

Inhaltsverzeichnis

Important notice	2
Power the Picam	2
First test	2
Connec the PiCam to your wifi	2
Connect to your PiCam in your Network	3
Using multiple PiCams	3
Use the camera without tenseserflow	3
.....	3
Trouble shooting	3
Location and short description of running programs	4

Important notice

The connector for the camera has connections on both sides so the cable has not to be twisted to align the camera properly. The drawback of that is that **you can destroy the camera and your raspberry pi module if you connect it wrong!** On the connections board pin 1 and 15 are marked. make sure to connect pin 1 from your camera with pin 1 from the PiCam board. For the Pi HQ camera you can hold the camera over the board like it would be normally positioned insert the cable properly on the camera and without twisting the cable connect it to the PiCam module.

Power the Picam

Connect your USB micro b connector to the left usb slot.



First test

The device will boot and start a hotspot.

SSID: **taskit pi cam**

password: **12345678**

Connect to the hotspot, open a web browser and navigate to **http://192.168.42.1/**

The login for the Motioneye setup are the defaults. For admin access username admin and no password.

For normal user access both can be left empty.

Username: **admin**

Password:

The raw video footage is located at **http://192.168.42.1:8000/video_feed**

Connec the PiCam to your wifi

Open a web browser and navigate to <http://192.168.42.1:8080>. Select the Wifi you want to use and input your wifi password.

When you confirm your wifi settings you will get the notice below.

PiCam has closed its hotspot and tries to connect to your wifi with your settings.

Any other inputs you do now will not be sent to the PiCam.

 **Note!** Applying changes... Your device will soon be online. If connection is unsuccessful, the Access Point will be back up in a few minutes, and reloading this page will allow you to try again.

If you get the following notification you have already clicked on connect and are disconnected.

So any changes you did wont be applied.

 **Danger!** Failed to connect to the network.
NetworkError when attempting to fetch resource.

Connect to your PiCam in your Network

You can access the PiCam with the url **taskitpicam.local** in your browser.

If you want to use its ipv4-address send "**ping taskitpicam.local -4**" from your command line and use the ip.

Using multiple PiCams

When you add a second PiCam to your network the second cam will set its hostname to **taskitpicam-2.local** every new PiCam will iterate on that.

To access the additional PiCams follow the steps from **Connect to your PiCam in your Network**.

Use the camera without tensorflow

You might want to disable tensorflow(Object detection) to get a smoother video feed.

To do that use your favorite SSH client and connect to the Pi. The login data are the default raspberry pi logins.

User: **pi**, password: **raspberry**.

Run **systemctl stop tflite_example.service** followed by **systemctl disable tflite_example.service**.

Now that you have disabled tensorflow you can go back into your webbrowser and navigate to **taskitpicam.local**.

If you are not logged in as admin click on the little person and login as one, with User: **admin** and no password.

Delete the existing camera setup by clicking on the trashcan and add a **Local MMAL camera**, it should detect the connected camera automatically, apply your changes and you have your expected output.

Trouble shooting

The PiCam is not connected to the wifi after the steps in Connec the PiCam to your wifi.

If the PiCam was not able to connect to the wifi it will open its hotspot again so reconnect to it and input your settings again.

Location and short description of running programs

wifi-connect is used to get the wifi setup screen. the service is called `wifi_connect.service` all other related files are located in `/usr/local/share/wifi connect/` , `/usr/local/sbin/` and `/usr/local/bin/`. For more information about wifi connect visit <https://github.com/balena-io/wifi-connect>

motioneye is used to host the webserver that shows the camera output. The service file is called `motioneye.service`. For more information about motioneye visit <https://github.com/ccrisan/motioneye/wiki>

tenserflow lite is used to detect objects. the service is called `tflite_example.service`. The script used to detect objects and stream the video feed to motioneye is located at `/home/pi/tflite1/TFLite_detection_webcam.py` it is setup to use the virtual environment `tflite1-env`. For more information about the tensorflow setup visit <https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi> for more information about the streaming of the video stream visit <https://www.pyimagesearch.com/2019/09/02/opencv-stream-video-to-web-browser-html-page/>