



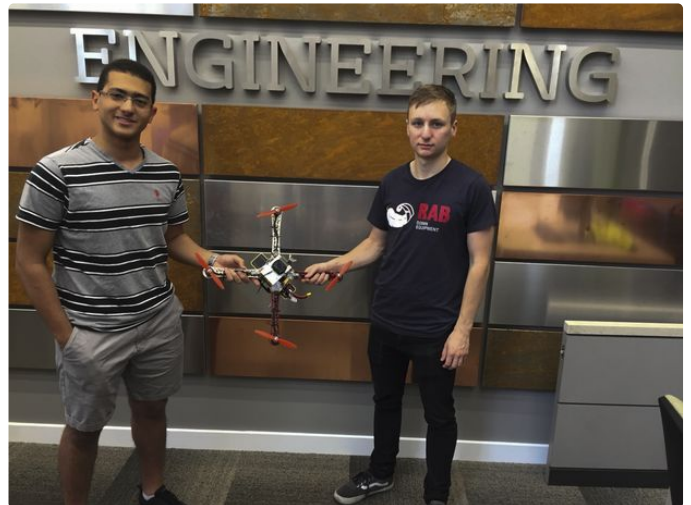
## Autonomous Drone

 by UoBDASAR

In this project you will learn the process of building and configuring a drone, before moving on to investigate autonomous flight using Mission Planner and MATLAB.

Please note this instructable is intended as guidance only. Using drones can be very dangerous around people and can get you in serious trouble with the law

if used inappropriately or in the wrong place. Ensure you adhere to all laws and regulations surround the use of drones. Furthermore, the codes provided on GitHub have not been fully tested so ensure you have other failsafes in place to avoid losing or damaging your drone.



---

### Step 1: Parts List

For this project you will need several parts. Before continuing with the rest of this project, make sure to buy the following components and download the files to 3D print and laser cut the custom parts.

## **Purchased Parts**

<https://www.unmannedtechshop.co.uk/rs2205-s-races-...>

Propellers x4: Gemfan Carbon/Nylon 5030

[https://hobbyking.com/en\\_us/gemfan-propeller-5x3-...](https://hobbyking.com/en_us/gemfan-propeller-5x3-...)

ESCs x4: Little Bee 20A 2-4S

[https://hobbyking.com/en\\_us/favourite-little-bee-...](https://hobbyking.com/en_us/favourite-little-bee-...)

Flight Controller: Navio 2 (with GPS/GNSS antenna and power module) <https://www.unmannedtechshop.co.uk/navio2-autopil...>

Raspberry Pi 3B

<https://thepihut.com/collections/raspberry-pi/pro...>

Transmitter: FRSKY TARANIS X9D+

<https://www.unmannedtechshop.co.uk/frsky-taranis-...>

Receiver: FrSky XSR 2.4 Ghz ACCST

[https://hobbyking.com/en\\_us/xsr-eu-lbt.html?\\_\\_st...](https://hobbyking.com/en_us/xsr-eu-lbt.html?__st...)

Batteries: TATTU 1800mAh 14.8V 45C 4S1P Lipo Battery Pack

<https://www.unmannedtechshop.co.uk/tattu-1800mah-...>

Battery Charger: Turnigy Accucell-6 50W 6A Balancer/Charger

[https://hobbyking.com/en\\_us/turnigy-accucell-6-50...](https://hobbyking.com/en_us/turnigy-accucell-6-50...)

Power Supply for Charger: RS 12V DC Power Supply

Frame: DJI F450 Flame Wheel

<https://www.buildyourowndrone.co.uk/dji-f450-flam...>

PDB: Matek PDB-XT60

<https://www.unmannedtechshop.co.uk/matek-pdb-xt60...>

Motors x4: Emax 2205s 2300kv

rel="nofollow"><https://www.amazon.co.uk/gp/product/B013ZPUXZS/re...>

WiFi Router: TP-LINK TL-WR802N

<https://www.amazon.co.uk/TP-LINK-TL-WR802N-Wir-ele...>

Micro SD Card: SanDisk 32GB

<https://www.amazon.co.uk/SanDisk-microSDHC-Me-mory...>

Standoffs/Spacers: Nylon M2.5 Thread

<https://thepihut.com/products/adafruit-black-nylon...>

Laptop

Cable Ties

Velcro Strap

Heat Shrink

## **3D Printed Parts**

Raspberry Pi / Navio 2 Case (Top and Bottom)


Battery Case (Box and Lid)

## **Laser Cut Parts**

Electronics Layers x2

## Battery Bags: Hobby King Lithium Polymer Charge Pack

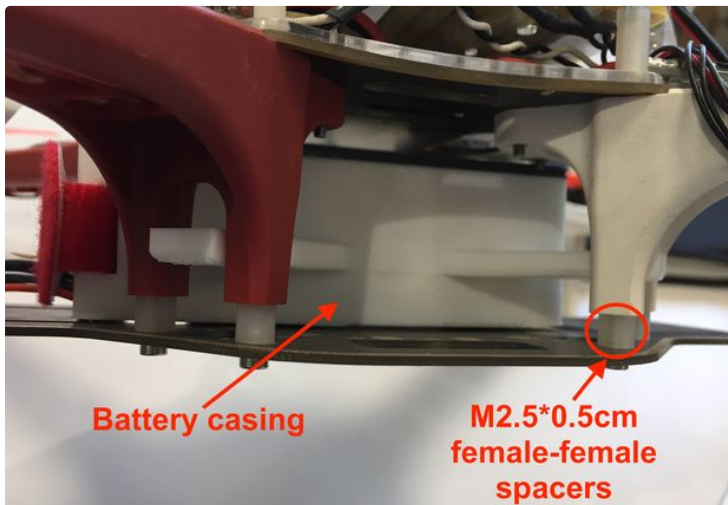
## Banana Connectors

	<a href="https://www.instructabl...">https://www.instructabl...</a>	<a href="#">Download</a>
	<a href="https://www.instructabl...">https://www.instructabl...</a>	<a href="#">Download</a>
	<a href="https://www.instructabl...">https://www.instructabl...</a>	<a href="#">Download</a>
	<a href="https://www.instructabl...">https://www.instructabl...</a>	<a href="#">Download</a>
	<a href="https://www.instructabl...">https://www.instructabl...</a>	<a href="#">Download</a>

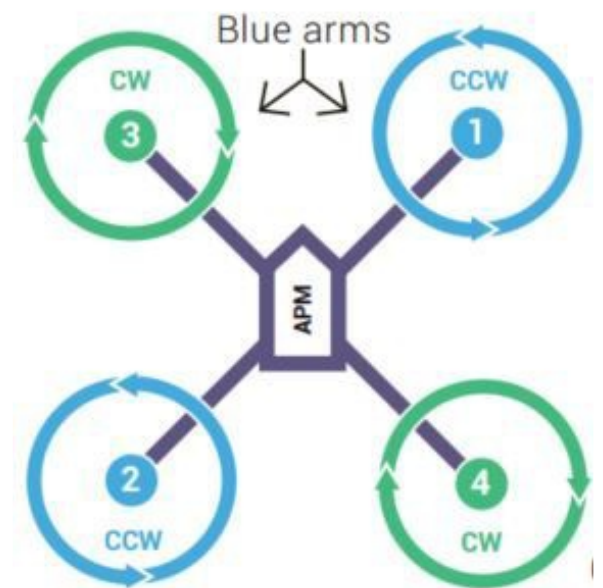
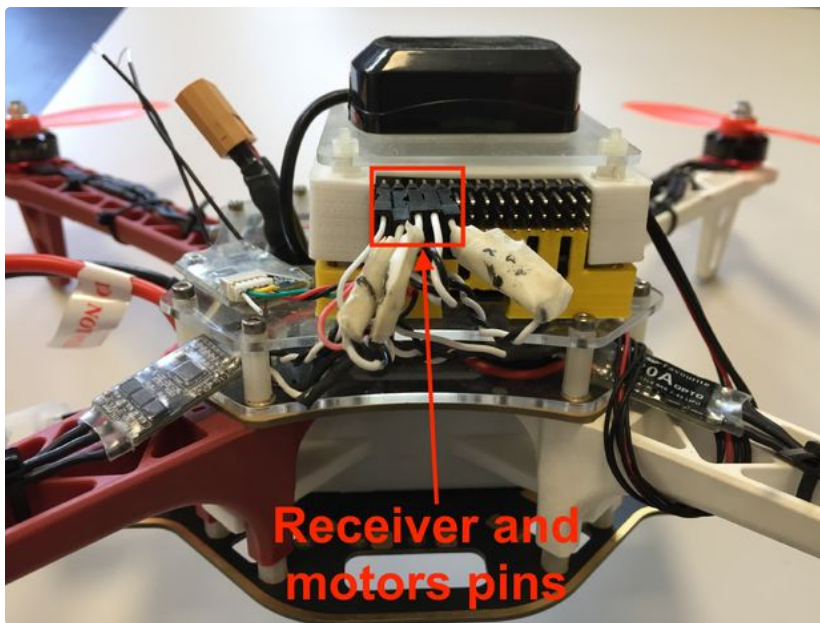
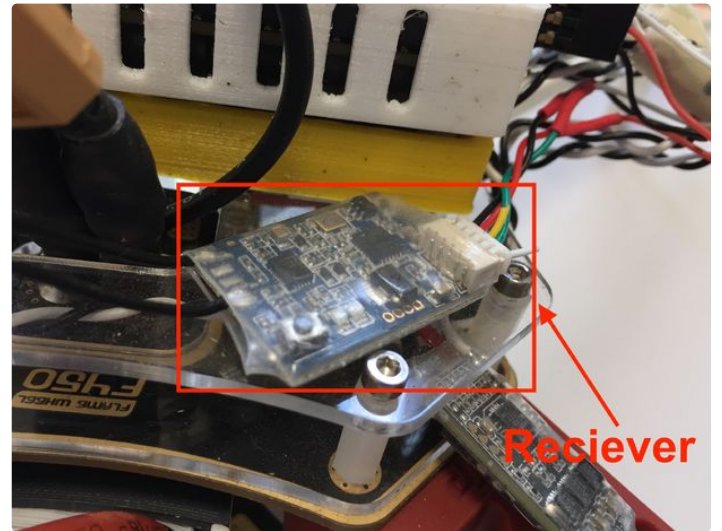
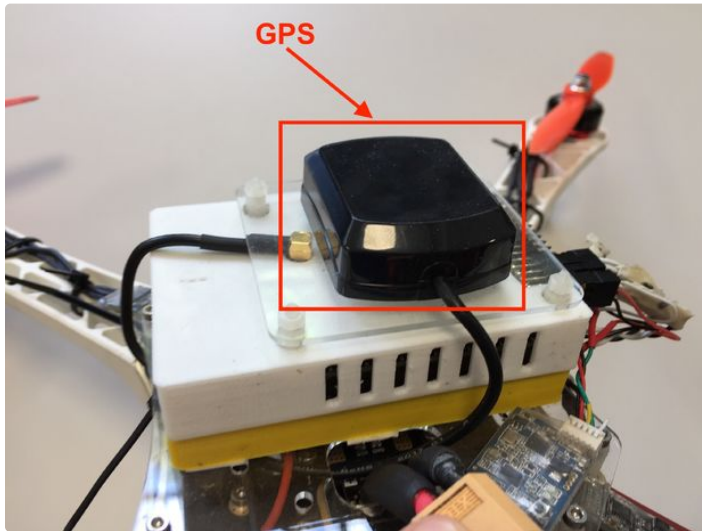
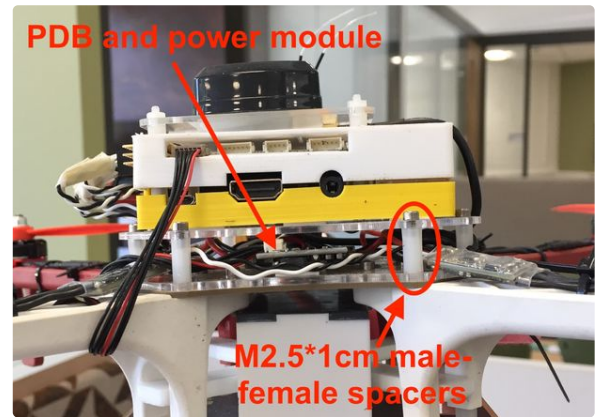
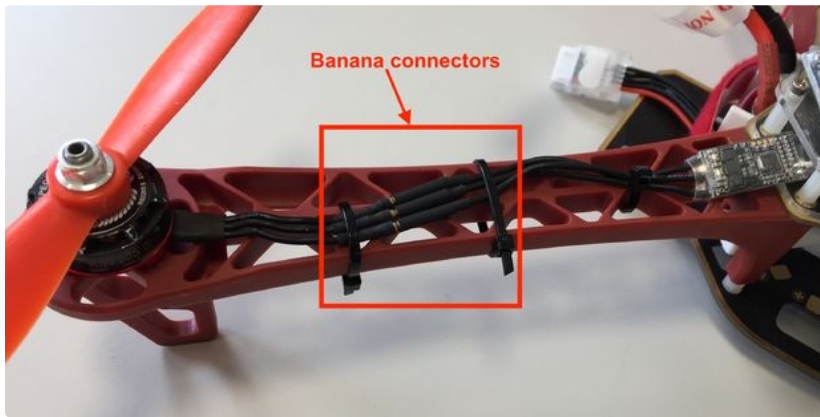
## Step 2: Hardware

### Hardware and Building Phase:

1. Assemble the F450 quadrotor frame and the printed battery casing in the middle (make sure to add the M2.5\*5mm spacers)
2. Attach the motors to the frame.
3. Solder the banana connectors to the ESCs and motors wires.
4. Solder the ESCs and the power module to the PDB. Note: Make sure **not** to use the 5V output of the PDB (it will not provide enough power).
5. Add the first laser cut layer to the top of the F450 frame using M2.5\*10mm male-female spacers; and attach the PDB and power module to this layer. Note: make sure to place the components such that the wires are long enough reach to all of the motors.
6. Connect the ESCs to the motors and use zip ties to secure the wires on to the frame.
7. Attach the Navio2 to the Raspberry Pi and place it in the printed casing.
8. Add the second laser cut layer on top of the first layer and attach the Raspberry-Navio casing using double sided sticky pads.
9. The GPS can be glued on top of the casing, however here it has been placed on another third layer that goes on top of the Raspberry-Navio casing as shown in the pictures, but it is entirely up to the person who is building it. Then simply connect the GPS to the Navio.
10. Fix the receiver on top of the second layer using double sided sticky pads. Connect the ESCs and receiver wires to the Navio pins. The receiver occupies the first column of pins and then the motors occupy the next four columns. Note: The front of the drone is determined by which motor is attached first. Whichever front direction you choose make sure that the motors are connected in the picture at the start of this step.
11. Add propellers. It is advised to leave the propellers to the very end i.e. after finishing the software section and always make sure you take safety precautions when the propellers are on just in case things go wrong.







## Step 3: Software

Software Phase: ([Reference Navio2 docs](#))

1. Get the latest Emlid Raspbian Image from Navio2 docs.
2. Download, extract and run Etcher with administrator rights.
3. Select the archive file with image and sd card drive letter.
4. Click "Flash!". The process may take a few minutes. ([Example video](#))
5. Now to configure the WiFi access we need to edit the **wpa\_supplicant.conf** file located on the SD card. Edit it to make it look like the first picture at the top of this step. Note: the ssid is the name of TP-Link as it appears in your computer. The best way of finding the exact ssid for your TP-Link is to connect your laptop to the TP-Link and then run the command below on a terminal window:

For windows: `netsh wlan show profiles`

For mac: `defaults read /Library/Preferences/SystemConfiguration/com.apple.airport.preferences |grep SSIDString`

psk is the password given on the card that comes with the TP-Link.

1. Eject the SD card and put it in the Raspberry Pi and power it.
2. To check if the Raspberry Pi is connected to the TP-Link you can use any of the apps available that show all devices connected to your network.
3. It is necessary to set fixed IP addresses to devices connected to your TP-Link so that you do not need to change the IP addresses on the codes you write every time. You can simply do so by opening [tplinkwifi.net](#) (while you are connected to the TP-Link of course). Enter the Username: admin and Password: admin. Go to the "DHCP" in the menu on the left of the screen then select "Address Reservation" from the dropdown menu. Add the MAC Addresses of the devices you want to assign the IP addresses for. Here the ground station (Laptop) has been assigned an IP address of 192.168.0.110 and the Raspberry Pi 192.168.0.111.
4. Now we need to download MAVProxy from the following [link](#).
5. Now create a .bat file that looks like the second picture at the top of this step, and make sure that you use the file path where your mavproxy.exe is saved on your laptop. You will need to run this file (by double clicking it) every time you want connect to your drone.
6. In order to get the Raspberry Pi to communicate with MAVProxy a file has to be edited on the Pi.
7. Type `sudo nano /etc/default/arducopter` into the Linux terminal of the Raspberry Pi hosting the Navio2 autopilot.
8. The top line of the file that opens should read `TELEM1="-A udp:127.0.0.1:14550"`. This needs to be changed so that it points to the IP address of your PC.
9. Install Mission Planner and move to the First Time Setup section.



```
1 echo off
2 "C:\Program Files (x86)\MAVProxy\mavproxy.exe" --master=udp:0.0.0.0:14550 --out=tcpin:0.0.0.0:14551 --out=tcpin:0.0.0.0:14552
3 pause
```

## Step 4: First Time Setup

To connect to your UAV follow this procedure:

1. Run both your MAVProxy .bat file, and Mission Planner.
2. Connect the battery to your UAV and wait approximately 30-60 seconds. This will give it time to connect to the wireless network.
3. Click the connect button in the top right of Mission Planner. In the first dialog box that appears type 127.0.0.1 and click OK. In the next box type the port number 14551 and click OK. After a few seconds Mission Planner should connect to your MAV and start displaying telemetry data in the left panel.

When you set up your UAV for the first time, it is necessary to configure and calibrate certain hardware components. The ArduCopter docs have a thorough [guide](#) on how to configure the frame type, compass calibration, radio control calibration, accelerometer calibration, rc transmitter mode setup, ESC calibration, and motor range configuration.

Depending on how you have mounted your Raspberry Pi on the drone it may be necessary to change the board orientation in mission planner. This can be done by adjusting the Board Orientation (AHRS\_ORIENTATION) parameter in the advanced parameters list under the Config/Tuning tab in Mission Planner.

---



## Step 5: First Flight

Once the hardware and software is ready, it's time to prepare for the first flight. It is recommended that before trying autonomous flight the UAV should be flown manually using the transmitter to get a feel for the handling of the aircraft and to fix any issues that may exist.

The ArduCopter documentation has a very detailed and informative section on your [first flight](#). It discusses the various flight modes that come with ArduCopter and what each of these modes does. For the first flight, stabilise mode is the most appropriate flight mode to use.

ArduCopter has many built in safety features. One of these features is the Pre-Arm Safety checks which prevent the aircraft from arming if any issues are detected. Most of these checks are important in helping reduce the chance of a crash or loss of the aircraft but they may be disabled if necessary.

After the first flight it may be necessary to make some changes. As long as the hardware is fully functioning and has been set up correctly, these changes will primarily be in the form of PID tuning. [This guide](#) has some helpful tips for tuning the quadcopter however in our case, simply slightly reducing the P gain was enough to make the aircraft stable. Once the aircraft is flyable it is possible to use the ArduCopter autotune function. This automatically tunes the PIDs

Arming the motors is when the autopilot applies power to the motors to allow them to spin. Before arming the motors it is essential that the aircraft is in a clear open area, well away from any people or obstacles or in a safe flying arena. It is also very important that nothing is near the propellers, particularly body parts and other things that will be damaged by them. Once everything is clear and the pilot is satisfied that it is safe to start, the motors can be armed. [This page](#) gives a detailed set of instructions on how to arm the aircraft. The only differences between that guide and the Navio2 lie in step 7 of arming and step 2 of disarming. To arm the Navio2, both sticks must be held down and in the centre for a few seconds (see picture). To disarm, both sticks must be held down and to the sides for a few seconds (see picture).

To carry out your first flight, follow [this guide](#).

to provide the quickest response whilst still remaining stable. The [ArduCopter documentation](#) provides a detailed guide of how to perform autotuning.

If you encounter problems in any of these steps, the [troubleshooting guide](#) may be able to help.





---

## Step 6: Autonomous Flight

### Mission Planner

Now that your copter has been tuned and can fly well under manual control, autonomous flight can be investigated.

The easiest way to get into autonomous flight is to use Mission Planner as it contains a large array of things you can do with your aircraft. Autonomous flight in Mission Planner falls into two main categories; pre-planned missions (auto mode), and live missions (guided mode). The flight planner screen in mission planner can be used to plan a flight consisting of waypoints to visit and actions to perform such as taking photos. Waypoints can either be chosen manually, or the auto waypoint tool can be used to generate missions to survey an area. Once a mission has been planned and sent to the drone, the Auto flight mode can be used so that the aircraft will autonomously follow the pre-planned mission. [Here](#) is a handy guide about planning missions.

Guided mode is a way of interactively commanding the UAV to do certain things. This is done by using the actions tab in Mission Planner or by right clicking on the map. The UAV can be commanded to do many things such as takeoff, return to launch, and fly to a chosen location by right-clicking the map at the desired location and selecting Fly To Here.

Failsafes are an important thing to consider during autonomous flight to ensure that if things go wrong, the aircraft is not damaged and people are not injured. Mission Planner has a built in Geo-Fence function which can be used to limit where the UAV can fly and stop it from going too far away or too high. It may be worth considering tethering the UAV to the ground for your first few flights as another backup. Finally, it is important that you have your radio transmitter on and connected to the drone so that if necessary you can switch out of the autonomous flight mode into a manual flight mode such as stabilise or alt-hold so that the UAV can be safely piloted to land.

### MATLAB

Autonomous control using MATLAB is far less simple and requires some prior programming knowledge.

The MATLAB scripts `real_search_polygon` and `real_search` allow you to generate pre-planned missions to search a user-defined polygon. The script `real_search_polygon` plans a path over the user defined polygon whereas the script `real_search` plans a path over the minimum rectangle encompassing the polygon. The steps to do this are as follows:

1. Open Mission Planner and go to the Flight Plan window.
2. Draw a polygon over the desired search area using the polygon tool.
3. Save the polygon as 'search\_area.poly' in the same folder as the MATLAB script.
4. Go to MATLAB and run either `real_search_polygon` or `real_search`. Make sure to choose your desired path width and change the `file_path` on line 7 to the correct directory where you are working.
5. Once the script has run and you are happy with the path generated go back to Mission Planner.
6. Click Load WP File on the right hand side and choose the waypoint file 'search\_waypoints.txt' that you have just created.
7. Click Write WPs on the right hand side to send the waypoints to the drone.
8. Arm the drone and take off either manually or by right-clicking the map and selecting takeoff.
9. Once at a reasonable height change the mode to auto and the drone will start the mission.

10. After the mission is over, click RTL in the actions tab to bring the drone back to the launch site.

The video at the start of this step is a simulation in Mission Planner of the UAV searching an area.

<https://youtu.be/4MHVO2xp9Oo>

## Step 7: Vision

The drone mission is to fly over mountains or wilderness and spot humans or irregular objects and then process that to see if that person needs help. This ideally would be done using an expensive infrared camera. However, due to the high costs of infrared cameras, instead the infrared detection is being resembled by detecting all non-green objects using a normal Pi camera.

1. ssh into the Raspberry Pi
2. First of all we need to install OpenCV on the Raspberry Pi. The following [guide](#) provided by pyimagesearch is one of the best available on the internet.
3. Download the code into the Raspberry Pi from GitHub through this following [link](#). To download the code onto the Raspberry Pi, you can download the file on to your computer and then transfer it to the Raspberry Pi.
4. To run the code, go to the directory where to code is on in the Raspberry Pi and then run the command:

```
python colour_target_detection.py --conf conf.json
```

### CONTINUOUS USE

Every time you restart the raspberry pi you need to run the following commands:

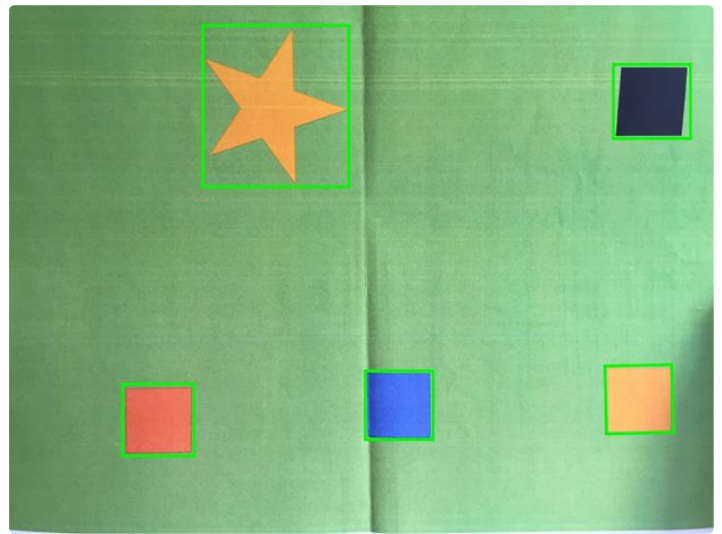
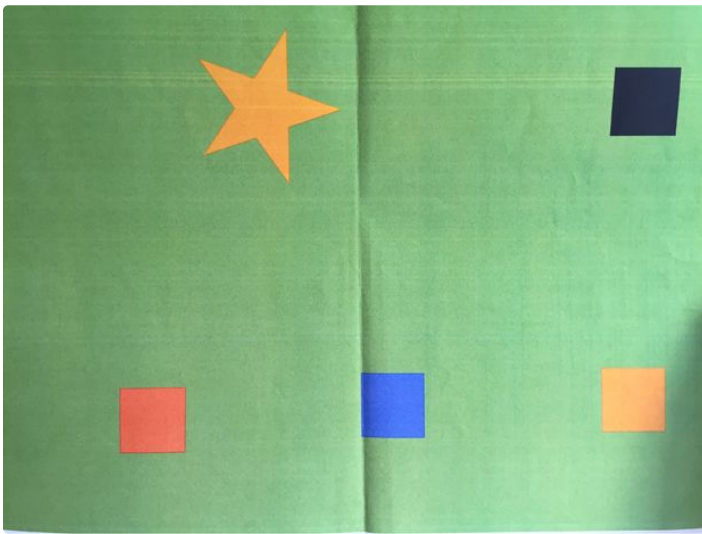
```
sudo ssh pi@192.168.0.111 -X
```

```
source ~/.profile
```

```
workon cv
```

Then continue with step 4 above.

Important note: **NOT** all terminals are capable of showing videos. On mac use the XQuartz terminal.



Great project. Something like this could be really useful for monitoring wildlife.



Or search and rescue. This is a really thorough Instructable!