

Home Security Projects for Raspberry Pi



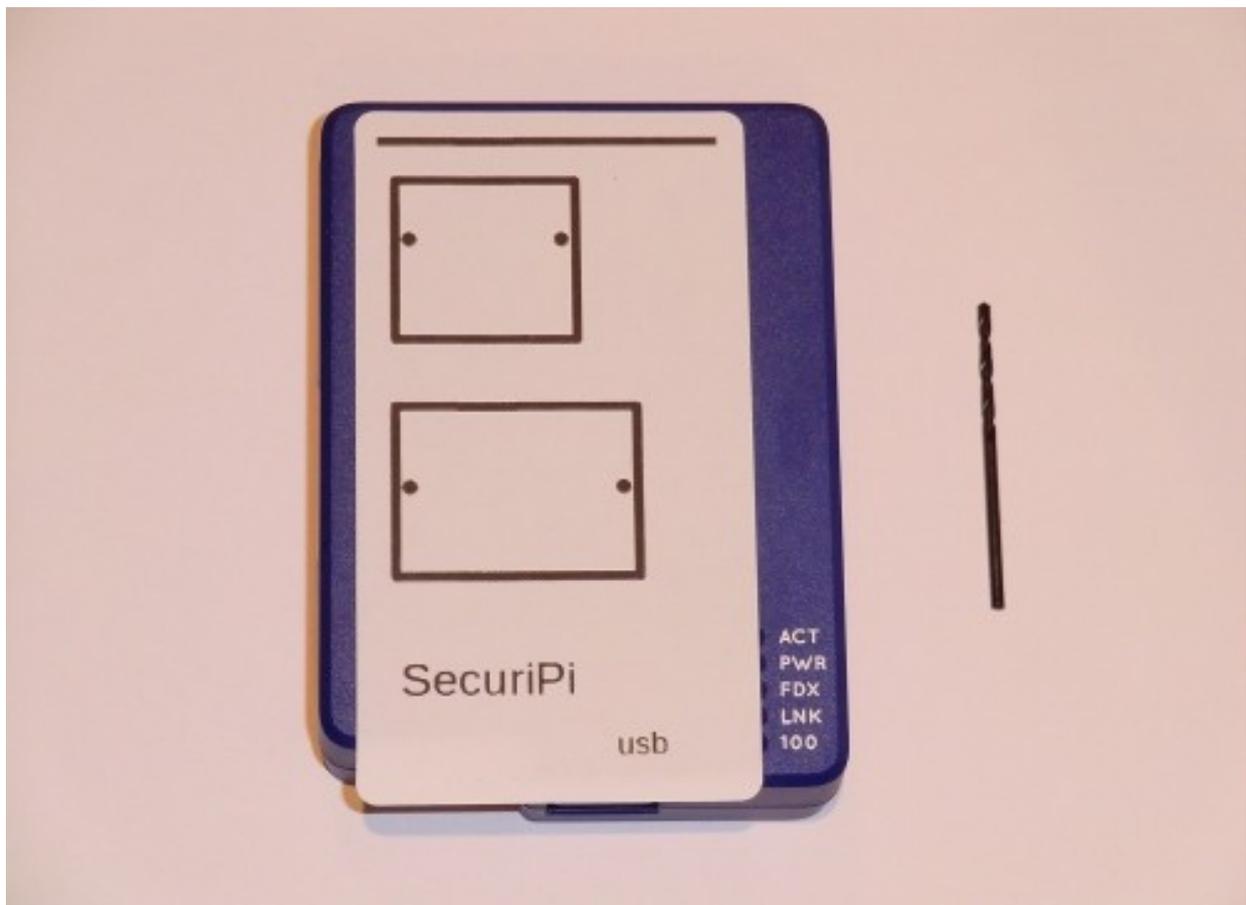
Tim Rustige

Add a PIR sensor to your Raspberry Pi.

Components:

- Mini PIR Sensor.
- Three IDC connection wires.
- M2 Nylon plastic nuts & 25mm bolts.
- M2 drill bit.
- M2 Nylon stand-offs.
- Drill template sticker.
- DVD with instructions and scripts.

Remove your Pi from the case before starting. Place the drill template over the front of your Pi, like in the photo below.



You need to drill four holes using the supplied 2mm drill bit. If you want to move the positions of the camera & PIR to suit your own case, then just cut out the squares and place them where you want. Make sure the ribbon cable from the camera module will reach, with the cables inserted the correct way around at both ends. To connect our camera module through this particular type of case we broke off the plastic near the power supply connector. If you have a Dremel mini circular saw you could cut a single neat line in the case for the camera's CSI cable to pass through.

The PIR sensor has three pin connections on the base, so you either need to drill a hole about 1cm in diameter through the case or bend the pins out very carefully at ninety degrees.

When you've finished your Pi should look like this:



It doesn't matter if the Pi camera is physically mounted upside-down, as the image can be flipped horizontal or vertical with a command line switch of `-hf` or `-vf`. The only important thing is that the CSI camera cable reaches without too much tension on it. The positioning shown in the photo worked fine for us.

It's simple to strap the Pi to a shelf using some plastic ties, which we supply.

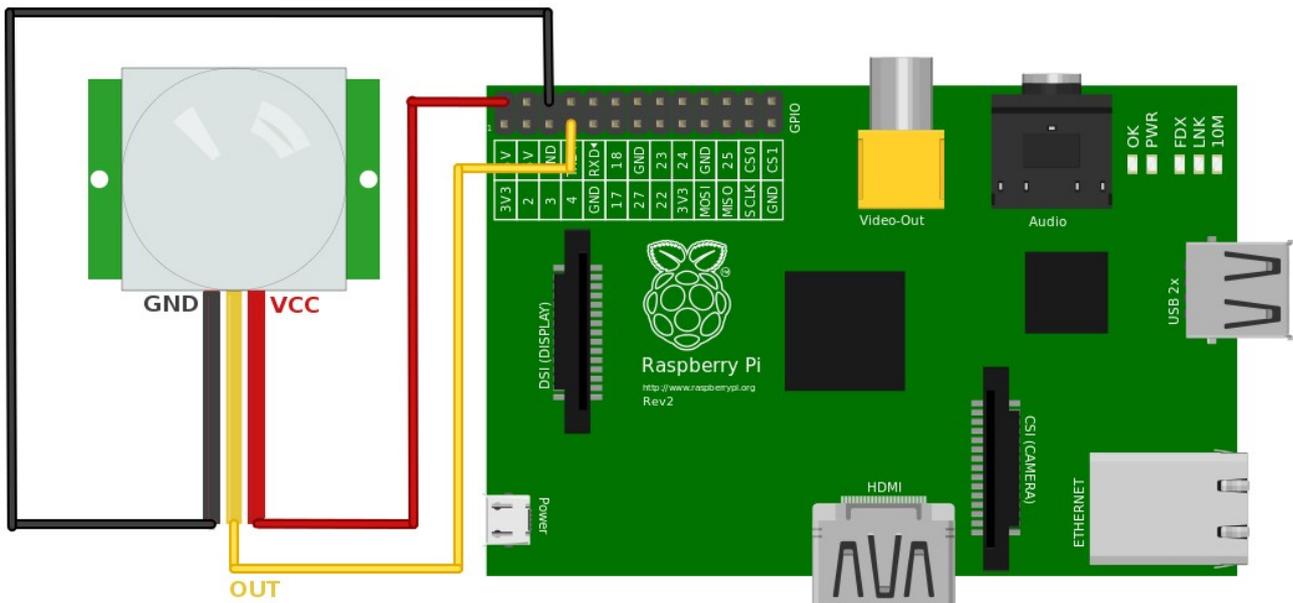
Or you could mount this model of Pi case on a wall using the screw mounting lugs on the rear.

If you want an exterior mounted camera, it would be equally possible to use a dummy CCTV camera housing, which will typically cost around £10, and drill out a large hole for the PIR dome.

If you need to run the Pi in a remote building, without power, you could rig the Pi up with a battery pack & WiFi or USB 3G Modem.



How to connect up the PIR module.



Made with  Fritzing.org

The wires we supply may vary in colour from those shown above, but that isn't important, only the points that are joined together matter – VCC to 5V, OUT to GPIO4 and GND to GND. Pin labels are screen printed on the PIR's circuit board. The two trimmers on the top of the PIR allow you to change the sensitivity & dwell time of the PIR, leave them set at defaults for now.

When the PIR unit detects motion GPIO pin 4 reads 1, otherwise it reads 0. We can use that in a script to take a photo & record video from the Raspberry Pi camera, which can then be emailed to a phone.

Download the scripts from our website.

While it's possible for you to manually type in each script, or transfer them from the DVD, you might find it easier to download them from our website. On your Pi, type in the following at the command line:

```
wget http://securipi.co.uk/securipi.zip
```

```
tar xvf securipi.zip
```

```
chmod a+x *.sh
```

```
ls -al
```

You should now see various scripts listed. To run the test script type:

```
sudo ./test.sh
```

How to make an executable script.

We've assumed you're running the latest version of the Raspbian OS for the Raspberry Pi. (There's an ISO image of it on the DVD and instructions for making a bootable SD card at the end of this document).

When you see us change font to `courier`, we are indicating a command you type into the command line. For example:

```
nano test2.sh
```

Will open a basic text editor & allow you to enter a series of commands, that form the basis of all our scripts. Enter the following text:

```
echo "hello world"
```

Then save the file & exit. To make the command executable type:

```
chmod u+x test2.sh
```

and then to run the script

```
./test2.sh
```

Some scripts will need to be run as root (the highest security level). To do that type:

```
sudo ./test2.sh
```

To see a list of files in the current folder type

```
ls -al
```

If you have any problem running a script, then you've either made a typo, forgotten to make it executable with `chmod`, or need to put a `sudo` in front of it.

If you get bored of entering `sudo` before each command, you can switch to root by entering:

```
sudo su
```

You'll notice the prompt then changes from `$` to `#`. You can exit root by pressing Ctrl-D. While you're in root mode it's a good idea to change the default passwords for root & user pi, like this:

```
passwd  
passwd pi
```

Run the test script

Once you have everything connected together correctly, the first step is to run the test script, which scrolls “1”s or “0”s up the display, depending on the state of the PIR sensor.

```
nano test.sh
```

Then enter the following commands into the nano editor:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
  trap 'echo "4" > /sys/class/gpio/unexport' 0
  stat=`cat /sys/class/gpio/gpio4/value`
  echo $stat
done
exit 0
```

Save the file & quit the editor. Next, make the script executable:

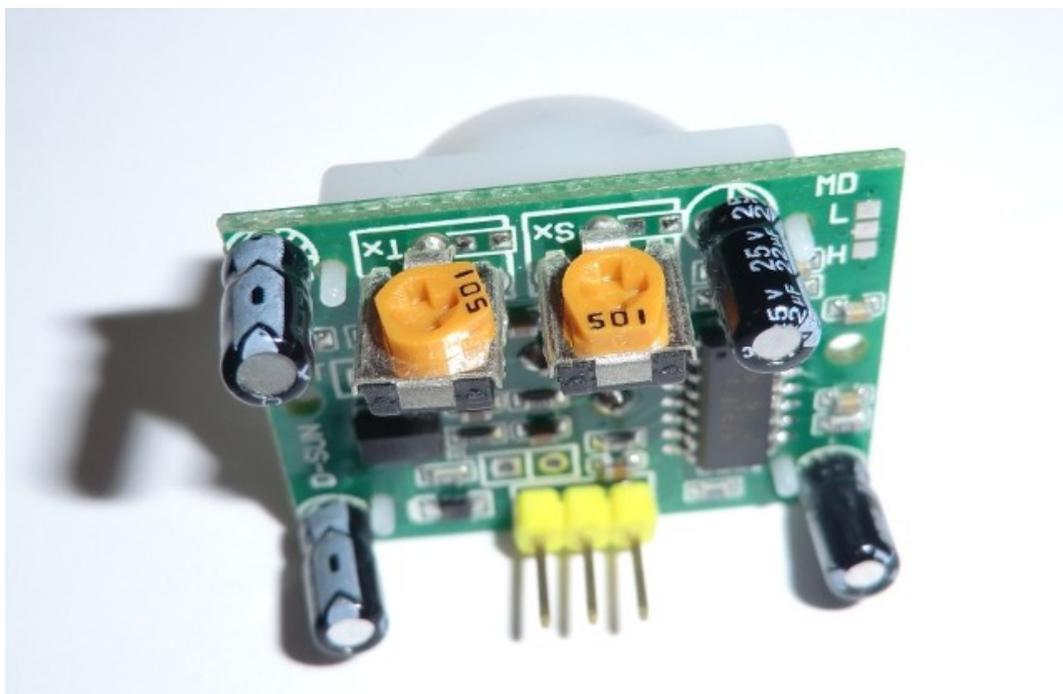
```
chmod u+x test.sh
```

Now run the script with:

```
sudo ./test.sh
```

You should now see “0”s scroll up the screen, when the PIR doesn't detect any motion. Move in front of the PIR and you should see “1”s scroll up the screen.

This would be a good time to adjust the two trimmers on the PIR to suit your detection range & hold time needs, if you need to. Sx controls detection distance & the default is normally set to 50% & Tx controls the time the output stays at '1' when read by the Pi (default is a few seconds), between 3-200 seconds.



Send an email when the PIR detects motion (updated for Raspbian Buster).

This script builds on the last one, by adding email & logging ability.

```
nano securipi.sh
```

Then enter the following commands into the nano editor:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
  trap 'echo "4" > /sys/class/gpio/unexport' 0
  stat=`cat /sys/class/gpio/gpio4/value`
  while [ $stat = "1" ]
  do
    d=`date +%d%m%y`
    t=`date +%T`
    echo -e "Motion Detected $t $d" | msmtplib your-email-address@your-domain.co.uk
    echo "Motion Detected $t $d"
    echo "Motion Detected $t" >> log$d.txt
    stat='0'
  done
done
exit 0
```

Save the file & exit.

Once again, make the script executable with

```
chmod u+x securipi.sh
```

Before we can run the script we need to install & configure several mail applications from the internet. You also need to have made an **extra Gmail account** (using your PC) for just the Pi to use, and configured it to “allow less secure apps” to use it.

```
sudo apt-get install msmtplib mailutils mpack ca-certificates
```

Setup default settings for MSMTPLIB.

```
cd /etc
sudo touch msmtplib
sudo nano msmtplib
```

```
account default
host smtp.gmail.com
port 587
logfile /tmp/msmtplib.log
tls on
tls_starttls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
auth login
user your-pi-new-gmail-account@gmail.com
password yourpassword
from First Last Name
account account2
```

save file & exit.

```
cd ~
```

To go back to your home folder.

Send a test email with this command, substituting the email address below with your own.

```
echo -e "Subject: Test Mail\r\n\r\nThis is a test mail" |msmtp --  
debug --from=default -t youremail@yourdomain.co.uk
```

That goes all on one line. If it worked you can remove the --debug option next time.

You can also use it like this

```
echo -e "Motion Detected" | msmtp your-email-address@your-  
domain.co.uk
```

Assuming that worked okay, you can now run the securipi.sh script.

```
sudo ./securipi.sh
```

When you move in front of the PIR you should see “Motion Detected” appear on the display, a record will also be added to the date stamped text file. Now check your emails & you should have a record of the PIR being triggered. You can quit the securipi.sh script with Ctrl-C.

You can check the contents of a time stamped log file with:

```
cat log060613.txt
```

You can see a list of all your log files with:

```
ls -al log*.txt
```

If you have lots of entries in your log file, you can pause the display with:

```
cat log060613.txt | more
```

or you can open the file in the editor with:

```
nano log060613.txt
```

If you notice the time & date aren't set correctly, you can run:

```
sudo dpkg-reconfigure tzdata
```

Setup the Raspberry Pi camera module.

Install latest version of Raspbian to an SD card.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo raspi-config
```

Enable camera support in raspi-config & reboot.

Check the camera's working with these two test commands:

```
raspistill -o image.jpg
```

```
raspivid -o video.h264 -t 10000
```

Install software needed to send emails and attachments. (Setup a spare account @ gmail.com and log into it using the web browser on your PC at least once a month, to keep it active.)

```
sudo apt-get install ssmtp mailutils mpack
```

Setup default settings SSMTP.

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Enter this into the editor:

```
AuthUser=your-gmail-account@gmail.com  
AuthPass=your-user-password  
FromLineOverride=YES  
mailhub=smtp.gmail.com:587  
UseSTARTTLS=YES
```

save file & exit.

Send a test text email and then an attachment.

```
echo "testing 1 2 3" | mail -s "Subject" you@yourdomain.co.uk
```

```
mpack -s "alarm photo" /home/pi/image.jpg you@yourdomain.co.uk
```

The standard photo resolution was too big to view on our mobile phone screen, so in the script below we've reduced the size to something easier to view on a small screen. The video that gets recorded to your SD card is still HD.

The video files captured by the script are typically too large to send as email attachments. If you want to view them remotely, it's best to login to the Pi using Filezilla on a PC in SFTP mode.

```
nano securipi-rpicam.sh
```

Enter the script:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
  trap 'echo "4" > /sys/class/gpio/unexport' 0
  stat=`cat /sys/class/gpio/gpio4/value`
  while [ $stat = "1" ]
  do
    d=`date +%d%m%y`
    t=`date +%T`
    raspistill -o $t$d.jpg -w 1024 -h 768 -q 30 -hf
    echo "Movement Detected $t $d" | mail -s "Movement Detected" you@gmail.com
    echo "Movement Detected $t $d"
    echo "Movement Detected $t" >> log$d.txt
    raspivid -o $t$d.h264 -t 10000
    mpack -s "Movement Detected photo" $t$d.jpg you@gmail.com

    stat="0"
    sleep 20
  done
done
exit 0
```

Save the file (Ctrl-O) & exit (Ctrl-X).

```
chmod u+x securipi-rpicam.sh
```

```
sudo ./securipi-rpicam.sh
```

The script takes a photo & video when the PIR is triggered & sends the photo as an attachment to the Gmail account specified in the script. It then records 10 seconds of HD quality video to the memory card. The time of the event is also recorded to a date-stamped log file.

Here's an analysis of what happens on the raspistill command line:

```
raspistill -o $t$d.jpg -w 1024 -h 768 -q 30 -hf
```

Output a still photo to a file whose name is the current time+date.jpg. Make the photo measure 1024 pixels wide (-w) by 768 pixels high (-h). Set the quality (-q) to 30, which is quite low but reduces the size of file we then send by email. Finally, horizontal flip the picture (-hf) because our test picture was the wrong way around. If the camera was mounted upside down we could include the vertical flip (-vf) switch to correct for that with both the raspistill & raspivid commands.

Text on photo.

How to overlay text on a photo from the Raspberry Pi Camera module.

You might want to overlay text on a photo to provide a time & date stamp, before emailing the photo to your phone. Firstly, we need to install the Imagemagick library:

```
sudo apt-get install imagemagick
```

Next, we need to take a test photo & then use the `convert` command to overlay the text.

```
raspistill -o 1.jpg -w 1024 -h 768 -q 30
```

```
d=`date +%d%m%y`
```

```
t=`date +%T`
```

```
convert -pointsize 20 -fill yellow -draw 'text 850,30 "'$t'  
'$d''"' 1.jpg 2.jpg
```

The top-left of the photo is co-ordinate 0,0 and the bottom-right would be 1024,768. We positioned



the time & date at co-ordinate 850,30. The file 2.jpg is now time & date stamped.

If you notice the time & date aren't set correctly, you can run:

```
sudo dpkg-reconfigure tzdata
```

The script below will email you a time & date stamped photo:

```
nano securipi-rpicamtd.sh
```

Enter the script:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
    trap 'echo "4" > /sys/class/gpio/unexport' 0
    stat=`cat /sys/class/gpio/gpio4/value`
while [ $stat = "1" ]
do
    d=`date +%d%m%y`
    t=`date +%T`
    raspistill -o 1.jpg -w 1024 -h 768 -q 30
    raspivid -o $d$t.h264 -t 10000
    convert -pointsize 20 -fill yellow -draw 'text 850,30 "'$t' '$d'' 1.jpg $d$t.jpg
    mpack -s "alarm photo" /home/pi/$d$t.jpg you@youreemailaddress.co.uk
    stat='0'
done
done
exit 0
```

Save the file (Ctrl-O) & exit (Ctrl-X).

```
chmod u+x securipi-rpicamtd.sh
```

```
sudo ./securipi-rpicamtd.sh
```

You'll also find the script on our DVD.

Using a Sony PS2 or PS3 Eye Toy USB webcam to take photos.



Install software needed to send emails and attachments. (Setup a spare account @ gmail.com and log into it using the web browser on your PC at least once a month, to keep it active.)

```
sudo apt-get install ssmtp  
sudo apt-get install mailutils  
sudo apt-get install mpack
```

Setup default settings SSMTP.

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Enter this into the editor:

```
AuthUser=your-gmail-account@gmail.com  
AuthPass=your-user-password  
FromLineOverride=YES  
mailhub=smtp.gmail.com:587  
UseSTARTTLS=YES
```

save file & exit.

Send a test text email and then an attachment.

```
echo "testing 1 2 3" | mail -s "Subject" you@yourdomain.co.uk
```

```
sudo apt-get install fswebcam
```

```
sudo fswebcam -d /dev/video0 -r 320x240 test1.jpg
```

```
sudo fswebcam -d /dev/video0 -r 640x480 test2.jpg
```

```
mpack -s "alarm photo" test1.jpg youremailaddress@gmail.com
```

Script to grab photo from webcam when PIR is triggered.

```
nano securipi-webcam.sh
```

Then enter:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
  trap 'echo "4" > /sys/class/gpio/unexport' 0
  stat=`cat /sys/class/gpio/gpio4/value`
  while [ $stat = "1" ]
  do
    d=`date +%d%m%y`
    t=`date +%T`
    fswebcam -d /dev/video0 -r 640x480 $t$d.jpg
    echo "Motion Detected $t $d" | mail -s "Motion Detect" you@gmail.com
    echo "Motion Detected $t $d"
    echo "Motion Detected $t" >> log$d.txt
    mpack -s "Motion Detected photo" $t$d.jpg you@gmail.com
    stat='0'
  done
done
exit 0
```

Save & exit.

```
chmod u+x securipi-webcam.sh
```

```
sudo ./securipi-webcam.sh
```

Script to capture photos from Easycap USB video grabber.

When the PIR detects motion, the script below will capture a series of photos from a Syntek 1160 chipset USB video grabber and save the results to the SD card. It then works out which are failed captures (under 20k in size) & deletes them. You need to use the very latest version of Raspbian to capture any good frames from the USB video grabber at all. On average, only half of the six captures will be good (increase the line `counter=6` to suit).

We still consider support for USB video grabbers on Raspberry Pi to be experimental only. This script emails the last successful capture.

Script to grab photo from Easycap USB video grabber STK1160 chipset when PIR is triggered.

```
nano securipi-easycap.sh
```

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
    trap 'echo "4" > /sys/class/gpio/unexport' 0
    stat=`cat /sys/class/gpio/gpio4/value`
    while [ $stat = "1" ]
    do
        d=`date +%d%m%y`
        t=`date +%T`

        # start code for usb video grabber
        counter=6
        while [ $counter -gt 1 ]
        do

            d=`date +%d%m%y`
            t=`date +%T`
            fswebcam -d /dev/video0 -i 0 -r 720x576 $d$t.jpg
            sleep 1
            counter=$(( $counter - 1 ))
            echo $counter

        done
        ls -l *.jpg | awk '{if ($5 < 20000) print $9}' | tee -a deletelog | xargs rm
        rm deletelog
        # end code for USB video grabber

        echo "Motion Detected $t $d" | mail -s "Motion Detect" you@gmail.com
        echo "Motion Detected $t $d"
        echo "Motion Detected $t" >> log$d.txt
        p=`ls *.jpg -Art | tail -n 1`
        mpack -s "Motion Detected photo" $p you@gmail.com
        stat='0'
    done
done
exit 0
```

Save the file (Ctrl-O) & exit (Ctrl-X).

```
chmod u+x securipi-easycap.sh
sudo ./securipi-easycap.sh
```

Bluetooth scanning.

The Bluetooth adapter we used is just a cheap thumbsize unit with a CSR chip inside. To install the Bluetooth stack for Raspbian on the Pi, at the Terminal prompt type:

```
sudo apt-get install bluetooth bluez-utils blueman
```

insert the Bluetooth adapter & scan for devices by typing

```
hcitool scan
```

if it doesn't work, then use

```
hciconfig hci0 up
```

to bring the interface up & then scan again.

The script below will scan for Bluetooth devices when the GPIO4 pin reads 1 (it normally reads 0, until someone trips the PIR).

Here's the shell script:

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
  trap 'echo "4" > /sys/class/gpio/unexport' 0
  stat=`cat /sys/class/gpio/gpio4/value`
while [ $stat = "1" ]
do
  x=`hcitool scan --flush`
  y=${x#*Scanning *...}
  d=`date +%d%m%y`
  t=`date +%T`

  echo $d,$t,$y | tr " " "\n"
  echo "-----"
  echo $d,$t,$y | tr " " ", " >> aa$d.txt
  y=''
  stat='0'
done
done
exit 0
```

Set up Wifi Sniffing on Raspbian build.

```
Sudo apt-get update
sudo apt-get upgrade
sudo apt-get install aircrack-ng
```

```
sudo airodump-ng-oui-update
```

```
sudo airmon-ng start wlan0
sudo tshark -i mon0 subtype probereq
```

Some other commands to try:

```
sudo tshark -i mon0 subtype probereq -w /tmp/rpi-cap.pcap
```

```
sudo airodump-ng mon0
```

Scan for WiFi Probe Requests.

The WiFi chip in a phone uses the same unique Mac style address (00:11:22:AB:CD:EF) as a Bluetooth chipset, so it's possible to record a unique phone identifier. You can see the unique Bluetooth & WiFi Mac addresses of your own phone under the 'Settings → About This Phone' menu.

When WiFi is turned on in my phone it sends probe requests every ten seconds. If I've previously associated successfully with other WiFi networks, then this information is also available, and may give you clues to who they are. This project works fine with the £10 WiPi USB wifi adapter available from cpc.co.uk

If you have the USB WiFi dongle connected you can use the `ifconfig` command to show information.

You should expect to see the adapter listed as `wlan0`. We now need to place the WiFi adapter into Monitor mode.

```
sudo airmon-ng stop wlan0
```

followed by the command

```
sudo airmon-ng start wlan0
```

should produce the `mon0` interface, you can do

```
sudo airodump-ng mon0
```

and see WiFi access points near you. CTRL-C to quit.

To see full probe requests from devices with WiFi enabled do

```
sudo tshark -i mon0 subtype probereq
```

This shows you the manufacturer of the device sending the probe, but it's also possible to just have the complete Mac address without the name resolution. This command records for 60 seconds to a log file:

```
sudo tshark -i mon0 -n -a duration:60 subtype probereq > cap.log
```

you can view the contents of the log file with

```
cat cap.log | more
```

We use the Tshark command in our script to grab the probe requests for 60 seconds, when the PIR alarm is triggered. The script also scans for Bluetooth devices. The script then processes the capture from Tshark to remove any duplicates and just leaves the unique Mac addresses spotted in the final text that are sent in the email to our Gmail account.

```
nano securipi-wifi.sh
```

```
echo "4" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio4/direction

while true; do
    trap 'echo "4" > /sys/class/gpio/unexport' 0
    stat=`cat /sys/class/gpio/gpio4/value`
while [ $stat = "1" ]
do

        x=`hcitool scan --flush`
        y=${x##*Scanning *}...}
        tshark -i mon0 -n -a duration:60 subtype probereq > cap.log
        egrep -o "[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}:[a-z0-9]{2}" cap.log > cap2.txt
        sed '/ff:ff:ff:ff:ff:ff/d' cap2.txt > cap3.txt
        sort -u -o cap4.txt cap3.txt
        z=`cat cap4.txt`
        d=`date +%d%m%y`
        t=`date +%T`

        echo $d,$t,$y,$z | tr " " "\n"
        echo "-----"
        echo $d,$t,$y,$z | tr " " ", " >> aa$d.txt
        echo -e "Subject: PIR Alert\r\n\r\n ALERT $t,$y,$z" |msmtp -from=default -t
you@youremailaddress.com

        y=''
        z=''
        stat='0'

done
done
exit 0
```

Then make the file executable with

```
chmod ugo+x securipi-wifi.sh
```

Run the scanner command with

```
sudo ./securipi-wifi.sh
```

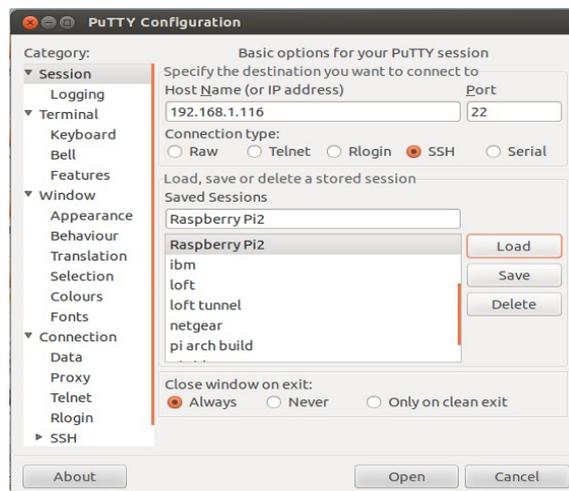
Connect to your Pi remotely from your PC.

It's a pain leaving a keyboard, mouse & monitor connected to your Pi when running the PIR alarm, so we login remotely using the free Putty terminal emulation package. Putty is available for PC, Mac & Linux from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> This gives you a remote text terminal window on your PC, where you can issue commands as if you were sat in front of the Pi.

When you first install Raspbian Wheezy onto an SD card & start your Pi, you'll notice the `raspi-config` command will give you the option to turn on SSH – this is the service we use to login remotely. When you boot up the Pi, you'll also see the IP Address allocated to your Pi, above the Login & Password prompt – it will look something like 192.168.1.149 ← *note down the address you see and enter it into the Putty software on your PC.*

If you don't see the ip address you can use the command:

```
sudo ifconfig
```



On the PC running Putty, make sure the IP address is entered, you are set to Port 22 & the SSH radio button is selected, choose Open. Login as `pi` & password `raspberrypi`. Once logged in use `passwd` to change the easily guessed default to something else. We suggest you also do `sudo su` and `passwd` again, to change the root password, then do CTRL-D to drop back to user Pi.

You can make the scripts you've created run in background on the Pi & then logout from Putty. You do this by adding an ampersand character '&' to the end. To run `securipicamtd.sh` in background type:

```
sudo ./securipicamtd.sh &
```

Make a note of the process number displayed (say 2501 for example below) & then logout using CTRL-D. You should now receive emails from your Pi every time the PIR alarm is tripped.

When you log back in you can kill the background process by rebooting the Pi, or by doing:

```
sudo kill -9 2501
```

How to transfer files from your Pi to PC using Filezilla.

On your Raspberry Pi type in

```
ifconfig
```

and make a note of your inet addr of eth0 interface, it will be something like 192.168.1.135

On your PC, Mac or Linux PC download & install Filezilla from <https://filezilla-project.org/download.php>

Launch Filezilla & go to File → Site Manager → New Site → name it Raspberry Pi.

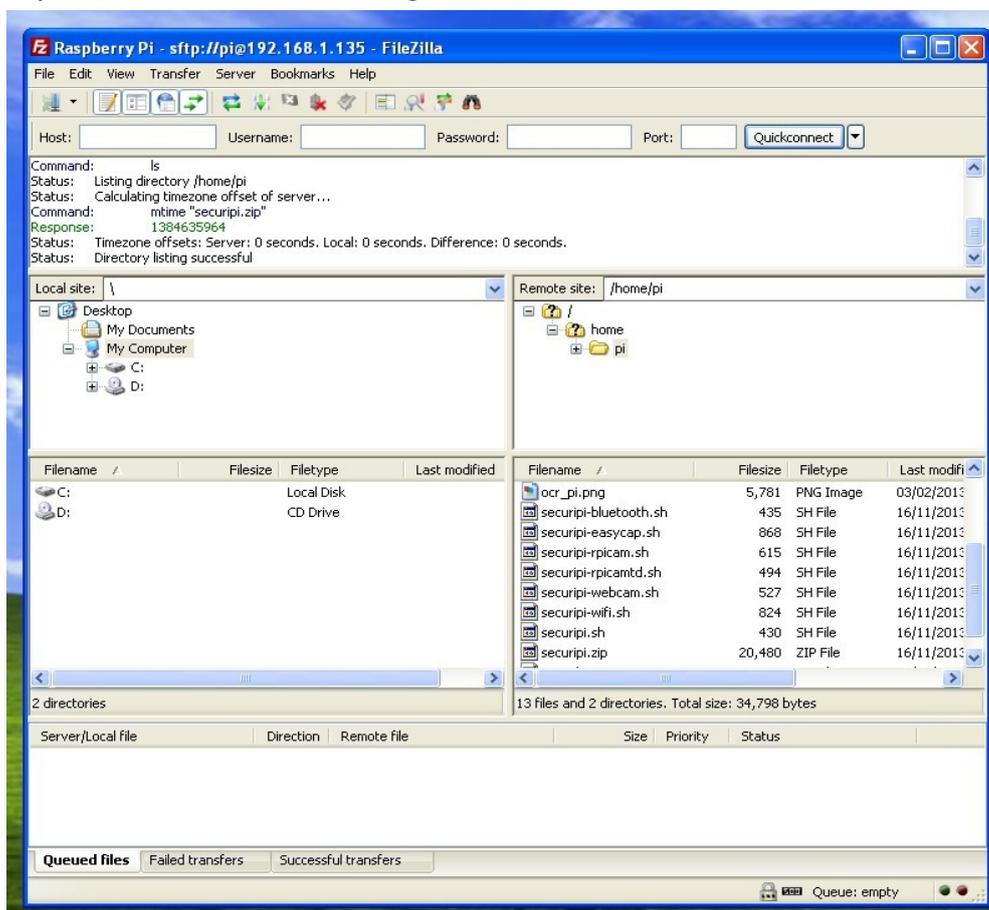
Then enter the following, remembering your IP address won't be the same as mine:

```
Host      : 192.168.1.135
Port      : 22
Protocol  : SFTP
Logon Type : Normal
Login     : pi
Password  : raspberry
```

Then click OK.

File → Site Manager → Raspberry Pi → Connect → tick always trust this host tickbox & OK. List of files on the Pi appears in the right hand pane. You can drag videos captured on your Pi onto your PC's Desktop (left hand pane).

Once you have the H.264 video files on your PC, you can play them using the free VLC Media Player. <http://www.videolan.org/vlc/index.html>



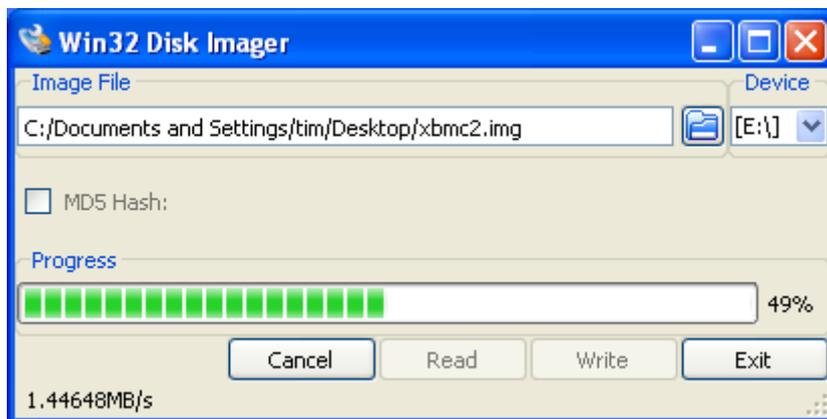
How to load Raspbian OS onto an SD card

The PIR alarm GPIO kit DVD also contains the latest Raspbian OS for the Raspberry Pi. The image needs to be unzip'd & loaded onto a blank SD card in the correct manner, using your Windows or Linux PC. You'll need to use Raspbian version 2013-09-25-wheezy-raspbian.img (or newer) if you have the Raspberry Pi camera module or a USB Video Grabber.

The Raspberry Pi can only boot from an SD card, not a USB stick or external hard drive. We recommend you use a fast Class 10 SD card, although Raspbian will also run fine on an older Class 4 card.

For Windows Users: Insert the SD card into your PC and make a note of the drive letter assigned to it – in our case it was the E: drive. Extract the Win32diskmanager-binary.zip software from the DVD onto your Desktop, and you'll see an icon like the one below appear. Then drag over to your Desktop & Unzip any of the software images you want to use. Run Win32DiskManager, choose the image file (2013-09-25-wheezy-raspbian.img) & upload to SD card, making sure you selected the SD card as the target Device and not your hard drive – this will then make an SD card that the Raspberry Pi can boot from.

We've also included HP's USB/SD card formatter called SP27608-2.1.8.exe which can be useful for blanking pre-used SD cards & USB sticks.



For Linux/Ubuntu Users: drag the image files from the DVD onto your Desktop, extract the images from the zip files. Have target SD card attached to PC. Open a Terminal by holding down keys CTRL-ALT-T together. Then type:

```
sudo apt-get install gparted
sudo gparted
```

Now check in Gparted that your SD card appears as drive letter SDB (top right change SDA to SDB). This is very important, if you have more than one hard drive your SD card could appear as SDC or HDC, in which case amend instruction below to reflect the difference. Assuming your SD card is SDB & you want to upload the Raspbian image, type this into the Terminal:

```
sudo dd if=2013-09-25-wheezy-raspbian.img of=/dev/sdb
```

Hit Enter & wait around 30 minutes while the image is copied to the SD card, you'll know it's finished when the \$ prompt re-appears. The Raspbian image gives you an options menu when you first run it in the Raspberry Pi, and you should choose to have Raspbian resize itself to the full capacity of the SD card.

Need more help? http://elinux.org/RPi_Easy_SD_Card_Setup