thumb drive top-level directory. As an example, see:

<http://www.cis.ksu.edu/~neilsen/RaspberryPi/raspistill.txt>

This was used to create the pictures and time-lapse video from my office in:

<http://www.cis.ksu.edu/~neilsen/RaspberryPi/pictures/>

5. Plug a single USB thumb drive into the USB ports on your Raspberry Pi, and set the camera, and start up the Raspberry Pi. It's probably good to test with a short running script first, then set it to run as long as you want. Note, you should not plug in a keyboard or mouse to the USB ports because they might be detected as device /dev/sda1, which will confuse the script. You can plug in an HDMI display device and watch the pictures getting taken. Enjoy!