Building a Security Camera with a Raspberry Pi

My daughter recently had a thief enter her apartment and steal her laptop computer. Later, when I asked her what she wanted for Christmas, her answer was immediate: a security camera, so in case she is robbed again she will have pictures of the thief to give to the police. But commercial security cameras are expensive, often several hundred dollars, and many are tied to security services with expensive monthly fees, which she wasn't interested in paying.

This tutorial describes the construction of a security camera using a Raspberry Pi single-board computer as the embedded controller. The parts cost about a hundred dollars, construction takes an hour or two, and the only on-going monthly cost is to provide an always-on internet connection with a wifi router, which you likely already have. The result is a camera that saves pictures offsite and sends an email whenever it detects motion. If you don't have an internet connection, you can still build the security camera as described here, but pictures will be stored in the camera, rather than offsite, so if the thief steals the camera, he will take his picture with him.

Hardware: Six parts are required to build the security camera, all readily available from on-line suppliers such as Amazon, Newegg or Adafruit or from brick-and-mortar stores like Fry's Electonics or Micro Center:

Raspberry Pi Model B+	$ 35
Raspberry Pi Camera Module	30
Wifi dongle	12
5V 1A power supply, micro USB	7
8GB Micro SD card, Class 10	6
Case	10
Total	$100

Copyright © 2015 by Programming Praxis of Saint Louis, Missouri, USA. All rights reserved. This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License; to view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/us/ or send a letter requesting a copy of the license to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The code described in this document may be used under the GNU General Public License 3; to view a copy of this license, visit http://www.gnu.org/licenses/gpl-3.0.txt or send a letter requesting a copy of the license to Free Software Foundation, 51 Franklin Street, Fifth Floor, Boston, Massachusetts, 02110, USA. The code presented in this document has been included for its instructional value. It has been tested with care but is not guaranteed for any particular purpose. The author does not offer any warranties or representations, nor does he accept any liabilities with respect to the code. "Programming Praxis" and the "sharpen your saw" logo are trademarks of Programming Praxis of Saint Louis, Missouri, USA. You can find this essay on the internet at http://programmingpraxis.com/essays, or contact the author at programmingpraxis@gmail.com. Published on January 12, 2015.

The Raspberry Pi Model B+ single-board computer is used as the embedded controller of the security camera system, chosen because it is inexpensive, widely supported, and sufficient for our needs. The $20 Raspberry Pi Model A+ will also work, though setup is harder because it doesn't have an ethernet port.

For the camera we choose the Raspberry Pi Camera Module because it is small and easily hidden. If you want low-light infrared pictures, you could instead choose the Raspberry Pi Infrared Camera Module. The camera plugs in on the surface-mounted port near the ethernet port, with the wires facing away from the ethernet port.

The security camera communicates with the internet using wifi, so you will need a thumb-nail sized wifi dongle. Any brand will do. In the early days of the Raspberry Pi, some wifi dongles were supported while others were not, but most of that problem is now in the past, and you should be able to use just about any wifi dongle.

Any 5-volt 1- or 2-amp power supply with a micro-USB connector will be suitable to supply power to the security camera. Beware that some telephone-charger power supplies, though they look suitable, provide only 500mA or 700mA and should not be used.

You will need a micro SD card of 4GB or larger. Be sure to buy a faster Class 10 card rather than the slower Class 4 card. Although it's not strictly necessary, you will probably want to buy a case to protect your Raspberry Pi. We assume that your environment includes a wifi router connected to the internet, so you can send pictures off-site.

Preparing the Raspberry Pi: The first step in the construction of the security camera is to perform the basic setup of the Raspberry Pi. It is also the trickiest step, with several different ways to do things depending on what equipment you have available. If you have a pre-imaged SD card, you can connect a monitor, keyboard and mouse to the Raspberry Pi and set it up directly. Otherwise, you can image the SD card on some other computer, then connect the Raspberry Pi to your router and access it via `ssh`. And there are other ways to get set up, depending on your situation.

Instead of trying — and doubtless failing — to cover all the possibilities, we will point you to two useful resources. The Raspberry Pi Foundation, which produces the Raspberry Pi,

has a web site at `www.raspberrypi.org` that provides tutorials and forums to get you started: click on the `Help` tab and look around, or ask in the `Forums` if you have specific questions. Don't be shy, and don't think your question is too stupid to ask; everyone on the forum was once in your position, and they read the forums because they are happy to help new users get started. Another useful site is Adafruit, which sells and supports the Raspberry Pi; see the tutorials at `http://learn.adafruit.com/category/raspberry-pi`.

When you finish setting up, you should have a Raspberry Pi running the Raspian operating system, configured with 128MB or more of graphics memory and with the camera enabled, able to login to user `pi` with password `raspberry` and able to access a terminal window via the graphic interface or `ssh`.

All system setup of the security camera is done by editing configuration files. If you're not familiar with working at the command line, an easy editor is `nano`. To edit a file, open the terminal window, then say `sudo nano /path/to/file`; for instance, to edit the `/etc/motion/motion.conf` file, open the terminal window, type `sudo nano /etc/motion/motion.conf`, and press the `Enter` key. Sometimes you will be prompted for your password; if you are, type your password `raspberry` and press the `Enter` key. The `nano` editor has a window in which your text appears, and a menu of control-key combinations at the bottom. If you need help with `nano`, go to `http://www.nano-editor.org`. Some of the configuration is done in the `crontab` file; to edit that file, say `sudo crontab -e`, then use the `nano` editor to edit the `crontab` file as recommended.

If you haven't done so already, you should execute the following commands in a terminal window:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo rpi-update
```

Those commands bring the software on your Raspberry Pi up to date. To keep your Raspberry Pi up to date in the future, automatically, add this line to `crontab`:

```
15 3 * * sun sudo apt-get update && sudo apt-get -y upgrade
```

This command causes the `cron` daemon to run the `update` and `upgrade` commands weekly, every Sunday morning at 3:15am; we don't run `dist-upgrade` or `rpi-update` because they require a reboot, though you may want to run them manually from time to time. If you want to know more about `cron`, `http://www.raspberrypi.org` has a page that will help.

By the way, SD cards aren't particularly robust, so it is a matter of when, not if, your SD card will become corrupt. There's no need to back up the pictures taken by the security camera, since they are all stored offsite and are useless once you know that you haven't had a break-in, but once you have a working system you should make a copy of the SD card so you don't have to do all this configuration again when your SD card becomes corrupt. On a Linux or Mac computer, you can copy the SD card by inserting it into another computer and entering the following command, substituting for `/dev/sdb` and `/path/to/raspi` as appropriate:

```
sudo dd bs=4M if=/dev/sdb of=/path/to/raspi
```

It's not hard to find the device where your SD card resides. Say `df` before inserting the SD card in the slot, then say `df` again after inserting the SD card in the slot, and note the two new devices that are reported; they will differ in their last character, but the prefix of both devices will be the same, and that is the `if` input file that you should use. To copy the saved image onto a new SD card, just reverse `if` and `of`. Windows systems have other ways of performing this function, which you can find by consulting Google (search for "windows usb image writer").

Network Configuration: The next step configures the network. You may have done some sort of network configuration as part of the basic setup instructions, but it will be easier to configure the security camera using a static ip address, whereas most basic setup instructions provide a dynamic ip address via DHCP.

To configure the network you will need three pieces of information: the ip address assigned to the Raspberry Pi, the netmask, and the local network address of your router. You may have determined each of those values during the initial setup, but if not, a little detective work is necessary. The `ifconfig` command gives you the internet address and netmask of your Raspberry Pi, and the `netstat -nr` command gives you the local network address of your router, which may be called the "gateway" address. It is likely that the netmask will be 255.255.255.0, and the first two octets of the ip address and gateway address will likely be 192 and 168, as these numbers are reserved for local networks.

You also need to know the userid and password by which you connect to the router. If you rent your router from your internet service provider, your internet service provider gave you that information at the time you started service; mine is on a sticker attached to the router. If you bought a router, that information may be in the documentation or printed on the box. You might also look for your router at `http://portforwarding.com`.

Once you have everything you need, remove the wifi dongle and connect via `ssh` and edit the `/etc/network/interfaces` file to add a `wlan0` interface; the file should look like this:

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static
    address 192.168.1.65
    netmask 255.255.255.0
    gateway 192.168.1.254
    wpa-ssid "YOUR_USERID"
    wpa-psk "YOUR_PASSWORD"

iface default inet dhcp
```

After you've finished the network configuration, you should shut down the Raspberry Pi with `sudo shutdown -h now`, remove the network cable, insert the wifi dongle, reboot, and make sure that you can still connect with the command `ssh pi@192.168.1.65` or whatever ip address you use.

Motion: The heart of the security camera system is the `motion` command. Although you will want to read the guide at `http://www.lavrsen.dk/foswiki/bin/view/Motion/-MotionGuide` to learn all of its capabilities, it's not too hard to get started. Begin by installing `motion` with the command `sudo apt-get install motion`.

Before looking at `motion` itself, you need to install the V4L Video For Linux driver that `motion` uses to connect to the camera. For testing, issue the command `sudo modprobe bcm2835-v4l2` (that's the letter ell, not the digit one). To install the driver every time the Raspberry Pi boots, add the following line to the `/etc/modules` file:

```
bcm2835-v4l2
```

You will also need to create a directory where `motion` writes files by saying `sudo mkdir /var/run/motion`, then set permissions `sudo chmod 777 /var/run/motion` so user `pi` is able to access the directory. Also create a directory to store images by saying `sudo mkdir /tmp/motion` and `sudo chmod 777 /tmp/motion`. And you probably want to turn off the lamp on the camera, so as not to draw attention to it, by adding this line to the `/boot/config.txt` file:

```
disable_camera_led=1
```

The actions of `motion` are controlled by the `/etc/motion/motion.conf` file, which you should make accessible to user `pi` by saying `sudo chmod 777 /etc/motion/motion.conf`. The file is large, with many options described in the guide. Here is a recommended list of options for you to change:

```
daemon on
width 320
height 240
threshold 500
gap 3
snapshot_interval 600
text_right \%m/\%d/\%y \%T
target_dir /tmp/motion
webcam_port 0
on_event_start mail -s "Motion detected"  YOUR_EMAIL_ADDRESS < /dev/null
on_picture_save mpack -s "Living Room" \%f YOUR_ACCOUNT@sendtodropbox.com
```

The `daemon on` parameter starts the program operating.

The width and height of your picture are controlled by the `width` and `height` parameters. You will have to experiment to determine the width and height that are suitable for you. You don't want your picture to be too small, because then you won't be able to recognize the thief. You also don't want the picture to be too large, because it will take too much time to process each picture during a motion event. You might like to try 480×352 or 640×480 to see how they work. Be aware that `motion` requires both `width` and `height` to be multiples of 16.

The `threshold` parameter determines how many pixels must change between two sucessive pictures before `motion` determines that a motion event has started. A smaller value makes the system more sensitive to small changes, such as when a pet walks through the camera range or blowing air roils the vertical blinds. A too-large value might cause you to miss a thief. You will probably want to try several different values before you decide how to set this parameter.

The `gap` parameter tells `motion` to determine the end of a motion event when the indicated number of seconds has passed without motion. The supplied configuration file sets this parameter to 60, but you probably want to make it much smaller, because during each motion event `motion` will be saving pictures every second, and there's no need to see multiple versions of the same picture at the end of the motion event. If you make `gap` too small, however, you will get many email notifications when motion occurs, one at the start of each of several motion events that occur in quick succession.

The `snapshot_interval` parameter causes a picture to be saved every ten minutes. This is a "heartbeat" picture that lets you know the system is alive and operating.

The `text_right` parameter causes the current date and time to be printed in the lower right hand corner of the picture. The `target_dir` parameter specifies where pictures are stored. The `webcam_port` parameter turns off the internal web server, which is not needed for this application.

The `on_event_start` parameter sends email to your account at the start of each motion event using the Linux `mail` command. The `-s` parameter specifies the subject line of the email; change it to your liking. The `< /dev/null` parameter causes the body of the email to be empty. You should specify an email address in `YOUR_EMAIL_ADDRESS` that you monitor regularly, so that you are notified of motion on a timely basis.

The `on_picture_save` command causes a picture to be saved to your Dropbox account, for both the heartbeat picture and any pictures taken during motion events. The parameters of the `mpack` command are the subject, which describes the location of the camera in case you want to distinguish between multiple security cameras; the filename of the picture being saved, specified as `%f`; and the email address specific to your Dropbox account, which we discuss shortly.

It will take some experimentation to determine the proper settings for many of these parameters. You should also read the manual for the `motion` command, which has many other parameters that may be of interest.

You will want to periodically remove old pictures. To do that automatically, add this line to `crontab`, which deletes all pictures over two weeks old every morning at 3:10am:

```
10 3 * * * find /tmp/motion -name *.jpg -mtime +14 -exec rm -f \{\} \;
```

To interactively start or stop the camera, say `sudo /etc/init.d/motion start` or `sudo /etc/init.d/motion stop`. If you like, you can add those commands to your `crontab` to start and stop the camera each day at specified times; for instance, these commands set the camera to run each weekday from 7:15am to 5:45pm:

```
15 7 * * mon,tue,wed,thu,fri /etc/init.d/motion start
45 17 * * mon,tue,wed,thu,fri /etc/init.d/motion stop
```

If you want the camera to start automatically every time you boot the Raspberry Pi, add the following line to the `/etc/default/motion` file:

```
start_motion_daemon=yes
```

Offsite File Storage: As we have said previously, we want to send our pictures offsite for safekeeping. There are many places to send them — Google Drive, Amazon S3, another Raspberry Pi at your neighbor's house — but we choose Dropbox, because it is simple to set up and works well by sending the pictures via email.

The standard Raspbian operating system doesn't include an email system, so we will use Google's servers to process the mail, using a system called `ssmtp`, which you can install with the command `sudo apt-get install ssmtp mailutils mpack`; a good description of the process is available at `http://iqjar.com/jar/sending-emails-from-the-raspberry-pi/`. Begin by setting up a Google account at `http://accounts.google.com`; if you already have a Google account, you will probably want to create a new one, so that your camera's use of the account doesn't interfere with your primary email account. Then edit the `/etc/ssmtp/ssmtp.conf` file to include the following lines, substituting the username and password of your new Google account where indicated:

```
root=postmaster
mailhub=smtp.gmail.com:587
hostname=raspberrypi
AuthUser=YOUR_USERID@gmail.com
AuthPass=YOUR_PASSWORD
UseSTARTTLS=YES
```

It's not ideal to store your username and password in cleartext in the configuration file, but this Google account is used only for your camera to forward mail, so it doesn't matter too much; that's why we recommend a new Google account. As a matter of basic password hygiene, do not use the same password for this account that you use anywhere else. You will need to edit the `/etc/ssmtp/revaliases` file to allow user `pi` to send email, and we also enable the `root` user just to be polite.

```
pi:pi@raspberrypi:smtp.gmail.com:587
root:root@raspberrypi:smtp.gmail.com:587
```

Finally, execute the commands `sudo chmod 774 /etc/ssmpt/ssmtp.conf` and `sudo chmod 774 /etc/ssmtp/revaliases` to give user `pi` access to the configuration files. Then, while you're logged in to your new Google account, go to `https://www.google.com/-settings/security/lesssecureapps` and enable access for less secure apps. That setting allows you to send email through Google's servers without the normal two-factor authentication that Google requires, which is impossible to automate; this is another reason to set up a new Google account instead of using an existing account.

To store files at Dropbox you will need an account; if you don't already have one, go to `http://dropbox.com` and create one. Then go to `http://sendtodropbox.com` and create an account that will let you email your pictures to your Dropbox account; the email address that you are given at `sendtodropbox` should be copied into the `on_picture_save` parameter in the `motion.conf` file. You should set up the `sendtodropbox` account to store pictures in the `Attachments/Date/Filename` destination; if you have multiple cameras with the camera name in the subject line of the email, store them in the `Attachments/Date/Subject/Filename` destination.

Unlike Linux `cron`, Dropbox provides no way to automatically delete files after a specified amount of time, so you will have to periodically log in to your Dropbox account and delete each day's folder manually.

Wrapping Up: You should situate your camera so it has a good view of the entrance to your apartment or the door to the room where your valuables are stored, to get a good picture of the thief. In some cases you might want to have more than one security camera operating in different locations. Be sure to arrange sufficient lighting to get a good picture.

You can display the camera openly, or hide it. One possibility is to buy a security camera housing (they are designed to be used as decoys, cheap at about $10, and watertight so you can use them outdoors) and fit the Raspberry Pi and camera inside it. On the other hand, my daughter's security camera is hidden in a box of kleenex with a quarter-inch hole cut in the side and the Raspberry Pi sitting underneath the stack of tissues; the power cord that comes out the back of the box is hidden in the clutter on the kitchen counter. If you can't decide, you might have *two* cameras, one obvious, the other hidden.

Thankfully, my daughter's security camera hasn't caught a thief. But it does get tested every day when she walks in the front door and gets a "Welcome Home" picture of herself.