One

Long-Range Surveillance Camera

James A Crawford

Synopsis

My driveway is on the order of 1,000 feet long so monitoring products like those from Blink are simply not suited to my needs. Cameras with (i) high resolution, (ii) optical gain on the order of 10X, (iii) communication link capabilities from 3 yards to over 300 yards, and (iv) with outdoor weather-proofing fall into the commercial/professional realm and are very expensive. In addition, I wanted (i) simple motion detection, (ii) an easy way to archive image data, and (iii) ideally an easy way to issue email alerts when motion was detected.

This rather brief memorandum outlines a solution I formulated and built around a Raspberry Pi4. The cameras have been operating without any problems for almost a year now and also survived a typical hot southern California summer without any thermal problems. My approach also proved very effective in combatting condensation which is always an issue during seasonal changes.

1 Introduction

The completed and installed long-range surveillance camera is shown in Figure 2 through Figure 5 with an example capture shown in Figure 1. The heart of the camera portion is a Raspberry Pi4 and associated Raspberry Pi CCD telephoto lens. The wireless communications link with my home office is provided by a pair of 2.4 GHz TP-Link units. Although only one camera setup is shown in the following pictures, I have multiple cameras scattered around my property all tied back through a master TP-Link unit, some running off their own solar panels for the units situated at larger distances.

The master TP-Link indoor unit is tied into my wired Ethernet network and all of the captured pictures/video are automatically dumped onto one of my network appliances which have multiple terabyte capacities. No chance of running out of storage with this scheme!

Figure 1 Example still photo capture in which approaching mail truck automatically captured and saved

Motion detection software running on each camera's Raspberry Pi4 automatically triggers image capture operations as well as sending the information via email to my cellphone.

I have been running the camera setup described in this memo continuously for almost a year now and am quite happy with the results.

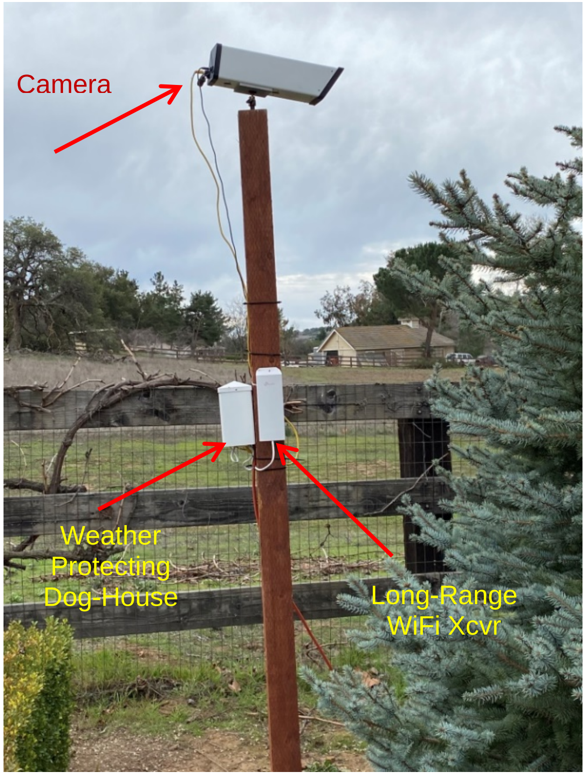

Figure 2 Surveillance camera mounted atop 8' post directed up the long driveway. The dog-house contains 115V AC splitter and low-voltage power brick for the WiFi transceiver.

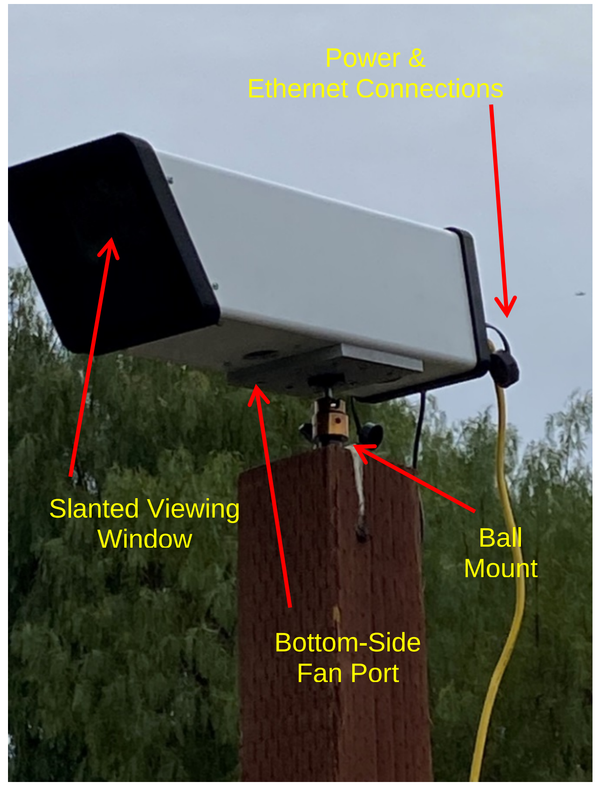

Figure 3 Close-up of installed surveillance camera. Black pieces printed on 3D printer.

Figure 4 View of mounted surveillance camera from driveway

Figure 5 Coming up the driveway from a distance

2 Partial Bill of Materials

Table 2-1 List of Primary Materials

Item Description	Additional Information	Cost
4" square section of white vinyl fence post material	From Home Depot, with 30^{o} angle slant-cut. Had to purchase 6'section.	$25
Ball-head swivel	Amazon, Camvate	$10
Front cap window (2 pieces)	Fabricated on my 3D printer. Fusion 360 design available for asking.	
Rear cap	Fabricated on my 3D printer. Fusion 360 design available for asking.	
CAT-5 outdoor connector	Amazon, RJ45, package of 2	$10
Outdoor power connector	Amazon, USB Type C	$13
Raspberry Pi4, 4 GB	PiShop.us	$84
Raspberry Pi HQ Camera	PiShop.us Raspberry Pi HQ Camera 16mm Telephoto Lens	 $50 $63
Miniature ventilation fans, 30x30mm	Amazon, package of 4	$11
TP-Link 2.4 GHz 300 Mbps 9dBi Outdoor CPE	Amazon, Pharos CPE210	$40
MotionEyeOS	Raspberry Pi4 operating system for camera(s)	$0

3 Camera Construction

The main body of the camera was cut from a section of 4"-square white vinyl fence post material, with the front edge angled at 30^{o}.

A piece of Trex-like decking material was milled down to about 0.5" thickness and used to support all of the electronics as shown in Figure 6 through Figure 9. This made it possible to easily slip the entire electronics assembly into the rectangular enclosure in one step.

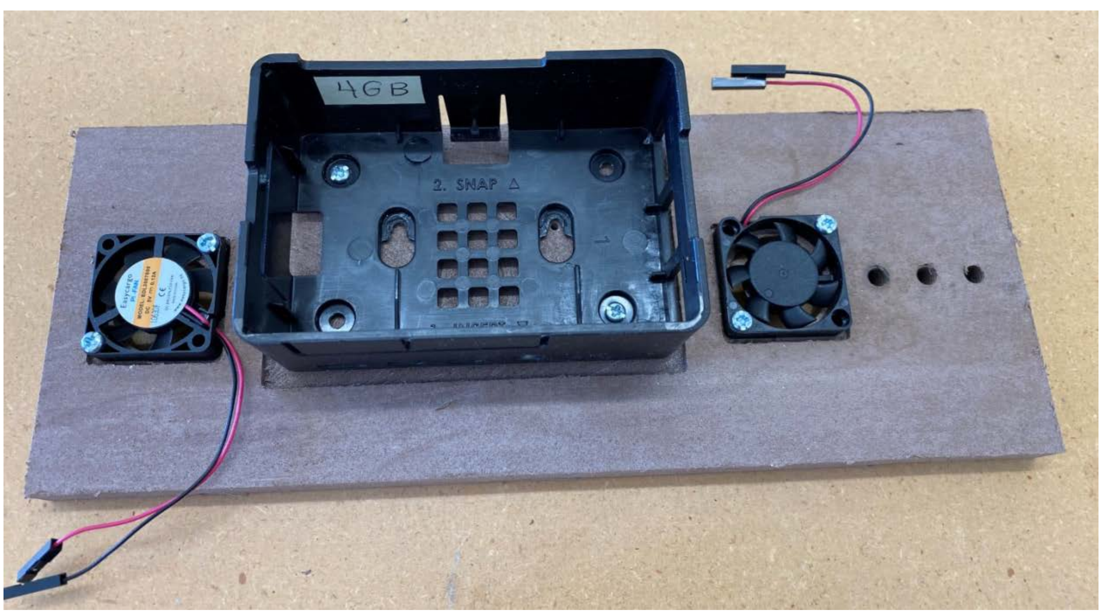

Figure 6 All electronic items were mounted on a milled piece of Trex-like decking material for easy installment into the rectangular enclosure. The two 5V fans work in unison to pull outside-air into the enclosure and push *used* inside-air out. These were absolutely necessary to prevent dew-point related moisture condensation from building up inside the enclosure. The large rectangular shell securely holds the Raspberry Pi4 PCB.

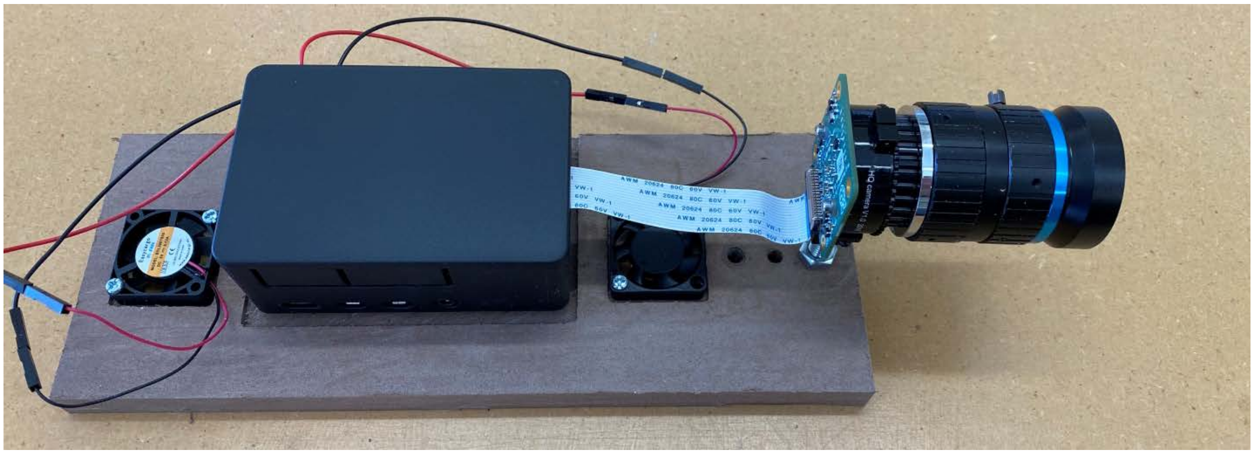

Figure 7 Electronics almost ready for insertion into the rectangular enclosure. Taking advantage of gravity, I also fabricated small square chimneys about 1.5" tall to snuggly fit over each fan in order to prevent possible small moisture drops (e.g., rain) from being vertically sucked inside the enclosure. Two rain storms have proven this approach very effective. The two fan-ports are shown in Figure 10.

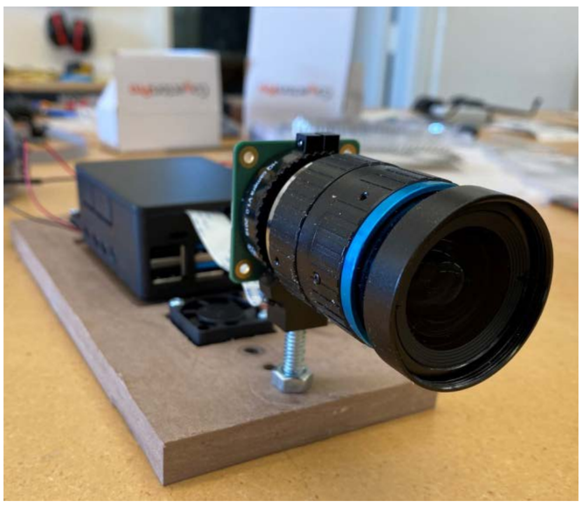

Figure 8 Front view of the Raspberry Pi HQ camera prior to installation into the enclosure

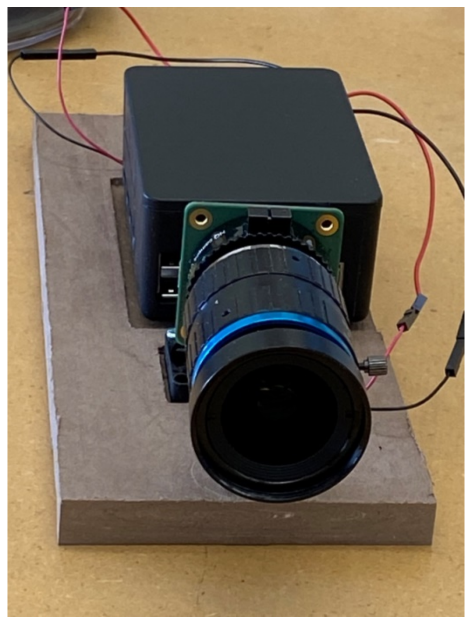

Figure 9 Another perspective of Figure 8

Additional close-up shots of the fully assembled camera are shown in Figure 10 through Figure 12.

Figure 10 Underside fan port-holes. Milled ¼” plate for secure attachment to the ball-mount also shown.

Figure 11 Underside perspective of the view window. Placing the polycarbonate window at an angle mitigates otherwise strong internal reflections.

Figure 12 Professional grade outdoor water-resistance connectors were installed for ease of use as well as aesthetic purposes

4 Wireless Communications Link

The wireless communication link from each remote camera is handled by a pair of TP-Link 2.4 GHz 300 Mbps units which I purchased on Amazon. These have proven to be very reliable and have a free-space communication range of multiple miles! The 9 dBi antennas help minimize Fresnel zone reflection issues.

5 Camera Software and Configuration

The MotionEyeOS operating system is available on the web for free. This is probably the one aspect of my setup which could stand some improvement. As mentioned earlier, I have all of the TP-Link base receivers tied into my wired Ethernet. I have a separate dedicated Raspberry Pi4 connected to a computer monitor at my office desk so that I can see what is going on around my property without even leaving my chair.

In my case, my longer-term plan is to use the Raspberry Pi4 for real-time image processing for my telescope project. I will likely use the results of that effort to improve the software for my surveillance camera system.

I am also anxious to use some of my older Canon telephoto lenses in a similar setup with a Raspberry Pi. The electrical control interface for focusing, zooming, and exposure is relatively simple and the built-in servo-motors would greatly simplify the overall project. I only need to mill an adapter to marry the telephoto lens to the Raspberry Pi’s sensor mount.

