LINUX VOICE TUTORIAL

RASPBERRY PI: MONITOR WOODLAND CREATURES

JON ARCHER

Set up a sturdy camera out in the woods and use Linux to take pictures of lions, tigers and bears.

PARTS REQUIRED

- Raspberry Pi
- SD Card (bigger the better)
- Pi Camera
- Waterproof case with see through area
- USB Wi-Fi dongle
- USB rechargeable portable battery pack

A trail camera will capture images of wildlife that frequent a certain area, such as woodland. These images, still or moving, can be captured without the need of the photographer to be present. The camera either constantly records video or uses motion detection technologies to trigger an image capture. Off-the-shelf versions of these are expensive, don't offer any option for customisation, and contain proprietary hardware and software.

A Raspberry Pi, with its fantastic range of hardware and software options, is an ideal platform to create a similar device with the potential to create your very own wildlife videos and photographs. Although this project is geared towards building a Pi-based trail camera, there are many other situations where this could potentially be deployed; it would also make a simple security camera, for example.

Installation

First, ensure your camera is connected to the Pi and that you have a network connection for installation. Now we can install the OS and the required software.

The central part of this project is a piece of software called RaspiMJPEG (which is based on the MMAL library) to control the Pi camera.

From the starting point of a base Raspbian install, we can start up the Pi and use the configuration tool to increase the free space, ensure that SSH starts on boot and set the password for the Pi user. One other vital step is to enable the Pi camera.

Once we have finished with the configuration tool the next thing is to ensure that the system is up to date (**sudo apt-get update**; **sudo apt-get upgrade**). One more update we need to pull in is the latest firmware for the Raspberry Pi. This includes the latest camera firmware, which is required by the RaspiMJPEG camera control software. This can be done by running **sudo rpi-update**.

Components cost money, but the beauty of the Pi is that everything can be re-used for your next project.

The box we used had an IP rating of 65, which essentially means it is nice and waterproof.

It should be part of the Raspian install, but we also need to ensure that Git isinstalled. This will be used to retrieve the software and scripts needed to complete this installation. Let's confirm it is installed by running the command **sudo apt-get install git**. We can now start to install the components required to provide the web interface, motion detection and live feed. Fortunately for us the majority of this is captured inside a script created by Silvan Melchior, who also created RaspiMJPEG. Run the command:

```
Git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
```

This will download the initial scripts for the install along with some configuration files and pre-compiled binaries (don't worry – these are open source, just pre-compiled to save time).

Using a cat5 for power and connection.

If cat5 were an option, then powering the Pi could be achieved by using a power over Ethernet injection kit. Also, if the cable is run over a considerable distance then voltage drop must be taken into consideration. Therefore a higher voltage power supply should be used and a voltage regulator at the Pi side to ensure that it receives the necessary 5V.

The PVC box is one option for housing the project, but the recently crowdfunded kickstarter campaign PiCE should also do the job quite nicely.

Software used in this tutorial:

- Apache httpd with PHP for the web interface
  http://httpd.apache.org & **www.php.net**
- Motion, used for the motion detection
  www.lavrsen.dk/foswiki/bin/view/Motion/WebHome
- Raspimjpeg – to interface with the camera and output as image/video/stream
  www.raspberrypi.org/forums/viewtopic.php?t=61771

You should see an output similar to:

```
Initialized empty Git repository in /home/pi/RPi_Cam_Web_Interface/.git/
remote: Reusing existing pack: 161, done.
remote: Total 161 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (161/161), 104.62 KiB, done.
Resolving deltas: 100% (70/70), done.
```

A new directory will be created called **Rpi_Cam_Web_Interface**. In this directory resides a script that will complete the rest of the installation. Use **cd** to move into the directory, and launch the script with

```
./RPi_Cam_Web_Interface_Installer.sh install
```

This script will go away and install all the required packages, of which Apache HTTPD, PHP and Motion are the most important.

Afterwards we have a couple of choices as to how the software will start, if at all, on boot. For this tutorial we will have it automatically start with motion detection. For this we need to edit the file **/etc/raspimjpeg**, the last line of which contains the line

```
motion_detection false
```

Use your favourite text editor to change this value to **true**, then re-run the install script with the option **autostart_yes** instead of **install** to set the software up to start on boot. That's all there is to the install. There is much that could be configured both within **/etc/raspimjpeg** or Motion, but for now we have a working system. Let's reboot!

Launch your favourite web browser from another PC and you should be presented with a live image from your camera with a series of buttons and a table of options underneath. If this is the case then our installation was a success.

Most of the buttons you see should be greyed out, with only the Motion Detection Stop button available. At this point you can test the motion detection by waving an object in front of the camera; subsequently clicking on the Download Videos And Images link you will see a video file listed.

Back on the RPi Cam control main page, the table of options presents a multitude of configuration; from here you can set image resolution, image quality, various levels of brightness, ISO, contrast etc. Experiment with these to find the best setup for you.

Enclosure and powering the device

As the Raspberry Pi and its camera will be outdoors, choosing a suitable enclosure is vital to ensure your Pi stays nice and dry. In our project we used a PVC outdoor electrical junction box (150x110x70 mm).

We were hoping to find badgers, but just got these deer. D'oh, a deer!

This box was all good and safe, but there was no opening for cables or view area for the camera to see out of. This is where a 45mm camera skylight lens and a hot glue gun came in handy.

If your Pi is already in a plastic case then simply glue this to the deeper side of the PVC junction box, otherwise some M3 size nylon stand-off spacers should be used to attach the board inside the box. You'll need to drill a hole into the junction box where the lens will be situated – ensure that when you attach the lens a liberal amount of glue and or sealant is used to ensure the box stays waterproof. Using one of the many available plastic camera mounts also helps with securing it inside the case with glue.

How you decide to power the device is all dependant upon the location you choose and the facilities available in that location. If your camera is to be situated in your garden or surrounding then laying a cat5 cable inconspicuously may be an option with some kind of power over Ethernet solution. Otherwise the Pi can be powered using a battery pack such as those used for emergency mobile phone charging, just make sure there is a reasonable capacity in the batteries such as 10,000mAh. If a battery pack is used then this must also be taken into consideration when deciding on an enclosure as extra room may be required. The downside to running on batteries would be that a live view would only be available if the box were be situated within the signal range of a wireless router. Using wireless would also have a bearing on the battery drainage and time available.

For simplicity we will power the Pi using a battery pack that will fit nicely inside the PVC box.

We won't go into the configuration of wireless dongles as this varies slightly for each device and is well documented, but once you have this configured and the battery pack fully charged, plug it into the Pi, secure your box and place it where you expect to see your target subject, then head back to your PC and watch the live feed through your browser. And don't forget to check the battery level regularly!

Jon Archer is a Free Software evangelist, Red Had ambassador and the founder member of RossLUG.