Building a Wildlife Cam with a Raspberry Pi

Introduction

At Porchester Junior School we had an idea to use a Raspberry Pi with its camera board as a remote webcam to watch the wildlife in our quiet area. The idea was to stream images from the outdoor camera to a screen within school. Before we began anything, we broke the idea down into the steps needed to achieve this idea;

1. Connect the Raspberry Pi to the school network.
2. Stream an image from the Raspberry Pi to another machine using the school network.
3. Access the Raspberry Pi without the need to plug in a keyboard / mouse /monitor.
4. Set the Raspberry Pi to login and run processes automatically.
5. Power it whilst outdoors.

We then began to experiment, and search the Internet for any help and advice. It took a long time, with lots of false starts and dead ends, but eventually we did find out how to make it work.
We have documented what we did (and where we obtained the information from) to help others who wish to create their own wildlife camera using a Raspberry Pi.

What do I need?

Have the following to hand;

- A Raspberry Pi (ModelB) - this will be used to set everything up
- A Raspberry Pi (ModelA) - this will be the device that is used remotely *optional
- Power supply for the Raspberry Pi
- The Raspberry Pi camera Board
- USB keyboard / USB mouse (or wireless equivalents)
- WiFi adapter
- Monitor / TV
- HDMI lead (or HDMI to VGA convertor)
- SDCard

Steps Taken

1. Create an SDCard with the Raspberry Pi 'out of the box' software - this can be created on Mac / Windows machines. Full Instructions for creating the SDCard can be found here; http://www.raspberrypi.org/downloads Alternatively, buy a pre loaded RasPi SDCard.

2. Once you have your SDCard ready, connect your Raspberry Pi to the camera module (instructions for this can be found here[1]), keyboard, mouse, power supply, monitor and the Internet (either wired or wirelessly - it doesn't matter at the moment)

NOTE: Because we were using a HDMI - VGA convertor with an older monitor we were unable to see any image from the computer when we began. The only way we found to get around this issue is to initially use a monitor / TV with a direct HDMI connection, as this allows the initial screens to be viewed.

3. Switch on the power to the Raspberry Pi. If everything is working, you should see a rainbow display on the HDMI connected monitor / TV, before the machine displays lines of text in the terminal view as it starts. The first time you use the SDCard you will be asked to choose an operating system (OS) from the supplied choices. We suggest selecting 'Raspbian' for this project. The Raspberry Pi will now begin to create the Raspbian OS on your SDCard.

4. Once completed, a configuration screen will present you with some additional choices. The important one that you need to scroll down to (using the arrow keys on your keyboard) is the camera option. Press return on the camera line, and then choose 'enable'. Once done, select 'finish' and leave the configuration settings.

5. If you are going to be using an older monitor with a VGA connection, you will also need to alter the config.txt file. You can see this in a tab above all of the OS choices. The file looks complicated, but just contains a list of commands. Most of these have a # before them. The # cancels out the command, and turns it into a comment that has no effect on how the Raspberry Pi works. There are a few lines that you will need to alter to make the Raspberry Pi display on a VGA monitor[2]

Look for these lines;

#hdmi_force_hotplug=1
#hdmi_drive=2

and remove the # on each line to change it from a comment to a command.

You can also alter the following lines to affect the resolution of your image;

#hdmi_group=1
#hdmi_mode=4

(remove the # and alter the numerical value)

6. Now you should be able to see your Raspberry Pi on the monitor / TV Screen. If everything has worked, it will be asking for the default username (pi) and password (raspberry) to start. If you are using a Wifi connection, you will need to move to step 7, however if you are using a wired network connection skip to step 8.

[1] http://www.raspberrypi.org/camera

[2] Information taken from http://www.makeuseof.com/tag/three-ways-to-display-your-raspberry-pi-on-a-monitor-or-tv/

7. Once you have entered your username and password, you need to go into the familiar desktop view of a computer. This is achieved by typing 'startx' then pressing return. On the desktop there is a wifi utility that will search out local wireless networks. Select your connection, enter your network password and connect.

8. If you are using the Raspberry Pi within a school, you are probably behind a proxy and will need to enter the proxy details. This is achieved by opening 'Midori' - the Raspberry Pi Internet browser - and entering the proxy details within the settings.

9. Now it is time to check for updates and upgrades. Either open the 'LXTerminal' within the desktop or, if you prefer, click the small red icon in the bottom right corner and return to the terminal view. Type the following line to check for updates;

sudo apt-get update <return>

Once completed, type the following line to get any upgrades installed;

sudo apt-get upgrade <return>

10. Now, it's time to test the camera and check it actually works! Type;

raspistill -o image.jpg <return>

(You can replace "image" with whatever you want to name the file)

An image should appear on the monitor / TV for a few seconds, before a picture is taken. If you also want to test the video feature of the camera, type;

raspivid -o video.h264 -t 5000 <return>

to create a 5 second video. (5000 is the number of milliseconds in the clip)

(You can replace "video" with whatever you want to name the file)

11. If everything works - brilliant! We can move on. If it doesn't, check you connected the camera to the Raspberry Pi correctly.

12. Before we continue - type

ifconfig <return>

and note the IP address of the Raspberry Pi. We'll need this for later on...

13. To use the camera to stream images, we found a free piece of software called 'motion'. The great feature of 'motion' is that images are streamed over a local network, rather than over the Internet. There is a detailed description of what is needed to install this, and get it working with the Raspberry Pi here[3] (this is the most straightforward guide to this process that we have found, and if you work through it carefully, everything will work fine). The relevant text from the website has been reproduced at the end of this user guide as a reference.

14. Test out the motion software. Make sure you are in the /mmal folder. If not, type

cd ~/mmal <return>

and then type

./startmotion <return>

[3] http://rbnrpi.wordpress.com/project-list/setting-up-wireless-motion-detect-cam/

If everything has worked then the camera light will come on. On another machine on the same network, open a browser (Safari works really well) and type in the IP address of the Raspberry Pi that you found earlier, followed by :8081 If everything has worked then you will see an image from the Raspberry Pi camera being streamed across your network onto another machine.

To stop the camera, type;

./stopmotion <return>

15. Optional step: If you wish to personalise the text that displays at the bottom left and bottom right of the camera image you can by typing

nano motion-mmalcam.conf <return>

When the editor opens look for, and alter, thes two lines;

text_left [writing will be shown here]

This is where you can add a description of the camera / location. Just type the words that you want displayed on screen. For example...

text_left The Bird Table Camera

text_right [a series of codes will be shown here]

This is where you can alter how the time / date is displayed. Use the following codes;

%H displays hours
%M displays minutes
%S displays seconds
%T displays HH:MM:SS

%d displays date
%m displays month
%Y displays year

You can also add separators between the codes like : or - For example...

text_right %T %d-%m-%Y

16. Now you need to create a way for the Raspberry Pi to automatically login whenever it is switched on. This can be achieved by making a few changes to some commands already set up on the Rasperry Pi. Firstly, open a terminal session and edit the inittab file. Type;

sudo nano /etc/inittab <return>

With the inittab file open, look for a line that says

1:2345:respawn:/sbin/getty 115200 tty1

(or something very similar to the above), and add a # at the beginning of the line to comment it out

Step 3: Add login program to inittab.

Below the line that you have just commented out, add;

1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1

to make the Raspberry Pi start without the need to enter the username and password

Press CTRL + O to save the file. Press Y to confirm, then press CTRL + X to quit the editor.

Turn off the Raspberry Pi then turn on again. Rather than see a prompt for username and password, it should proceed further and be ready to run.

17. Finally, when the Raspberry Pi is positioned outdoors and running we want to access it without the need to plug in a keyboard, mouse and monitor as this might not be possible. This can be achieved through Virtual Network Computing (VNC). In the terminal session type;

sudo apt-get install tightvncserver <return>

This will find and install a VNC system onto the Raspberry Pi. Once it has finished installing, type

vncserver <return>

to start it. When asked for a password, enter something that you will remember[4], and then choose 'No' for a view only password.

18. Create a file so that the VNC server will start automatically when the Raspberry Pi is turned on. To achieve this, create a new file in the init.d directory by typing;

sudo nano /etc/init.d/tightvncserver <return>

and then typing out the following instructions

```
#!/bin/sh
# /etc/init.d/tightvncserver
VNCUSER='pi'
case "$1" in
  start)
    su $VNCUSER -c '/usr/bin/tightvncserver :1'
    echo "Starting TightVNC Server for $VNCUSER "
    ;;
  stop)
    pkill Xtightvnc
    echo "TightVNC Server stopped"
    ;;
  *)
    echo "Usage: /etc/init.d/tightvncserver {startlstop}"
    exit 1
    ;;
esac
exit 0
```

Once you have entered this press CTRL + O to save the file. Press Y to confirm, then press CTRL + X to quit the editor. Type

sudo chmod 755 /etc/init.d/tightvncserver

[4] Choose something simple and memorable - like your school name!

to give the script executable permission, and then type

sudo update-rc.d tightvncserver defaults

to make the VNCserver run whenever the Raspberry Pi turns on.

The VNC information was obtained from here[5]

19. Install a VNC server on another machine, (we used RealVNC), run the software and enter the IP address of the Raspberry Pi when asked[6]. Enter the password you set up in (16) and you will then see the Raspberry Pi desktop remotely, allowing you to access the LXTerminal window on the Raspberry Pi desktop whilst it sits outside. This means that you can turn on (./startmotion) and turn off (./stopmotion) whenever you decide.

20. So far, everything has been created on a Raspberry Pi ModelB (with two USB slots and an ethernet connection). Turn off the Raspberry Pi, remove the SDCard, and place it in a ModelA Raspberry Pi (with the single USB slot and no ethernet connection). Insert the wifi dongle into the USB slot, and place everything into a watertight container to protect from the elements.

Why use a ModelA? The ModelA machine requires less power than the ModelB.

21. Depending on where your Raspberry Pi and camera will be positioned, you also need to consider how to power it. Options could include wired from the mains, using Low Voltage, a Leisure Battery (caravan, not car), or Solar / Wind Power.

We've just tested a 6V fire alarm battery that powered a ModelB without problem. The idea now is that we'll use one battery whilst another charges in school, and change over in a morning as needed before the children arrive in school. As the Raspberry Pi will only be running during the school day, the power should be fine. We will also be looking at the possibility of an additional solar panel to trickle charge the battery to extend the charge of it.

[5] http://www.neil-black.co.uk/raspberry-pi-beginners-guide#.Uj9KOBZTPmk

[6] If the IP address kicks up an error, add a :1 at the end of it.

~ Appendix ~

Setting up motion

(this text has been taken from the URL below in case you cannot access the page during setup)

http://rbnrpi.wordpress.com/project-list/setting-up-wireless-motion-detect-cam/

Install the motion program
sudo apt-get install motion

We don't actually use this version, but it does insure that all the necessary dependencies are installed as well

sudo apt-get install libjpeg62

create and enter a directory called mmal in the user pi home directory using
cd ~/
mkdir mmal
cd mmal
download dozencrow's version of the motion program using
wget https://www.dropbox.com/s/xdfcxm5hu71s97d/motion-mmal.tar.gz
and extract the contents with
tar -zxvf motion-mmal.tar.gz
you now modify the configuration file, by opening it with the editor nano
nano motion-mmalcam.conf
nano hints
To insert a blank line, move to the beginning of a line and press return
To delete a line position the cursor on the line and press ctrl+k
use the backspace arrow to delete characters
you can search for a string by typing ctrl+w and inserting the string you wish to find.
When you have finished amending the file use ctrl+o to write out the amended file, selecting the existing filename to overwrite by pressing return. Then select ctrl+x to exit the editor.

you are going to alter the lines for
width
height
target_dir
output_pictures
text _left
logfile
search for them in turn using ctrl+w and set them as follows

width 640
height 480
target_dir /home/pi/m-video
output_pictures off

text_left Pi-cam %t
logfile /home/pi/mmal/motion.log

then write the changes out using ctrl+o push return and exit the editor with ctrl+x

So that there is no confusion as to which motion binary is running I rename the one in the mmal directory using mv motion motion-mmal

Assuming your camera board is correctly installed, you are now in a position to test it out
In your terminal window, from the mmal directory type
./motion-mmal -n -c motion-mmalcam.conf
where upon the camera board red led should light, and if you start a browser on another computer on your local network (Internet Explorer doesn't work but Chrome, Firefox and Safari are ok) you should see live pictures at by typing [your Raspberry Pi IP address in a browser window]

You quit the process by typing ctrl+c

The start stop process is hardly ideal, so I have written a couple of scripts to start and stop the process.

type nano startmotion
In the editor window type
#!/bin/sh
nohup ~/mmal/motion-mmal -n -c motion-mmalcam.conf 1>/dev/null 2>&1 </dev/null &

type ctrl+o then push the return key to write the text, followed by ctrl+x to exit the editor
chmod 755 startmotion to make it executable

type nano stopmotion
In the editor window type

```
#!/bin/sh
ps -ef | grep motion-mmal | awk '{print $2}' | xargs kill
```

note: I put the script above in a code block as wordpress changes the quotes to smart ones otherwise which don't work if you copy and paste

type ctrl+o then push the return key to write the text, followed by ctrl+x to exit the editor
chmod 755 stopmotion to make it executable

now you can type ./startmotion to start the camera and ./stopmotion to stop it
The videos created are stored in the m-video in the mmal home directory and the process is logged in the motion.log file