

AFFIDAVIT OF MINA CHING

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.
2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.
3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.
4. The archived data made viewable and browseable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.
5. The Internet Archive assigns a URL on its site to the archived files in the format `http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL]` aka an "extended URL". Thus, the extended URL `http://web.archive.org/web/19970126045828/http://www.archive.org/` would be the URL for the record of the Internet Archive home page HTML file (`http://www.archive.org/`) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.





6. Attached hereto as Exhibit A are true and accurate copies of browser screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each screenshot.
7. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

DATE: 11/04/2025 _____



Mina Ching

Please see attached
All Purpose
Jurat form
for additional
Notary Events



EXHIBIT A



<https://web.archive.org/web/20150130074036/https://musingsofadrian.blogspot.com/2013/04/raspberry-pi-controlled-canon-dslr.html>



Musings of Adrian

This blog will detail the trials, tribulations & sometimes even successes of projects that I find of interest. I hope they may be of use to someone other than myself.

Monday, 8 April 2013

Raspberry Pi Controlled Canon DSLR

As I work from home, overlooking the garden, I decided to build myself a bird table to have something interesting to look at whilst working out whatever network problem has fallen into my lap that day.

I am by no means a carpenter but I think it turned out OK, and the birds seem to like it.



Click for Larger Version

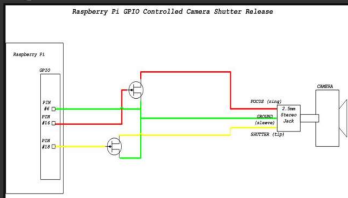
Next it was time to take some pictures of the visitors I had, this turned out to be very difficult with only a 250mm zoom lens. As soon as you moved the little blighters flew off! plus I had work to do, couldn't do both at the same time.

Having made a DIY shutter release for my Canon1000D camera the obvious next step was to build a remote release and the device most suited to this task was of course my RaspberryPi.....

This post will not cover setting up and updating your RPi, there are plenty of those already, suffice to say, keep your system updated. There is also a myriad of websites detailing the physical differences between camera shutter release connections, please check your own camera/connectors. Although each one will have at least 3 pins to connect to, these are usually pre-focus, shutter & ground. You'll need to know which is which.

So, let's get to it.....

Step1 : The Circuit



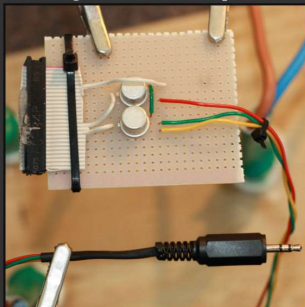
Click for Larger Version

The circuit comprises of two transistors which replace the physical buttons used on a standard handheld shutter release, a transistor is a switch after all - just one controlled by electricity instead. The transistors I used were "2N3440", I decided quite arbitrarily on these (Lots of Googling shows these are quite 'beefy' so not much chance of any power from the camera getting to my RPi) - I'm not an electronics person either, although I do have a soldering iron !

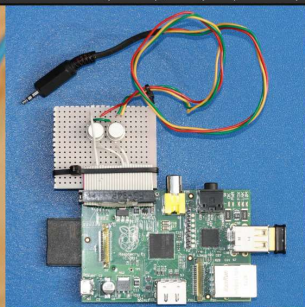
The basic logic is that each transistor has 3 'legs' Base, Collector & Emitter, they usually have a flat edge or a bump to show it's orientation - again, google it.

We apply current to the 'Base' from the RPi GPIO pin (via software which we'll get to later), which throws the transistor 'switch', the 'Emitter' is ground and the 'Collector' is connected to the respective pin in the 2.5mm jack. You can see all of the 'grounds' are connected together.

The pictures below show the finished board, and with it connected to the RPi. The 3 grey connector wires, with respect to the diagram above, from top to bottom, GPIO Pin#6(Green), #16(Red) & #18(Yellow)



Click for Larger Version



Click for Larger Version

That covers the hardware side of things, next is the software..

Sites I Like:

- [Diving Museum](#)
- [Historical Diving Society](#)
- [Raspberry Pi](#)

Blog Archive

- ▼ 2013 (1)
 - ▼ April (1)
 - [Raspberry Pi Controlled Canon DSLR](#)

About Me



[A.J. Bevan](#)

I am a 20+ year IT Professional with interests in a wide variety of subjects, primarily Astronomy, Computing, Photography, Physics, Networking & Food/Drink...

[View my complete profile](#)

Search This Blog

The software side of this project is surprisingly simple.

I used a Youtube video to understand how the GPIO is used and what software you need to install in order to write your own routines.

I'm not going to re-invent the wheel here so I suggest you watch it too : [GPIO For Beginners](#)

OK, so now you have the software installed and understand a bit more about GPIO.

To fire the camera shutter all we need to do is set the correct pin to 'high' for a specified amount of time and then bring it back down to 'low'

Below is the short script that I used, actually fires the camera for 4 seconds - you can of course change this to whatever you want. I have setup pin 23 as well (shutter half press), although I'm not using it in this script at the moment.

The code (shutter.py):

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)
GPIO.setup(24, GPIO.OUT)
GPIO.output(24, True)
time.sleep(04)
GPIO.output(24, False)
print "Done..."
```

Make it executable :

```
sudo chmod +X shutter.py
```

To actually run the code :

```
sudo python shutter.py
```

So you can see that by setting the pin number to 'True' we are in fact delivering current from the RPi GPIO pin to the 'Base' of the transistor which then fires the camera, then switch it off by setting the pin to 'False'.

This short video shows the whole setup working, this was my first YouTube video and the quality isn't great but it gives you the idea : [RPI GPIO Controlled Canon DSLR](#)

Next step is make the whole setup battery powered!!

Again, I'm not going to re-invent the wheel here, this site is where I got all the information needed :

[RPI Running From Batteries](#)

The only changes I made were getting a 10xAA Battery box and connecting a salvaged micro USB instead of using the GPIO header to connect the power. So far I have had my setup running for about 5 hours on 10xAAAs. In the end you then get :



[Click for Larger Version](#)

The webcam is used in the next part, detecting motion then activating the camera shutter!

We now have an isolated platform (associated with my wireless network) which can be placed in the field. Set up the tripod, camera etc approx 8-10 feet from the bird table, connect the 2.5mm jack to the remote shutter release port of the camera, SSH to RPi from the comfort of the conservatory and wait.....sometimes for a while.....until something lands on the table. When it is roughly in the area you have setup the camera for then run the script that fires the shutter, and repeat until your satisfied you have enough material to work through. When your finished you can get some good pictures.

Selection of pictures taken with this setup can be found here : [Recent Blackbird Pictures](#)

Next we want to fire the camera only when motion has been detected, so we don't have to sit there waiting, ie: automate the process.

Software we need for motion detection is called 'Motion', so first of all we need to install it, which is as simple as :

```
sudo apt-get install motion
```

Motion has its' own configuration file, 'motion.conf', we need to edit that to make a few changes, so....

```
sudo nano /etc/motion/motion.conf
```

Motion.conf is a long file, lines starting with a '#' are comment lines and are ignored. Listed next are the parameters I changed :

```
# Image width (pixels). Valid range: Camera dependent, default: 352
width 320

# Image height (pixels). Valid range: Camera dependent, default: 288
height 240
```



```
# Maximum number of frames to be captured per second.
# Valid range: 2-100. Default: 100 (almost no limit).
framerate 50

# Image File Output
output_normal off

# Live Webcam Server
webcam_port 8081

# Restrict webcam connections to localhost only (default: on)
webcam_localhost off

# Command to be executed when an event starts. (default: none)
# An event starts at first motion detected after a period of no motion defined
on_event_start sudo python shutter.py
```

So when we start 'motion' you can use your favorite browser to go to your RPi's IP address and port 8081 eg: <http://192.168.1.50:8081> to see what the webcam can see. Once motion is detected then the script shutter.py is called and fires the camera - that's it !!

The short video below shows the setup working, this was before the battery pack was completed:
[RPi Controlled Canon DSLR Motion Detect](#)

This does work 'in the field' but is a bit restrictive by the length of USB cable from the webcam to the RPi. To overcome this we need the webcam independent of our RPi running motion.
We'll use another program called 'mpeg-streamer', details and instructions of which are found here :
[RPi-Webcam_Streaming](#)

There is information regarding other programs in there but we are interested only in mpeg-streamer for this.

All we do then is edit motion.conf again and add the line

```
netcam_url http://192.168.1.64:8080/?action=stream
```

In the NetCam section of the file, amending the IP address according to whatever your second RPi's address is. I have had this working on the bench but no video available.

So that's it, you should now be able to get 'up close & personal' with whatever you want to get pictures of, without having massive lens or sitting around getting bored, hope it's useful.

Thanks go to people who I have linked to, it's thanks to their efforts that this work has been possible.
I'd also like to add my personal gratitude to Adam Kovacs for supporting me in this piece of work - Thanks Adam - Top Man !!

Posted by [A.J.Bevan](#) at 09:04



Labels: [canon](#), [DSLR](#), [GPIO](#), [rasberry pi](#), [RPi](#)

Location: [United Kingdom](#)

22 comments:



insomniak 23 April 2013 at 00:49

nice project, thank you, I will try this myself.
btw, you can upgrade your camera's firmware from current 1.0.3 to 1.0.7

[Reply](#)

▼ [Replies](#)



A.J.Bevan 23 April 2013 at 00:55

Thanks 'insomniak', I will look into firmware upgrade you mention.



Mario Henrique Cruz Torres 23 April 2013 at 08:43

Amazing shots Adrian.
Of course you can't get the flexibility you got with the RPi solution, but have you seen the MagicLantern (<http://www.magiclantern.fm/>) firmware hack for Canon Cameras? There are many pretty cool functions, even for controlling the camera via USB.

thanks for sharing.
mario h.c.t.



A.J.Bevan 23 April 2013 at 08:58

Thanks Mario, I'll have a look at Magic Lantern, but I must say my solution gives me what I need - at the moment. Thanks again for the link, it's appreciated.

[Reply](#)



What you make of it 23 April 2013 at 11:29

Excelent.. can't wait to get a moment to try this.

Thanks for sharing! People like you make the world a better place.

[Reply](#)



▼ Replies



A.J. Bevan 23 April 2013 at 12:32

Thank You Sir, very kind indeed...I wish you luck and if you need anything please ask.

Reply



Ian 26 April 2013 at 07:56

I have the bits to do this, but not quite got around to it yet:) I have built the hardware to trigger the shutter on my Canon. Now I just need to get the webcam setup. Did you put the webcam on the bird table, or on a tripod nearby?

Incidentally motion does work with ip cameras, but you need to do a bit of work to find the url of the mjpeg stream. I have got a Panasonic, which is part of my security system and use motion with it.

Reply



A.J. Bevan 26 April 2013 at 08:04

Hello Ian, the webcam was setup on a separate tripod with USB extension cables. Yes, Motion does work with IP cameras and I have had that working 'on the bench' with Pi #2, I need another wifi adaptor to get it off ethernet connection, so it can be placed anywhere.

Reply



Ornotermes 26 April 2013 at 13:05

If you wouldn't want to do it with GPIO, you should be able to control your camera over USB with piggyphoto (uses libgphoto2) as well. Depending on the camera you might be able to do more than just shoot pictures.

<https://github.com/alexdu/piggyphoto>
<http://www.gphoto.org/proj/libgphoto2/>

I made a simple python program to make a time-lapse with my Canon EOS 650D.
Video: http://www.youtube.com/watch?v=Am1P_fADJ90
Source: <http://pastebin.com/Xxiu3Rn7>

Reply



A.J. Bevan 26 April 2013 at 13:36

Hello Ornotermes, i did try Gphoto2 but had difficulties in the USB connection, including playing with the USB reset scripts that are available. In the end I went for a hardware solution and some good old fashion photography skills to make up the difference.

I do appreciate your contribution, links and your advice though, many thanks.

Adrian....

Reply



Guillaume 27 April 2013 at 12:59

There is a error in the schematic of the circuit : a mofset is drawn instead of a transistor. Have a look to the datasheet of the 2N3440 here : <http://www.datasheetcatalog.org/datasheet/stmicroelectronics/4080.pdf>

Regards

Reply



A.J. Bevan 27 April 2013 at 17:25

Thank you Guillaume, I think the basic objective was met in the diagram, but your correction is understood and will be observed in any future work.

Regards, Adrian...

Reply



Ian 8 May 2013 at 05:48

You should really place a shebang line at the start of your python script:

```
#!/usr/bin/python
```

Then you don't need to explicitly run the script with python, simply running the script itself will automatically invoke python. Also not having a shebang can sometimes cause weird errors with your scripts.

Reply



A.J. Bevan 8 May 2013 at 06:42

Hello Ian, many thanks indeed, very useful information. I will also add "automagically" to my lexicon - super!

Adrian..

Reply



Garry Smith 19 May 2013 at 10:56

Hi Great project.

I have set this up just with one Rpi and a web cam attached and everything is working fine, except after the motion software triggers the camera, the picture is taken, but the shutterpy program is still showing done but does not exit to the motion software. I'm new to python, so do need to something to the shutter.py, or is their something in the motion.conf what needs setting up after the shutter.py has been executed. Also if I stop the motion software and run it again and motion is detected the shutterpy runs, but complains the GPIO channels are still in use and nothing happens, just gets stuck at the print "done", with no picture taken. I have to reboot to clear the channels. Any idea how to solve.

Thanks Garry

Reply



Replies



A.J. Bevan 20 May 2013 at 04:03

Hi Gary, once shutter.py is 'Done' there is nothing else to do, the script finishes. I suspect that you are expecting motion to fire the camera again immediately if motion is detected, if so then you need to change a variable in motion.conf. Look for "Gap", this is a numerical value expressed in seconds, and effectively waits 'x' number of seconds after an event before it starts to look for more motion events, default is 60 seconds. Config options here : <http://www.lavrsen.dk/foswiki/bin/view/Motion/ConfigFileOptions>



Garry Smith 20 May 2013 at 10:36

Thanks for your reply, yes you are correct in me expecting the camera to fire again. I would like to take multiple shots, then have the motion to wait for the next occurrence of motion. One idea is to get motion to restart after it has taken shots of the subject. I will take a look at the Gap in the conf file.

Thanks again.

Garry

Thanks



A.J. Bevan 20 May 2013 at 12:00

Garry, it will fire again, after the 'GAP' timeout. This is where a bit of photography comes in as well. I set mine to continuous shooting and shutter.py set to fire for 3-4 seconds, so you get multiple shots. Add that time and a time for the camera to store the pictures and be ready for the next lot, so you then create a 'GAP' time that suits. To reiterate once shutter.py is finished there is no procedure necessary to return control to 'Motion' it's automatic, there is no restarting.

Reply



andycrofts 9 October 2013 at 02:04

Hi, Adrian.

One small thing, (as an electronics engineer) you are missing current-limiting resistors between the base of the transistors, and the associated GPIO pin. This means the Pi is under serious danger of overcurrent - when fired, the pin's effectively shorted to ground, or nearly so. Stick a resistor between each GPIO pin and the associated base of the transistor, and you'll be fine. Anything from about 4.7K to 22K should be fine - whatever you can find. (The batteries might last a fraction longer, too!)

Reply

Replies



A.J. Bevan 14 October 2013 at 06:09

Hi Andy,

Many thanks for dropping by and taking the time to comment. I will in future add the resistors as you suggest. My reason for not doing in the first place was I don't know how to work out what a reasonable Ohm to use. My reading suggested I would be safe enough in not using any, but I accept that they probably should be there for belt and braces.

Regards.

Reply



Ernie King 10 October 2013 at 19:26

I enjoy since websites that understand the value of providing a prime supply for free. I truly loved analysis your post.
<http://buyadslrcamera.org/>

Reply

Replies



A.J. Bevan 14 October 2013 at 06:11

Very kind of you to say so Ernie, many thanks.

Reply



INTERNET ARCHIVE

DONATE

WayBackMachine

Sorry

You have already reached the limit of active Save Page Now sessions. Please wait for a minute and then try again.

[Return to Save Page Now](#)

[Home](#)

Subscribe to: [Post Comments \(Atom\)](#)

Total Pageviews





<https://web.archive.org/web/20150228100916/http://www.raspberrypi.org/products/raspberry-pi-2-model-b/>





RASPBERRY PI 2 MODEL B

The Raspberry Pi 2 Model B is the second generation Raspberry Pi. It replaced the original **Raspberry Pi 1 Model B+** in February 2015. Compared to the Raspberry Pi 1 it has:

- A 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM

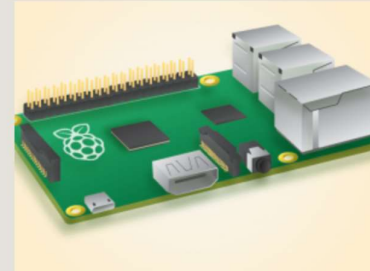
Like the (Pi 1) Model B+, it also has:

- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

Because it has an ARMv7 processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10 (see the blog for more information).

The Raspberry Pi 2 has an identical form factor to the previous (Pi 1) Model B+ and has complete compatibility with Raspberry Pi 1.

We recommend the Raspberry Pi 2 Model B for use in schools: it offers more flexibility for learners than the leaner (Pi 1) **Model A+**, which is more useful for embedded projects and projects which require very low power.



BUY FROM OUR DISTRIBUTORS



[« RETURN TO PRODUCTS](#)



About us
Trademark rules

FAQs
Contact us

Cookies
Creative Commons

RASPBERRY PI FOUNDATION
UK REGISTERED CHARITY 1129409



JURAT ATTACHMENT

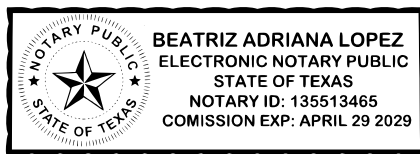
A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

STATE OF Texas }

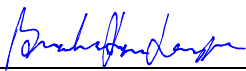
COUNTY OF Harris }

The foregoing instrument was subscribed and sworn before me this date of
11/04/2025 by Mina Ching

This notarial act was an online notarization.



(Notary Seal)
Document Notarized using a Live Audio-Video Connection

Notary's Signature 

Registration No.: 135513465

Commission Expiration Date: April 29, 2029

