

Introduction	2
Materials	3
Setup Trial & Error Log	5
Setup Instructions	8
Camera Tests	8
OV9281	9
AR0144	10
OV9281 + AR0144	10
Aruco Tests	12
AR0144	12

Introduction

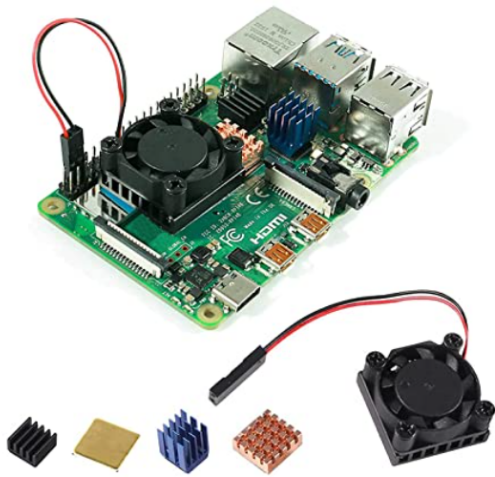
My adventures with Photonvision setup on an Orange Pi 5. The Opi5 is an excellent coprocessor for Apriltag processing, with impressive computing power in a very small package. Characteristics like its low price, 5V power supply, and bare-board packaging make it a very attractive Limelight alternative. Performance is excellent, capable of supporting a main robot thread running at 50fps at 640x480p, or even higher resolutions with some decimate. All this while drawing well under 10W of power at 5V, making it easy to supply with a readily available USB Buck converter.

Materials

- Orange Pi 5, 4GB RAM, \$100 (\$70 on Aliexpress)
 - Get a kit that comes with the case and a fan if possible. Aliexpress has many.
 - <https://www.aliexpress.us/item/3256804787372388.html>



- 5V 3A USB C power supply (wall wart for testing), \$10
- USB-C to HDMI converter, \$10
- Heatsink set (in case you forgot to get a case like I did), \$7



- OV9281 USB camera module, [\\$50](#)
- 5V 3A buck module, \$7 for 4



-
- 5V 2A battery bank, simultaneous charge/discharge (for on-robot backup battery)
 - Already owned this, your choice whether to buy one. Only some brands will support simultaneous charge/discharge so be careful. Order, test, and return if it doesn't work. Anker makes some good packs.
 - It may be possible to use something advertised as a Raspberry Pi UPS on the robot, as they are basically battery bank with 5V output. I would recommend wrapping them in something if you go this route to make them look nicer to an inspector. [Example](#).

Total cost: \$187

Setup Trial & Error Log

SKIP TO THE NEXT SECTION FOR ACTUAL SETUP INSTRUCTIONS. THIS IS JUST A LOG OF MY TRIALS.

I purchased an Orange Pi from Aliexpress for \$70 + \$41 shipping (ouch). I also found a USB-C to HDMI adapter in a drawer and a 2A 5V power supply with a USB C cable. The Orange Pi does not ship with these accessories and apparently only supports display output from its USB-C port. It also calls for a 5V 4A supply, but I don't own one of those, so... sucks. I also owned a 32GB uSD card already, so I'm using that. I also own an Ethernet extender that I used to connect the OPi to the internet, as it does not have built-in WiFi.

Total cost is maybe \$140? \$130? Depends on what you already own.

I downloaded the Armbian Bullseye CLI image from the place as mentioned in the PV documentation beta:

https://github.com/PhotonVision/photonvision-docs/blob/5303e7487ef04de67f2c8ef2563981cd928d3b3d/source/docs/getting-started/installation/sw_install/orange-pi.rst

Now, apart from those instructions, a change had to be made to the CPU "governor" to enable the high-performance cores (CPUs 4 to 7) for Photonvision at all times. To do this, uncommenting Line 92 in the install.sh script as shown here works:

<https://github.com/PhotonVision/photonvision/blob/0b5256df12fb751a1bc150450eaa98391b057ba9/scripts/install.sh#L92>

CPU settings:

For setting the CPUs to "performance", you want to do:

```
for f in /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor; do echo performance > $f; done
```

after each boot. I don't know of a standard way to do this; using "/etc/rc.local" is reasonable.

For forcing photonvision to run on the fast CPUs, add:

```
AllowedCPUs=4-7
```

into the "photonvision.service" file. (CPUs 4 through 7 are the fast ones on this model, and others I have seen.)

EDIT: corrected CPU model to "RK3588"

Supposedly, the install.sh script with the AllowedCPUs line uncommented will already do this.

Unfortunately the flash failed, so I tried with the official OrangePI Debian image [here](#) ([main page](#)) Debian Bullseye Server Linux. I chose the version with no desktop. I used the most recent version of Balena Etcher to flash the uSD card. This time, it started to boot.

Once the command line came up, I ran “sudo -s” to enter root user mode, then ran “apt-get update && apt-get upgrade” to update all packages. The username and password are both “orangepi” by default.

Then I ran the Photonvision installation instructions for [Debian-based coprocessors](#), with the caveat that I edited the install.sh script to uncomment the CPU line before continuing. I rebooted the PC as per the instructions and waited...

It rebooted in about 10 seconds. Using “systemctl status photonvision.service” showed it running. Photonvision.local did not come up. I was able to connect by directly accessing the IP address (which I found using “ip a”). Restarting Photonvision after setting a team number made photonvision.local work, which was good.

Plugging in a Lifecam to the sideways USB port did not work. But the horizontal ones worked sort of, but didn’t display a camera feed. Rebooting the device made the web UI go away. However, I did see that the ip address had been successfully set to what I entered in the config menu! The hostname also now showed up as “photonvision” on the OrangePi monitor.

So, I just need to get the thing back...

Connecting through a robot radio fixed everything. Must be an issue with my wifi extender. It takes about 20 seconds to boot from nothing to getting something at ipaddress:5800. Photonvision.local does not seem to work atm.

The OrangePi does seem to reboot occasionally, probably because of power spikes dropping the voltage or something. I will look for a stronger 3A supply.

Update: found a 3A supply and issues are gone. Most importantly, this one puts out 5.1V by default, so the CPU never undervolts. It only draws ~1.1A average, 1.2A max with an OV9281 USB camera connected.

Beta 8 appears to work fine.

A picture of my setup, with tiny heatsinks I found in a bin:



A proper casing will likely increase the performance of the camera up to or beyond Beelink N5095 Mini S levels, if paired with overclocking. After adding a proper fanned heatsink like the one linked on the first page, performance doesn't improve much without overclocking. Although the CPU is kept incredibly cool at 40C.

Setup Instructions

Download the official OrangePI Debian image [here](#) ([main page](#)). I chose the version with no desktop. This will be Debian Bullseye 11. Use the most recent version of Balena Etcher to flash it to a 8GB or larger uSD card (I used 32GB). Plug the SD card into the Orange Pi. Connect the USB-C to HDMI adapter to the Orange Pi and connect a monitor/display to check the output. A USB keyboard is also recommended. Add your casing/heatsinks to the Orange Pi. Connect the Orange Pi 5 to a 5V 3A power supply, although 5.1V 2A should work fine as well. Let the operating system come up.

Once the OS is setup, follow the Photonvision instructions for [installation on Debian-Based Coprocessors](#). However, before running install.sh, uncomment Line 92 to force PV to run on the “performance” CPU cores 4-7.

```
89  nice=-10
90  # for non-uniform CPUs, like big.LITTLE, you want to sel
91  # look up the right values for your CPU
92  # AllowCPUs=4-7
93
94  ExecStart=/usr/bin/java -Xmx512m -jar /opt/photonvision/
95  ExecStop=/bin/systemctl kill photonvision
```

Run install.sh. Sudo reboot like normal. You should now be able to access Photonvision by going to photonvision.local:5800. Set a static IP, team #, and all that, then reboot. You will be able to access Photonvision via the IP address like normal, or possibly with photonvision.local. mDNS is a little weird so only one might work - make sure your computer and the range Pi are on the same network!

And that's it. It should just work.

Camera Tests

I tried a couple cameras. These tests were performed at the following settings unless otherwise noted:

INPUT

APRILTAG

OUTPUT

Select target family

tag16h5

Decimate

1

Blur

0

Threads

3

Refine Edges

Decision Margin Cutoff

10

Pose Estimation Iterations

50

All cameras with M12 lenses used 100deg no-distortion lenses unless otherwise noted.

OV9281

Resolution	Framerate (FPS)	Latency (ms)	Range (ft)
640x480 MJPEG 120fps, Decimate 2	64	11	21
640x480 MJPEG 120fps	44	19	25
800x600 MJPEG 120fps	40	21	28
1280x720 MJPEG 120fps, Decimate 2	30	30	31
1280x720 MJPEG 120fps	22	48	32+

FPS is a little lower than the above measurements for about 10 seconds, before stabilizing at the higher numbers.

The Opi drew 5V 1.2-1.3A in this config. Adding the heatsink did not increase performance much (the heatsink-less numbers are maybe 10% worse than these) but it does reduce power draw to 1-1.1A.

AR0144

Another classic global shutter camera, albeit one with lower maximum FPS. 4 threads used for all tests.

Resolution	Framerate (FPS)	Latency (ms)	Range (ft)
640x480 MJPEG 60fps, Decimate 2	60	10	25
640x480 MJPEG 60fps	30	21	27
1280x720 MJPEG 60fps, Decimate 2	28	32	32
1280x720 MJPEG 60fps	15	54	33+

OV9281 + AR0144

I plugged in both cameras to the horizontal USB ports. The vertical USB port was spotty on mine but appears to work fine now, so it might have been a transient bug or a mis-seated connector. I just have the keyboard dongle plugged in there.

Both cameras were run with 2 threads each.

Unfortunately the interface kept bugging out, so I will do a fresh install with the newest version before trying this again. Edit: the bug was that if resolution of the OV9281 was changed with another camera plugged in and the OV9281 using more than 1 thread, OV9281 would stop sending new frames. This was fixed by unplugging and replugging the camera.

Resolution	OV9281 FPS	OV9281 Latency (ms)	AR0144 FPS	AR0144 Latency (ms)
640x480 MJPEG 120fps, Dec. 2	55	17	55	17
640x480 MJPEG 120fps	45	18	28	30
1280x720 MJPEG 120fps, Dec. 2	31	31	25	35
1280x720 MJPEG 120fps	18	50	20	48

At 1280x720, the camera varied from 20-30fps in a cycle lasting about 5 seconds. Kind of annoying. The values in the table were taken with the OV9281 running only 1 thread. Performance was always better with multiple threads, but swung around.

8-31-23 OV2311

Using a 2023.4.2 build, I tested a fancy \$100 2MP OV2311 global shutter Arducam module.

Resolution	Framerate (FPS)	Latency (ms)	Range (ft)
640x480 MJPEG 50fps, Decimate 2	50	11	?
640x480 MJPEG 50fps	25	25	?
1280x720 MJPEG 50fps, Decimate 2	25	33	?
1280x720 MJPEG 50fps	16	48	?
1280x960 MJPEG 50fps, Decimate 2	25	30	?
1280x960 MJPEG 50fps	14	60	?
1600x1200 MJPEG 50fps, Decimate 2	16	50	?
1600x1200 MJPEG 50fps	10	92	?

There was some motion blur at high speeds unless the exposure was set dangerously low. This seemed to get worse at higher resolutions.

8-31-23 OV9712

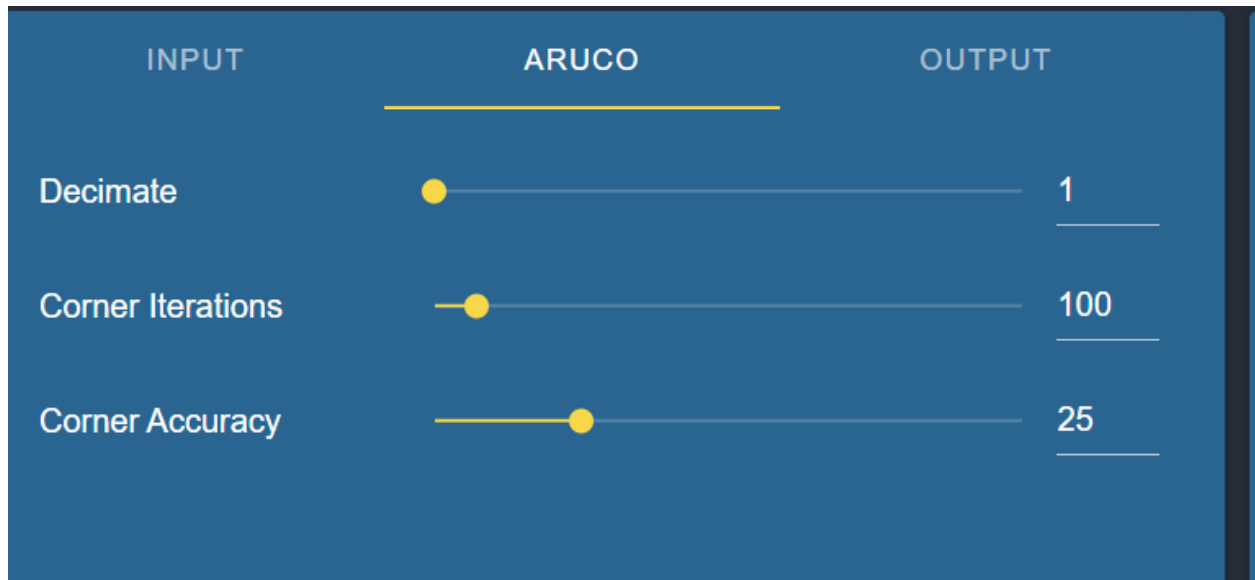
This was a fairly moderately priced camera at \$31. It supports 60fps up to 640x360, and 30fps up to 1280x720. This has an M7 lens and often comes in fisheye, which photonvision has trouble parsing. Overall, just a decent camera, with only a little motion blur and easily adjustable exposure. I would not run this camera at higher resolution, because although it doesn't have a lot of motion blur, it does process weirdly slowly. Definitely avoid the fisheye lens.

Resolution	Framerate (FPS)	Latency (ms)	Range (ft)
640x360 MJPEG 60fps, Decimate 2	54	12	?
640x360 MJPEG 60fps	26	31	?

1280x720 MJPEG 30fps, Decimate 2	15	39	?
1280x720 MJPEG 30fps	10	90	?

Aruco Tests

The new Aruco tag detection method for Apriltags has the capacity to be much faster.
The below tests were done with the following settings:



AR0144

Resolution	Framerate (FPS)	Latency (ms)	Range (ft)
640x480 MJPEG 60fps	60	10	17.5
800x600 MJPEG 60fps	40*	20*	20
1280x720 MJPEG 60fps	23*	40*	21

* = jumpy/inconsistent