This document's purpose is to provide an explanation of how the camera system works as all of the working parts are a little convoluted.

How an ideal camera system should work

- Camera streams data to flight controller, either directly or via a Raspberry Pi
- Flight controller streams information back to ground station so that live image can be seen
- Image is processed (either onboard or on ground station) and flight controller moves the drone and/or sprayer to address camera feedback about where has been sprayed or needs to be sprayed or sprayed more.

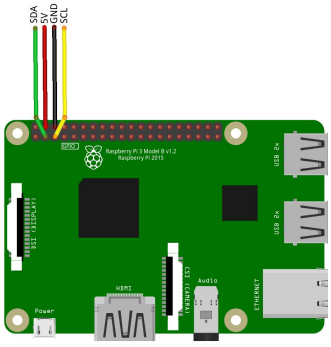
How the camera system actually works in its current form

- Camera is attached to Raspberry Pi
- Camera sends data to Pi over I2C
- Pi displays image on desktop using example program
- Screen recording software records the part of the screen that is displaying the camera
- The recorded screen video is saved locally on the Pi to be retrieved later for debugging and analysis

Obviously the present setup is not optimal, but it is currently working

Hardware

- There are 4 wires coming out of the camera. Each is labeled on the board but the wires are also color coded for convenience:
    - Red>Orange - 5V power
    - Black - Ground
    - Green - Data (SDA)
    - Yellow>Orange/Black stripe - Clock (SCL)
- These 4 wires connect to the appropriate GPIO pins on the Raspberry Pi according to the pin map and as shown in the image below:



- The pins need to have I2C enabled in the Pi software for them to perform as those types of inputs
- The GPIO pin layout is the same on the Pi 3 and Pi zero

Software

- The git repository is located at /home/pi/snow-drones
- The mlx90640-library folder is just a (slightly modified) copy of the example library that is provided by the manufacturer.
    - examples/sdlscale shows a fullscreen image of what you would expect the output of the thermal camera to look like
    - examples/fbuf is the program the begins on startup that has been modified to show a grayscale image from the camera, ideal for computer vision tasks
    - Both of these programs are run by navigating to the mlx90640-library folder via the terminal and typing “examples/*programname*”
- Inside /home/pi/.config/autostart there are a number of .desktop files that start programs once the pi is powered on
    - fbuf.desktop starts displaying the camera feed as described above
    - simplescreenrecorder.desktop starts SimpleScreenRecorder, the program used to record the video stream
        - SimpleScreenRecorder is preloaded with a profile that captures the video as we want for this application (only the part of the screen with camera images, 10 FPS etc.). This can be easily changed to suit your needs.
    - screenRecord.desktop runs startCameraRecord.sh
        - startCameraRecord.sh is a bash script that automates the process of starting and saving the recorded videos
        - It is fairly well commented and includes the contents of all of the .desktop files as well so that they are backed up to git
        - It uses xdotool to move the mouse out of the video frame and presses the hotkey (q) that tells SimpleScreenRecorder to start recording
        - The video will not save if power is cut while recording, so every 60 seconds it saves the current recording and starts a new one. This process takes ~0.2 seconds, so 2 frames will be lost when stitching each video that makes up a single flight together (If you choose to do so)
- Recordings from the camera are saved as .mp4 files to /home/pi/Videos

    - The file name of each video is SnowDrone with the date and time it was captured appended to the end

Tips

- VNC is installed on the Pi, allowing for remote desktop access which can be easier than plugging everything in every time you want to do something on it
    - Also note that the SD card can be moved between the Pi 3 and Pi zero with no issues so it is easier to do work on the Pi 3 as it has a full HDMI port and standard USB ports, and then move the card into the zero for flight testing.
- The Pi will not save the current recording when power is cut, so *if you want to be sure you have recorded every second of your flight, wait 60 seconds before cutting the power to the Pi.*
- The Raspberry Pi can be set up as a companion computer (Additional potentially useful information from Ardupilot) to the flight controller. This is likely the best way to get them to work together in the future.
- In case of any issues, a full system image of the Pi's SD card was backed up and shared with the professors