

Using USB camera

Using USB camera

[Install FSWebcam](#)

[View USB camera device](#)

[Photograph](#)

[Time-lapse photography](#)

[Using Cron \(scheduled tasks\)](#)

Take photos and videos on your Raspberry Pi using a standard USB camera.

Install FSWebcam

FSWebcam is a simple and clear webcam application. The software installation command is as follows:

```
sudo apt install fswebcam
```

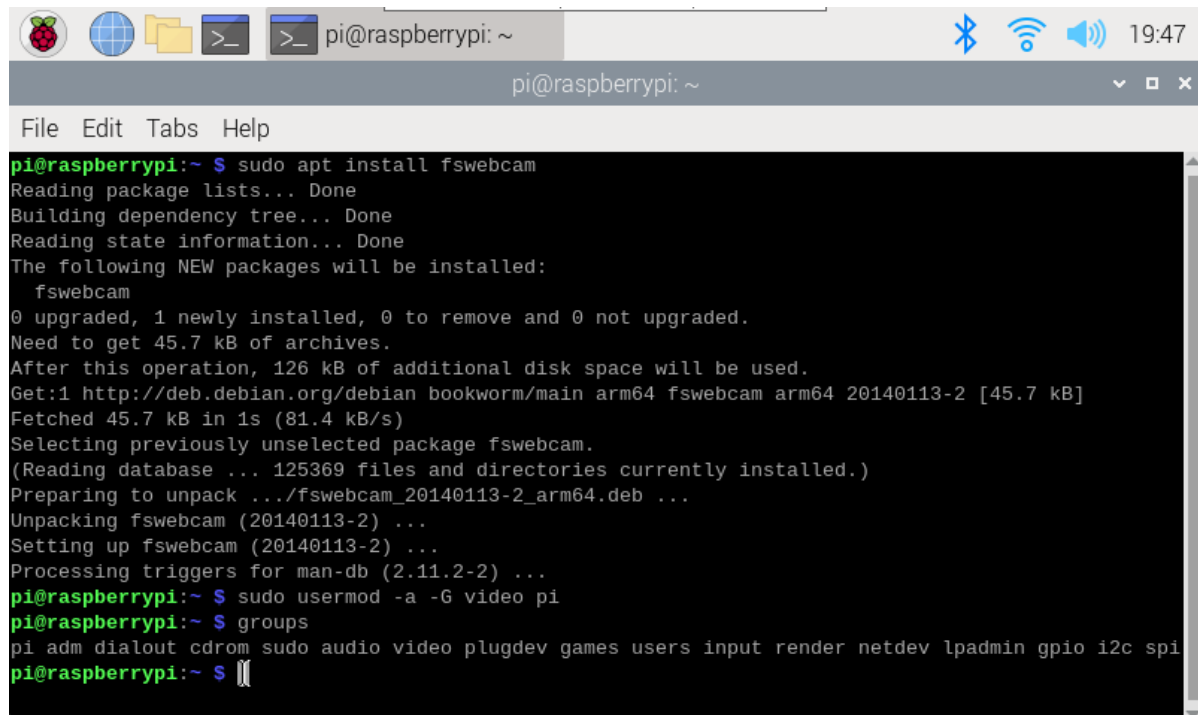
- Add user permissions: `sudo usermod -a -G video`

Example: Add pi user permissions to the group

```
sudo usermod -a -G video pi
```

- Check if the user has been added to the group correctly

Command: `groups`



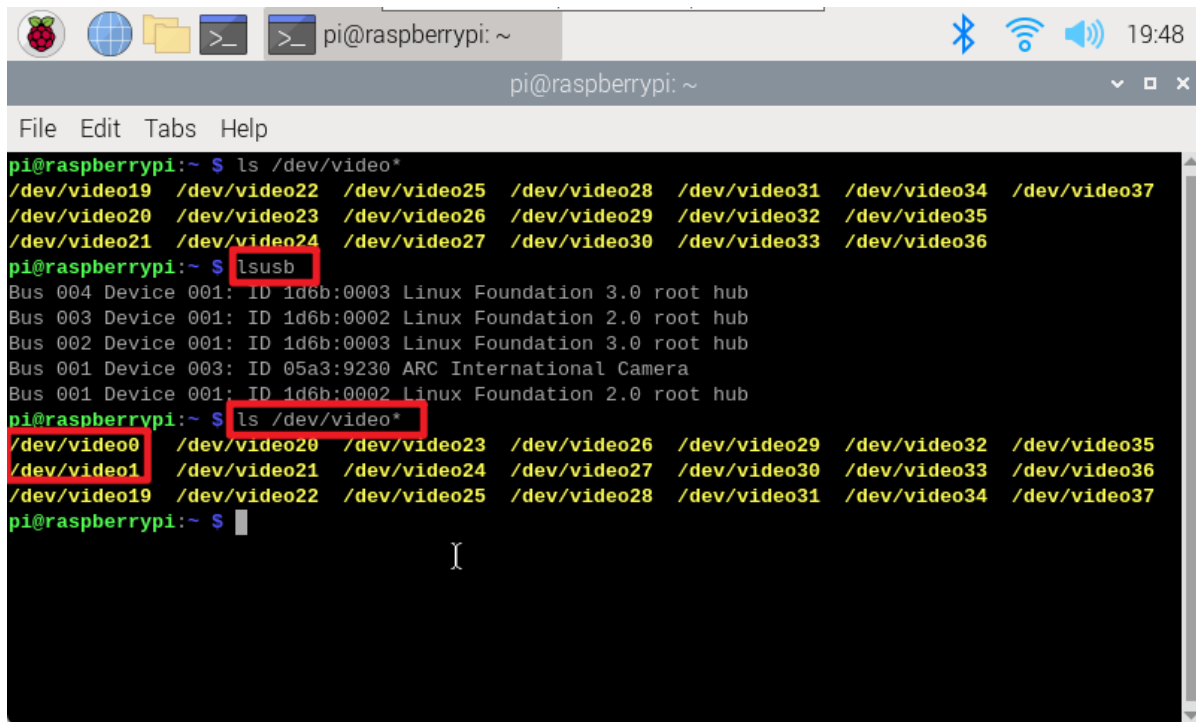
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt install fswebcam  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  fswebcam  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 45.7 kB of archives.  
After this operation, 126 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian bookworm/main arm64 fswebcam arm64 20140113-2 [45.7 kB]  
Fetched 45.7 kB in 1s (81.4 kB/s)  
Selecting previously unselected package fswebcam.  
(Reading database ... 125369 files and directories currently installed.)  
Preparing to unpack .../fswebcam_20140113-2_arm64.deb ...  
Unpacking fswebcam (20140113-2) ...  
Setting up fswebcam (20140113-2) ...  
Processing triggers for man-db (2.11.2-2) ...  
pi@raspberrypi:~$ sudo usermod -a -G video pi  
pi@raspberrypi:~$ groups  
pi adm dialout cdrom sudo audio video plugdev games users input render netdev lpadmin gpio i2c spi  
pi@raspberrypi:~$
```

View USB camera device

Use the `lsusb` command to view all USB devices recognized by the system;

Use the `ls /dev/video*` command to list all video devices recognized by the system.

The next two commands are to detect the information displayed by the camera. You can compare the differences by yourself:
One is image/video collection and the other is metadata collection.



The screenshot shows a terminal window on a Raspberry Pi. The top bar displays the Raspberry Pi logo, window icons, the user 'pi@raspberrypi', and the time '19:48'. The terminal has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The command prompt is 'pi@raspberrypi:~'. The first command executed is 'ls /dev/video*', which lists video devices from /dev/video19 to /dev/video37. The second command is 'lsusb', which lists USB devices, including the Linux Foundation hubs and the ARC International Camera. Red boxes highlight the 'lsusb' command and the camera entry in the output.

```
pi@raspberrypi:~$ ls /dev/video*
/dev/video19 /dev/video22 /dev/video25 /dev/video28 /dev/video31 /dev/video34 /dev/video37
/dev/video20 /dev/video23 /dev/video26 /dev/video29 /dev/video32 /dev/video35
/dev/video21 /dev/video24 /dev/video27 /dev/video30 /dev/video33 /dev/video36
pi@raspberrypi:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 05a3:9230 ARC International Camera
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~$ ls /dev/video*
/dev/video0 /dev/video20 /dev/video23 /dev/video26 /dev/video29 /dev/video32 /dev/video35
/dev/video1 /dev/video21 /dev/video24 /dev/video27 /dev/video30 /dev/video33 /dev/video36
/dev/video19 /dev/video22 /dev/video25 /dev/video28 /dev/video31 /dev/video34 /dev/video37
pi@raspberrypi:~$
```

Photograph

- `fswebcam <image_name>`

Example: Take a photo and save it as image.jpg (the file saving path defaults to the user directory)

```
fswebcam image.jpg
```

- `fswebcam -r resolution <image_name>`

Example: Take an image file with a resolution of 1280x720 and save it as image2.jpg

```
fswebcam -r 1280x720 image2.jpg
```

- `fswebcam -r resolution --no-banner <image_name>`

Example: Take an image file with a resolution of 1280x720, no information such as time is displayed on the picture, and save it as image3.jpg

```
fswebcam -r 1280x720 --no-banner image3.jpg
```

Time-lapse photography

Create a new Webcam folder and enter the file

```
mkdir webcam
```

```
cd webcam
```

Create a new webcam.sh script file and edit the content

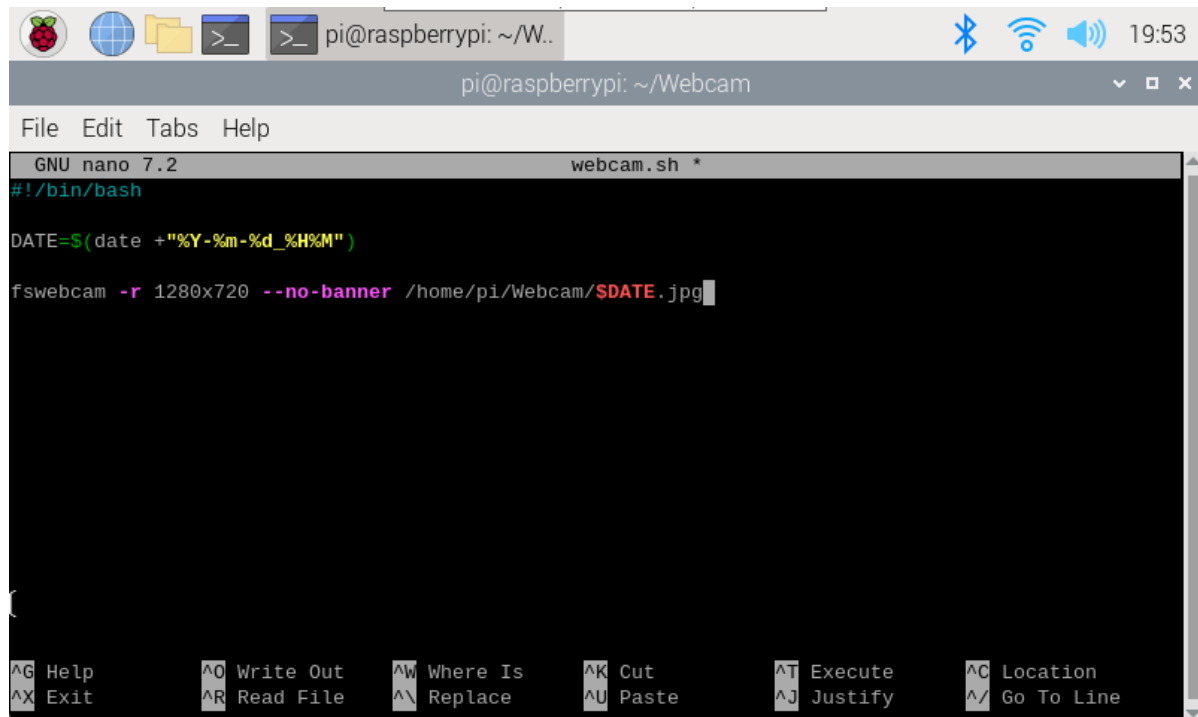
```
sudo nano webcam.sh
```

File content: The file saving path needs to be modified by yourself. My system username directory is yahboom.

```
#!/bin/bash

DATE=$(date +%Y-%m-%d_%H%M")

fswebcam -r 1280x720 --no-banner /home/pi/webcam/$DATE.jpg
```



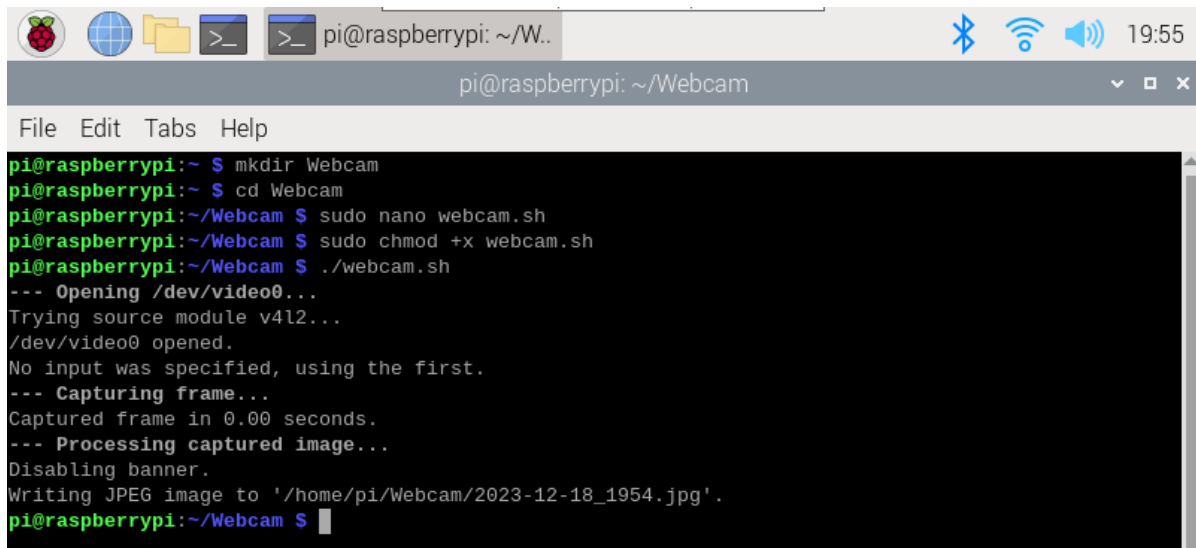
Hold down Ctrl+X, enter Y, and press Enter.

Add executable permissions

```
sudo chmod +x webcam.sh
```

run script

```
./webcam.sh
```

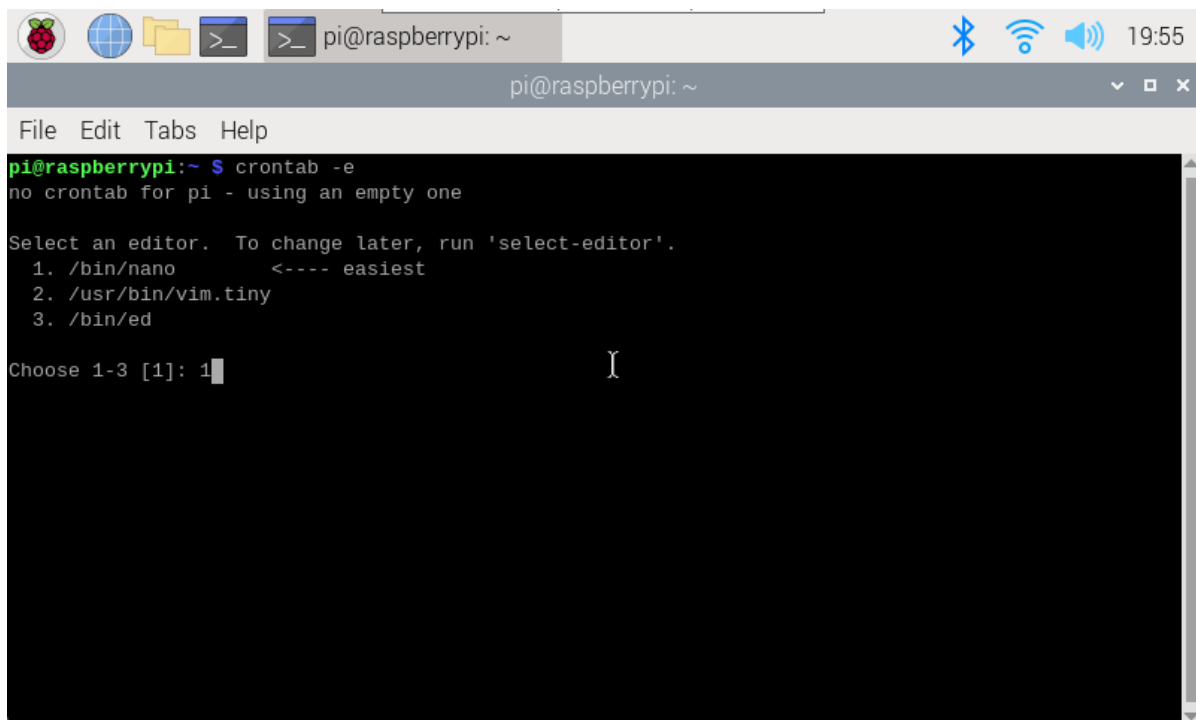


```
pi@raspberrypi: ~/W..  
pi@raspberrypi: ~/Webcam  
File Edit Tabs Help  
pi@raspberrypi:~$ mkdir Webcam  
pi@raspberrypi:~$ cd Webcam  
pi@raspberrypi:~/Webcam$ sudo nano webcam.sh  
pi@raspberrypi:~/Webcam$ sudo chmod +x webcam.sh  
pi@raspberrypi:~/Webcam$ ./webcam.sh  
--- Opening /dev/video0...  
Trying source module v4l2...  
/dev/video0 opened.  
No input was specified, using the first.  
--- Capturing frame...  
Captured frame in 0.00 seconds.  
--- Processing captured image...  
Disabling banner.  
Writing JPEG image to '/home/pi/Webcam/2023-12-18_1954.jpg'.  
pi@raspberrypi:~/Webcam$
```

Using Cron (scheduled tasks)

Open the cron table for editing. You will be prompted to select an editor when using it for the first time. It is recommended to use the nano editor.

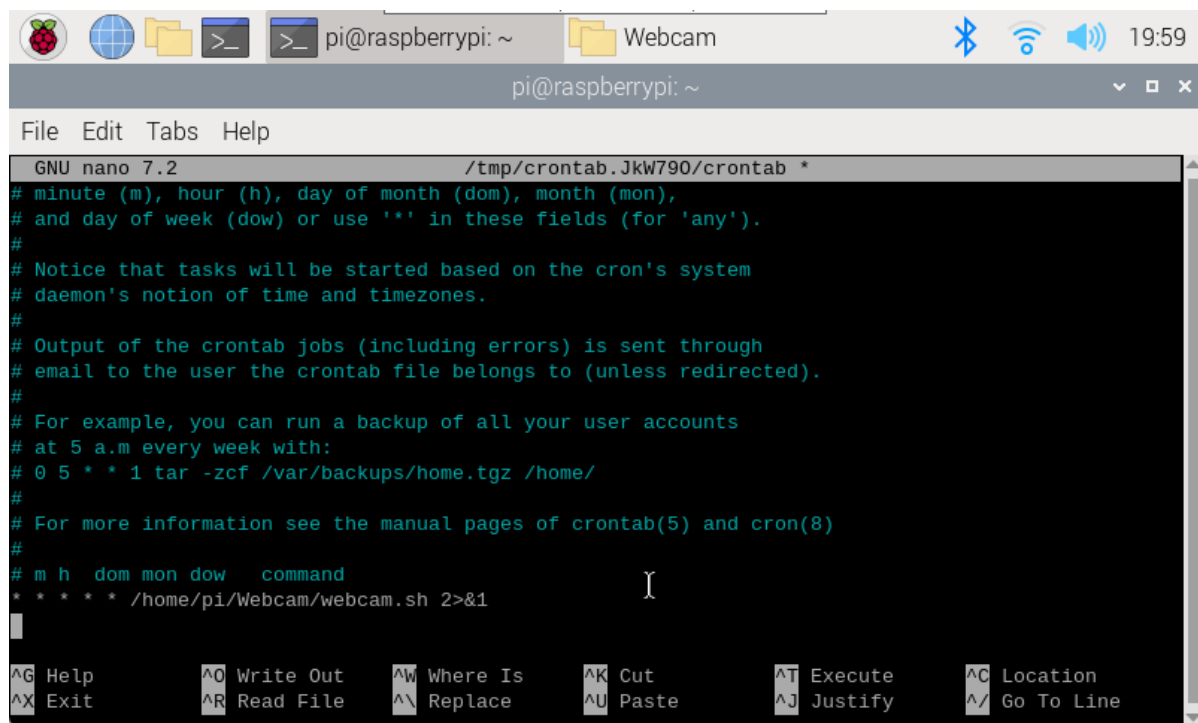
```
crontab -e
```



```
pi@raspberrypi: ~  
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ crontab -e  
no crontab for pi - using an empty one  
Select an editor. To change later, run 'select-editor'.  
1. /bin/nano <---- easiest  
2. /usr/bin/vim.tiny  
3. /bin/ed  
Choose 1-3 [1]: 1
```

Add the following code to the edited document: the first 5 * symbols represent a timer of 1 minute, and 2>&1 is to input the error output to the standard output.

```
* * * * * /home/pi/webcam/webcam.sh 2>&1
```



```
GNU nano 7.2 /tmp/crontab.Jkw790/crontab *
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /home/pi/Webcam/webcam.sh 2>&1
```

After saving the file and exiting, the terminal will output the following content:

```
crontab: installing new crontab
```

For Cron jobs, you can learn about format and syntax by yourself!

If no pictures are generated after one minute, you can restart the service and check whether the path is correct!

Start cron service: `sudo service cron start`

Stop the cron service: `sudo service cron stop`

