Politeknika Txorierri

Name of the Author/s: GR Team

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Logo:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Table of Contents **2**

1. Project Title 3

2. Project Category 3

3. Project Description 3

4. Keywords 4

5. Target Audience 4

6. Project Objectives 4

7. Project Timeline 4

8. Resources Provided 5

9. Tools and Technologies Used 5

10. Step-by-Step Implementation Plan 5

11. Evaluation Criteria 9

12. Languages Available 9

13. Contributors 9

14. Feedback and Continuous Improvement 10

Submission Notes **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

1. Project Title:

Raspberry Pi Timelapse Camera

2. Project Category:

AI / IoT / Computer Vision – STEM Maker Project

3. Project Description:

This project demonstrates how to build a time-lapse photography system using a Raspberry Pi and a Pi Camera module.

The system captures images at regular intervals over a long period of time, and later combines these images into a video showing slow processes in accelerated form (time-lapse). Typical examples include plant growth, weather changes, sunsets, construction progress, or scientific experiments.

Using simple commands and scripts, the Raspberry Pi can automatically capture photos every few seconds or minutes and store them as numbered images. These images are later compiled into a video sequence using tools such as mencoder or similar video-processing software.

This project introduces students to embedded systems, automated image capture, scripting, and basic multimedia processing.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Keywords:

Raspberry Pi

timelapse photography

computer vision

automation

camera module

STEM learning

IoT imaging

scientific observation

---

5. Target Audience:

Vocational education and training (VET) students

STEM and robotics clubs

Beginner programmers interested in computer vision

---

6. Project Objectives:

Understand the concept of time-lapse photography

Learn how to use a Raspberry Pi camera module

Program automated image capture scripts

Learn basic Linux command-line operations

Convert sequences of images into video animations

Explore applications of computer vision and environmental monitoring

---

7. Project Timeline:

Estimated duration: 6–8 hours

Phase	Time
Introduction to time-lapse photography	1 hour
Hardware setup	1 hour
Camera configuration	1 hour
Programming image capture	2 hours
Video creation	1 hour
Testing and experimentation	1–2 hours

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Resources Provided:

Raspberry Pi setup instructions

Camera configuration guide

Image capture commands

Video creation scripts

Example time-lapse footage

---

9. Tools and Technologies Used:

Raspberry Pi

Raspberry Pi Camera Module

microSD card

Power supply

Tripod or camera mount

Software

Raspberry Pi OS

Terminal commands (raspistill)

Linux shell scripts

Video tools (e.g., mencoder or ffmpeg)

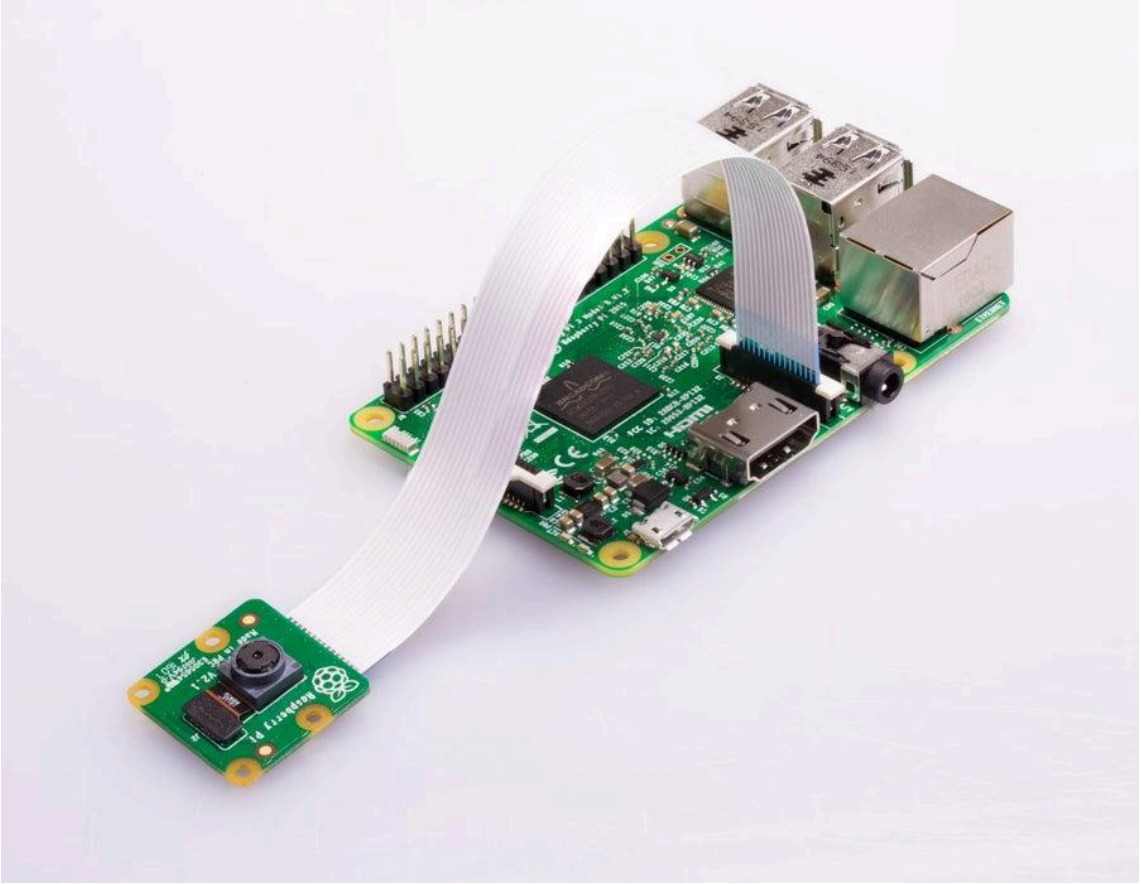
---

10. Step-by-Step Implementation Plan:

Step 1: Preliminal Steps

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



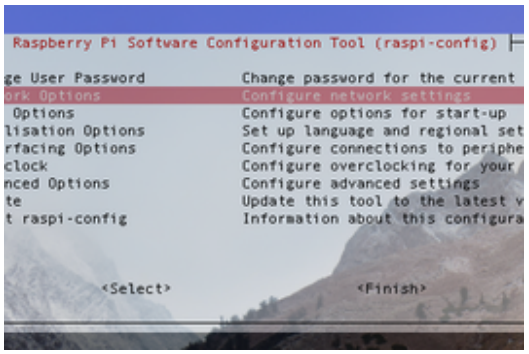
You need only this package on Raspbian Jessie:
sudo apt-get install libav-tools
First we need to connect the RasPi Camera correctly (first image).
Run this command on terminal:
vcgencmd get_camera
Supported and Detected must be 1, or the script does not run.
If Supported is 0, run this command on terminal sudo raspi-config and enable the camera.
If Detected is 0, the camera is not connected to the Raspi.
Step 2: Internet Connection

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

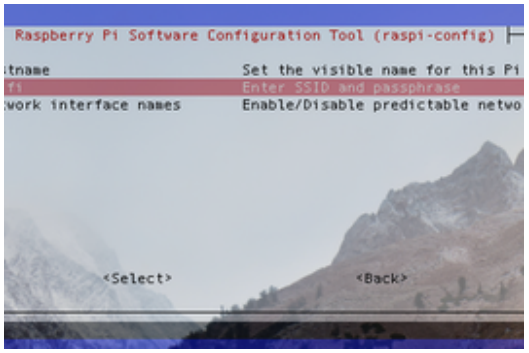






Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



You can decide how to connect your raspberry pi on internet:

1. **Cable**
2. **USB WIFI DONGLE**

Static IP

Why i need to fixed the Static IP?

Every time you connect a device at your network, the router give it a new IP address.

Since you want connect at the some RPi IP, for starting a new timelapse remotely, you now the correct IP.

I remember to you how you can connect at your RPi by SSH command: ssh pi@IP_ADDRESS

Start by editing the dhcpcd.conf file

sudo nano /etc/dhcpcd.conf

If you use the Cable:

interface eth0

static ip_address=192.168.0.10/24

static routers=192.168.0.1

static domain_name_servers=192.168.0.1

If you use the WiFi Dongle:

interface wlan0

static ip_address=192.168.0.10/24

static routers=192.168.0.1

static domain_name_servers=192.168.0.1

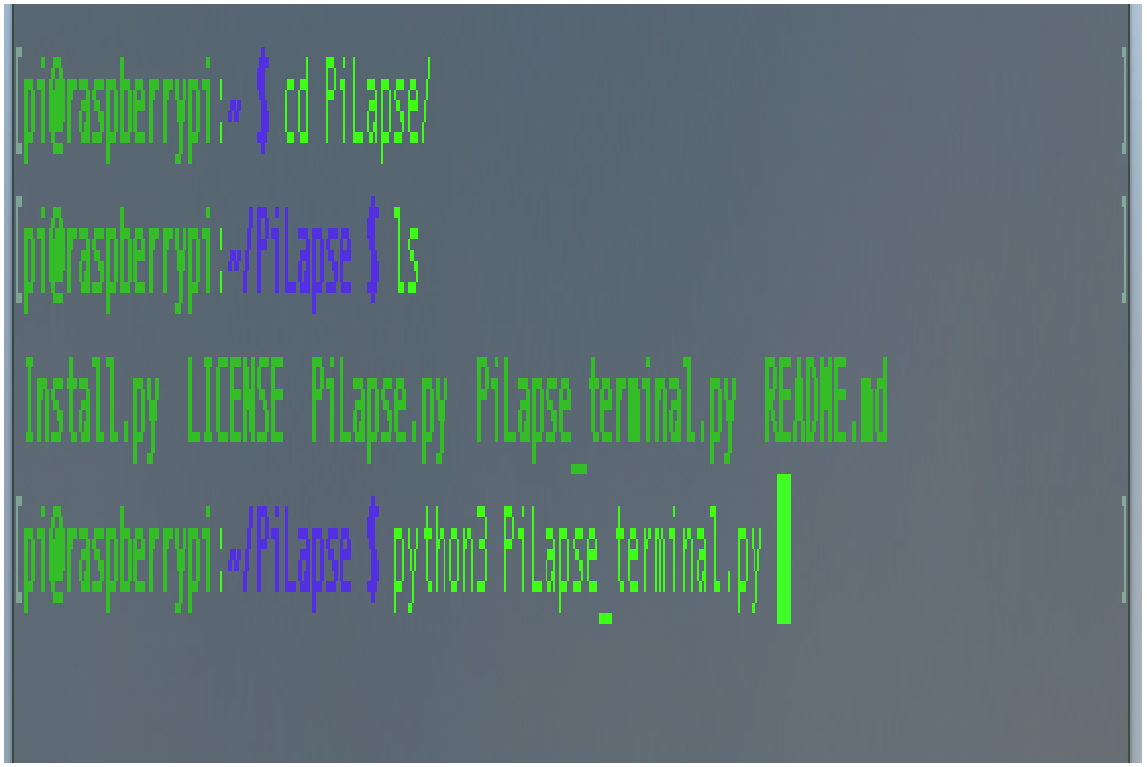
Now using " sudo raspi-config " to connect the WiFi Dongle at your WiFi connection. (3° and 4° images)

Now every time you will disconnect or reboot the RPi, the RPi will connect at the same Ip address: ssh pi@192.168.0.10

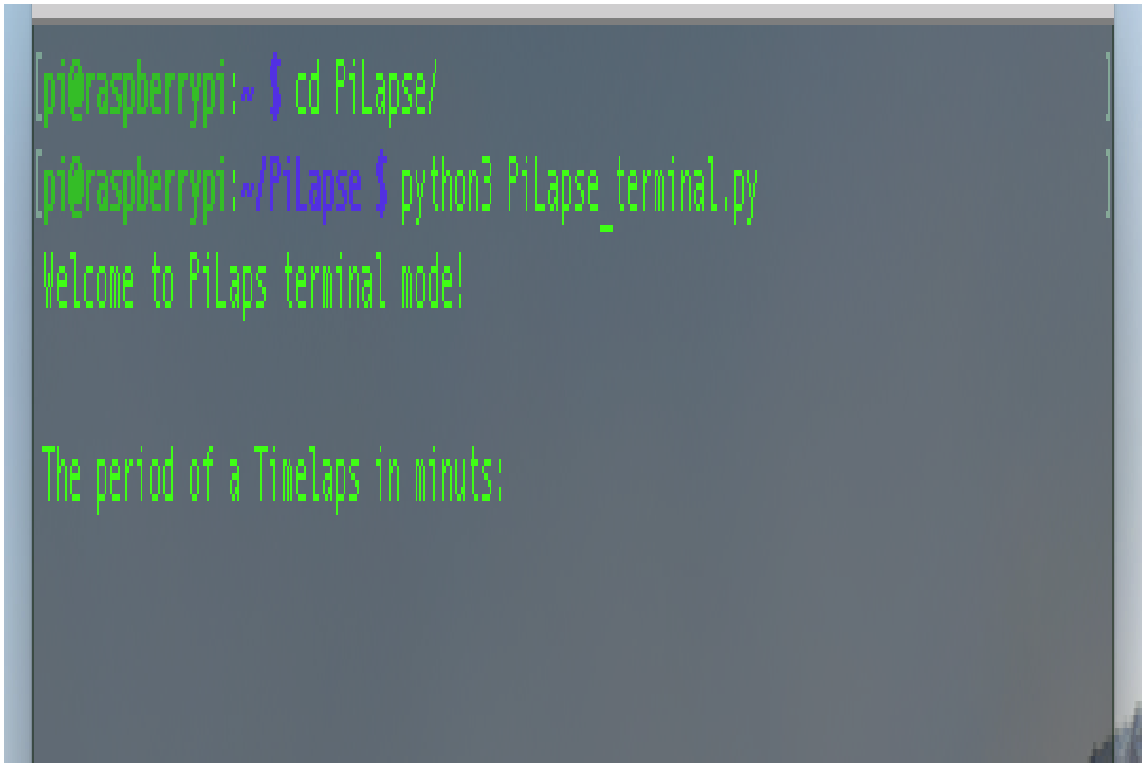
Step 3: Install & Run

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Download the script folder from GitHub or run this command:
git clonehttps://github.com/DaveCalaway/PiLapseInstall
The folder "PiLapse" MUST stay at "/home/pi/" and you can install the script by running the command:
python3 Install.py
The script will start autonomously every time the RPi boot.
Step 4: Operation Mode: Terminal Mode



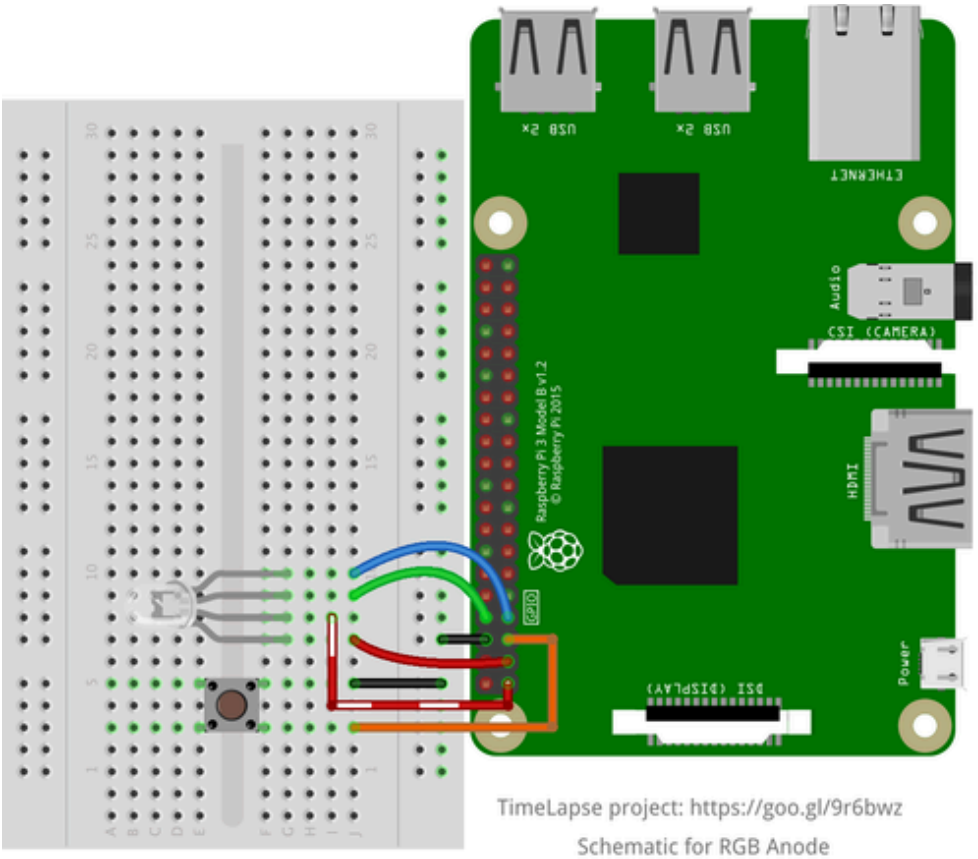
It's time to understand how the Script receive the information for the timelapse.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

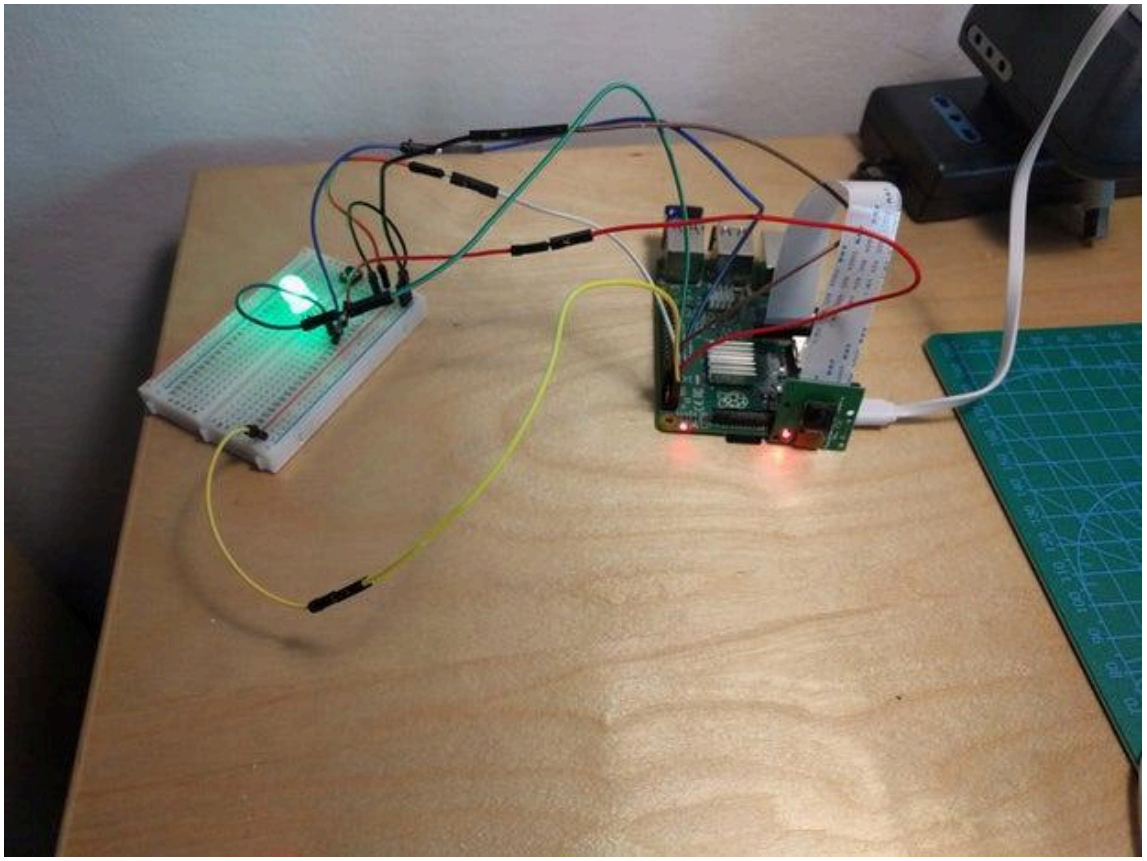
It's very useful to use with the SSH from remote computer.
In the PiLapse's folder, run:
python3 PiLapse_terminal.py
and follow the guide on the monitor.
When the script will finish to capture the images, will create the timelapse video.
Everything images and the timelapse are in the folder called with the name inserted during the "PiLapse_terminal.py" execution.

Step 5: Operation Mode: Button Mode



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



The Daemon version help you to control the TimeLapse with an external button.

The Daemon run continuously.

Single shot

Enabled by pressing the button one time.

Time Lapse

Enabled by hold down the button for 3 or more seconds.

It takes 1 pic every 10 second by default. Press the button again to end the time lapse.

You can change this default period by opening the *PiLapse.py* file and edit the VARIABLES -> freq_button.

It creates a folder called "year-month-day time".

If do you want to use a RGB led, check if it's an Anode or Cathode common!

If it's an Anode common, the code is ok, but if you have an Cathode common, open the PiLapse.py file and edit Anode = 0 in the VARIABLES.

Step 6: DropBox Upload

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



You can upload your photos and timelapse directly on DropBox. Run this comand only the first time:

cd /home/pi/PiLapse/

curl "https://raw.githubusercontent.com/andreafabrizi/Dropbox-Uploader/master/dropbox_uploader.sh" -o dropbox_uploader.sh

chmod +x dropbox_uploader.sh

Now run:

./dropbox_uploader.sh

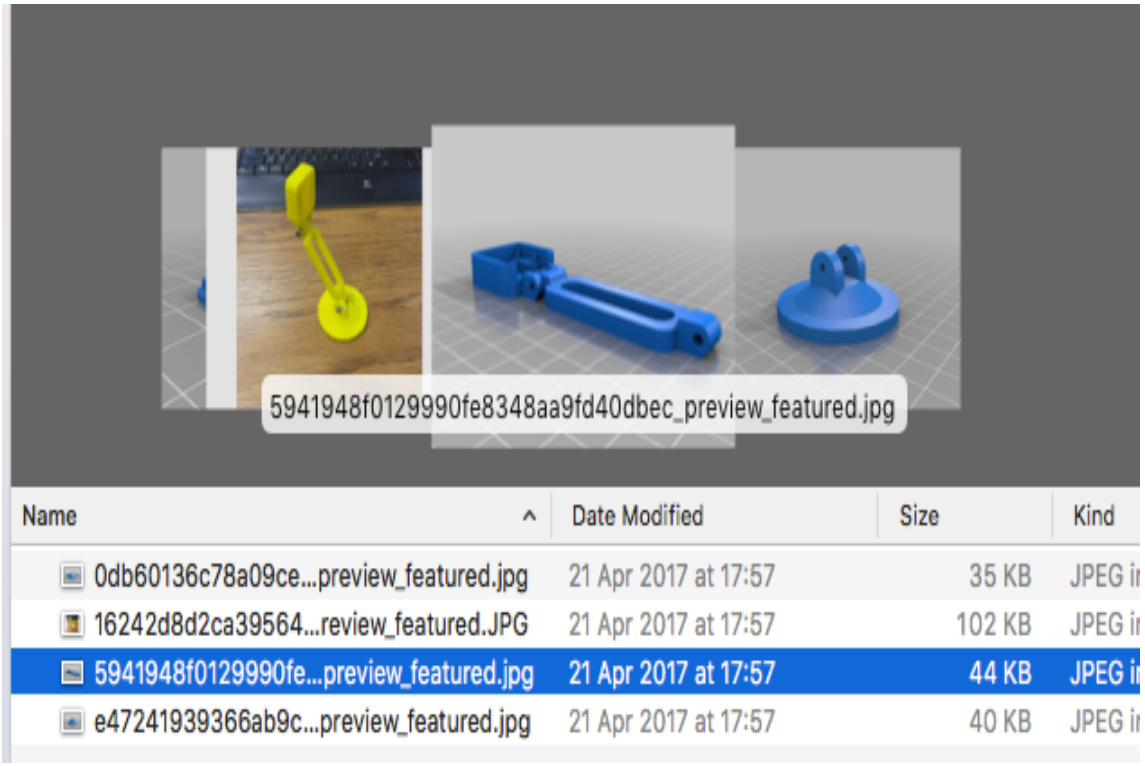
and follow the guide.

You can modified the variables relative to the DropBox Upload by this guide: https://github.com/DaveCalaway/PiLapse#variable-fo...

Step 7: Raspberry Pi Camera Stand

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



For my project i need a stand for the Raspberry Camera.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

11. Evaluation Criteria:

---

12. Languages Available:

---

13. Contributors:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback and Continuous Improvement:

.

Submission Notes:

Please ensure that all required documents and resources are attached to your submission. The AI-SAVES@ProjectHub review team will evaluate the submission based on the provided information. Once approved, the project will be published on the hub, and partners will be notified.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.