# Description

This report is not necessarily a software project, but rather a description of the work performed to augment the Quadcopter to a Raspberry Pi (RPi) Camera Drone.

A picture containing indoor

Description automatically generated

Figure – Drone Cam

# Requirements

The following requirements describe the functionality set forth by the Drone Cam project:

* The Drone shall transport an RPi SoC board.
  + Derived: The Drone shall be able to supply 5 volts.
* The Drone shall transport a camera.
* The Drone camera shall communicate with the onboard RPi.
* The Drone RPi shall transmit the camera footage to a Host
* The Drone shall be registered as a UAV under FAA regulations.

# Design

The Drone Cam project consists of two sections: Hardware and Software.

## Hardware

The following Hardware use to augment the basic Drone to the Drone Cam:

* Raspberry Pi Model 3B
* MicroSD Card
* ~12v to 5v Buck Converter
* USB to Micro-USB cable (~5 inches in length)
* Old Mains Power Cable (no ground) (this will be spliced and stripped)
* Extra Double-Sided Mounting Tape (3M brand was used)
* Smallest Mini 2.0 50.0 Mega Pixel USB HD Video Camera Webcam Web Cam For PC Laptop
* Tools:
  + Solder Iron
  + Solder
  + Wire Cutters
  + Wire Strippers
  + Multimeter

A picture containing text

Description automatically generated

Figure – Drone Cam Closeup

### 5 volt supply

The Drone mains battery supplies ~12 volts on the rails. Supplying the RPi with this voltage will fry the board. This necessitated the need for a 12v to 5v buck converter. This section describes the actions taken to provide this.

To connect the mains battery to the buck converter, a special cable needed to be created. The following procedure outlines the actions taken:

1. The Mains Power Cable terminals (plug and jack) were cut off using wire cutters. NOTE: I elected to sacrifice a power cable since the battery has an amperage rating of 2200mAh. I could have tested this with normal small jumper cables (since I don’t think the buck converter would draw anywhere near that much), but I did not want to warrant risk dealing with this amount of current.
2. An appropriate length of cable (~7 inches) was cut.
3. The two lines of the cable were separated at most 1 inch at both ends.
4. Both lines at both ends were stripped.
5. The copper at both ends were tinned.

The modified power cable was soldered in parallel to the connecting pads for the battery connector. The other end of the power cable was connected to the VIN+/VIN- ports on the buck converter. Before connecting the battery, a continuity test was performed against the positive and negative ends of the battery terminals versus the buck converter to verify wiring. The battery was connected and 5 volts was verified on the output end of the converter.

The buck converter was mounted on an open surface of the drone plate using double sided mounting tape. The modified power cable was routed within the housing for wire management.

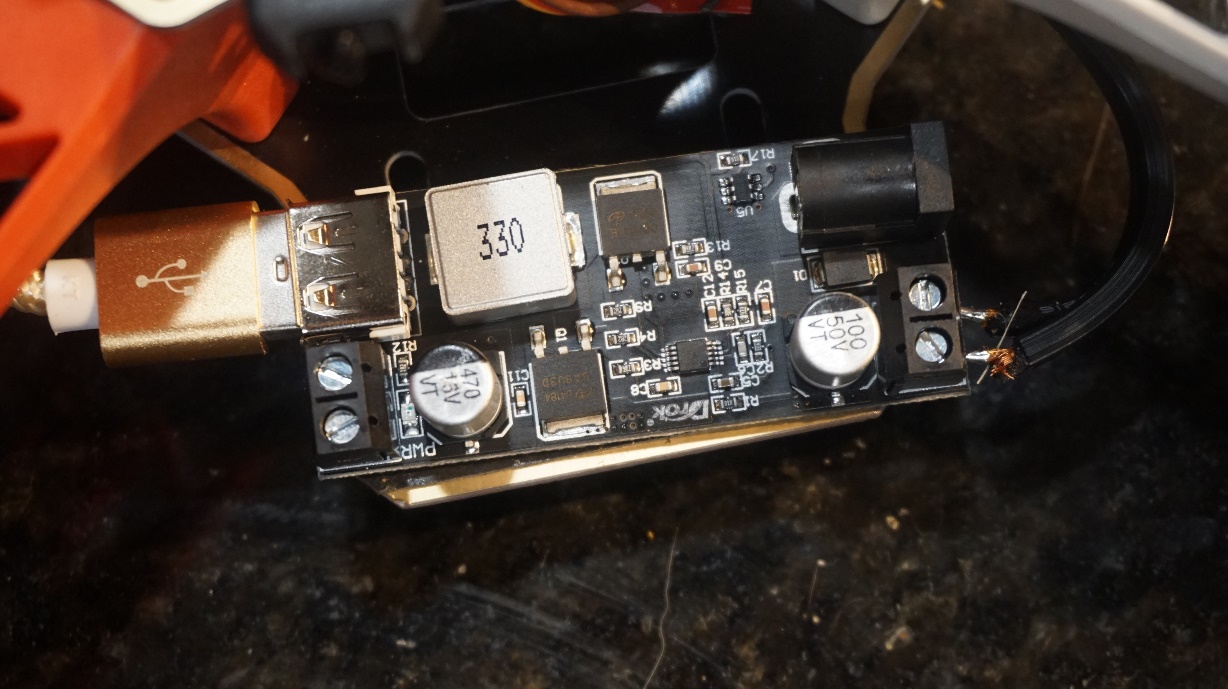


Figure – Buck Converter

### Raspberry Pi and Camera

The RPi was mounted at a central location on top of the drone. Because the battery and RPi both take up a significant footprint, their placements must be negotiated. It was decided that the RPi would be “tail-heavy” with the battery placed in an aft position with its orientation parallel to the drone’s longitudinal or pitch axis. The RPi would take up the space immediately in front of the battery, with as much of its footprint on the mounting plate.

The camera was mounted on the Drone bottom plate. The clip-on provided on the camera will serve as mounting hardware. This is not robust for flight and this issue has been notated.

The Raspberry Pi and Camera were connected via USB. The RPi connects to the output end of the buck converter using the USB to Micro-USB cable. All cables were stowed in the inner Drone housing for wire management.

A close-up of a circuit board

Description automatically generated with medium confidence

Figure – RPi Mounted



Figure – Cam Mounted

## Software

The RPi was installed with the Raspbian OS. The Motion package was installed and enabled on boot. Unfortunately (at the time of writing), booting up the RPi does not spin up the Motion daemon and must be manually started by invoking ‘motion’ on the command line. This issue has been noted.

### IP Settings

The RPi was configured as a client in my home network. I modified the DHCP reservation for the RPi to always allocate a specific IP address for it.

### Viewing Footage

The Drone camera footage may be viewed by opening a web browser to <RPi IP Address>:8081.

### Configuration

The following is the Motion configuration file used in the RPi (/etc/motion/motion.config):

# Rename this distribution example file to motion.conf

#

# This config file was generated by motion 4.3.2 Documentation: /usr/share/doc/motion/motion\_guide.html

#

# This file contains only the basic configuration options to get a

# system working.  There are many more options available.  Please

# consult the documentation for the complete list of all options.

#

############################################################

# System control configuration parameters

############################################################

# Start in daemon (background) mode and release terminal.

daemon on

# Start in Setup-Mode, daemon disabled.

setup\_mode off

# File to store the process ID.

; pid\_file value

# File to write logs messages into.  If not defined stderr and syslog is used.

; log\_file /var/log/motion/motion.log

# Level of log messages [1..9] (EMG, ALR, CRT, ERR, WRN, NTC, INF, DBG, ALL).

log\_level 6

# Target directory for pictures, snapshots and movies

target\_dir /var/lib/motion

# Video device (e.g. /dev/video0) to be used for capturing.

videodevice /dev/video0

# Parameters to control video device.  See motion\_guide.html

; vid\_control\_params value

# The full URL of the network camera stream.

; netcam\_url value

# Name of mmal camera (e.g. vc.ril.camera for pi camera).

; mmalcam\_name value

# Camera control parameters (see raspivid/raspistill tool documentation)

; mmalcam\_control\_params value

############################################################

# Image Processing configuration parameters

############################################################

# Image width in pixels.

width 640

# Image height in pixels.

height 480

# Maximum number of frames to be captured per second.

framerate 120

# Text to be overlayed in the lower left corner of images

text\_left CAMERA1

# Text to be overlayed in the lower right corner of images.

text\_right %Y-%m-%d\n%T-%q

############################################################

# Motion detection configuration parameters

############################################################

# Always save pictures and movies even if there was no motion.

emulate\_motion off

# Threshold for number of changed pixels that triggers motion.

threshold 1500

# Noise threshold for the motion detection.

; noise\_level 32

# Despeckle the image using (E/e)rode or (D/d)ilate or (l)abel.

despeckle\_filter EedDl

# Number of images that must contain motion to trigger an event.

minimum\_motion\_frames 1

# Gap in seconds of no motion detected that triggers the end of an event.

event\_gap 60

# The number of pre-captured (buffered) pictures from before motion.

pre\_capture 3

# Number of frames to capture after motion is no longer detected.

post\_capture 5

############################################################

# Script execution configuration parameters

############################################################

# Command to be executed when an event starts.

; on\_event\_start value

# Command to be executed when an event ends.

; on\_event\_end value

# Command to be executed when a movie file is closed.

; on\_movie\_end value

############################################################

# Picture output configuration parameters

############################################################

# Output pictures when motion is detected

picture\_output off

# File name(without extension) for pictures relative to target directory

picture\_filename %Y%m%d%H%M%S-%q

############################################################

# Movie output configuration parameters

############################################################

# Create movies of motion events.

movie\_output on

# Maximum length of movie in seconds.

movie\_max\_time 60

# The encoding quality of the movie. (0=use bitrate. 1=worst quality, 100=best)

movie\_quality 50

# Container/Codec to used for the movie. See motion\_guide.html

movie\_codec mkv

# File name(without extension) for movies relative to target directory

movie\_filename %t-%v-%Y%m%d%H%M%S

############################################################

# Webcontrol configuration parameters

############################################################

# Port number used for the webcontrol.

webcontrol\_port 8080

# Restrict webcontrol connections to the localhost.

webcontrol\_localhost off

# Type of configuration options to allow via the webcontrol.

webcontrol\_parms 0

############################################################

# Live stream configuration parameters

############################################################

# The port number for the live stream.

stream\_port 8081

# Restrict stream connections to the localhost.

stream\_localhost off

##############################################################

# Camera config files - One for each camera.

##############################################################

; camera /usr/etc/motion/camera1.conf

; camera /usr/etc/motion/camera2.conf

; camera /usr/etc/motion/camera3.conf

; camera /usr/etc/motion/camera4.conf

##############################################################

# Directory to read '.conf' files for cameras.

##############################################################

; camera\_dir /usr/etc/motion/conf.d

# Known Issues

The following list are known squawks encountered during Drone development. It should be noted that some issues have been resolved during development while others are still present and monitored.

* Bad Battery (RESOLVED):
  + ~~The battery delivered was apparently INOP. An attempt was made to charge the battery with the provided charger. The charger did not indicate a charge being performed. After letting charge for approximately 2 hours, the main output terminals indicated only approximately 3 volts DC, rather than the expected ~11 volts.~~
  + Extra batteries were ordered via Amazon. Specs are similar to the original battery (voltage is matches, but the milliampere hours (mAh) and charge constants (C) are allowed to vary).
* Yaw Potentiometer Broken (RESOLVED – WORKAROUND):
  + The Yaw Potentiometer in the KK Controller seems to broken. Rotating the pot does not have a stopping point. Further investigation is needed.
  + No noticeable impact from this issue. This is a non-grounding squawk.
* Inconsistent Motor (RESOLVED)
  + ~~The aft motors seem to be slow to start. If under no load, the aft motors spin at the same time as the front, however, having a propeller load impedes the motor start.~~
  + Extra motors were ordered via Amazon. Investigation over the impact of swapping motors is to be conducted in the future.
  + ~~There is a possibility that calibration is lacking or improperly configured. Future investigation will be conducted.~~
  + Replacing the bad motor allows for an actual takeoff. More practice is needed for proficiency.
* Motion Application Issues
  + Annoyingly, support for motion startup at boot (daemon mode) does not work.
  + The framerate of the Motion RPi library is 1 FPS on browser. Need a more robust method and investigation
* Poor Camera Mounting
  + The camera will shift or break off at hard landings. Need to find a more secure way to lock position.

# FAA Registration

The Drone was registered under FAA Part 107. The Drone is designed under Tail Number (Nickname) of N42069 (pronounced “November-four-twenty-sixty-nine”) and Registration Number of FA3NL7PRFE.



# DEMO

A demonstration of the Drone Cam can be found here:

<https://youtu.be/CoE0_4jI_nw>