

Long-run remote underwater stereo camera systems

Nick Mortimer | 1-Feb-2024







QR Control



GoPro Labs

[View on GitHub](#)

QR Control

Configuration Commands for All Labs Enabled Cameras

- Set camera modes with the [Customizable QR Code Creator](#) and here optimized for **Bones** and as a mobile app:



- Set Precision [Local Date and Time](#) or [UTC Time](#) or
- Set Local Time:



- Personalize your cameras with [Owner Information](#)

Capture Triggers

- Use the image sensor for [Motion Detection](#), capturing only scenes with motion
- Capture using [IMU Motion Triggers](#) for detecting camera movement
- **HERO7/8/9/10/MAX** - Start capture only when traveling fast with [GPS Speed Triggers](#)
- **HERO7/8/9/10/MAX** - Use the [USB Power Trigger](#) to start and stop the camera (pseudo dash-cam mode)
- **HERO9/10/Bones** - Start capture using [Sound Pressure Level Trigger](#)

Extended Time-lapse Features

- Start a 24-hour long time-lapse or greater with [Extra Long Time-lapses](#)
- Use your GoPro to time-lapse [Construction Projects](#)
- Schedule your camera to automatically [Time-lapse Sunsets and Sunrises](#)
- **HERO9/10/Bones** - Daily Time-lapse with [auto upload](#) (GoPro Subscription required)

Video Alteration Features

- **HERO8/9/10/MAX/Bones** - Set the [Max Shutter Angle](#) for control over stabilization in low light



GoPro Labs

[View on GitHub](#)

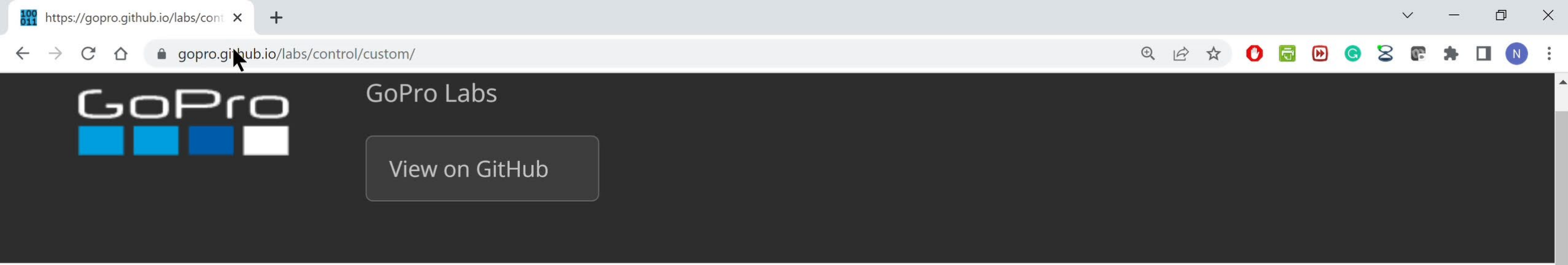
Precision Date and Time (Local)

Simply point your Labs enabled camera at this animated QR Code, to set your date and time very accurately to local time. This is particularly useful for multi-camera shoots, as it helps synchronize the timecode between cameras. As the camera's internal clock will drift slowly over time, use this QR Code just before your multi-camera shoot for the best synchronization.



QR Command: **oT220512080447.18**

Compatibility: Labs enabled HERO5 Session, HERO7, HERO8, HERO9, HERO10, MAX and BONES



GoPro QR Code Creator

Create a custom camera mode, and even start a capture all through QR Codes. This is the fastest way to access many of the new GoPro Labs' firmware features. † indicates the control requires HERO10 Labs 1.20 or better. Many features of this page are also available as a mobile app.



Camera Mode: not set

Video Performancet† Endurancet† Stationary†

Presets †: Standard Activity Cinematic SloMo Basic

VideoTL-Modes: TimeWarp Timelapse Nightlapse

Photo-Modes: Photo Burst Live Burst Night Timelapse Nightlapse



GoProQR:

mVr1080p30fW0te0hS0dR0aScN
d0b0wAi16M4sMag1v0q0dVoV9
B9D0R1C30S3W0L0!M64T=1!S

<https://gopro.github.io/lab/control/set?code=mVr1080p30fW0te0hS0dR0aScNd0b0wAi16M4sMag1v0q0dVoV9B9D0R1C30S3W0L0!M64T=1!S>

WEST COAST
EAGLES



Bandwidth Calculator

Size

16	TB	◆
----	----	---

Speed

100	MB/s	◆
-----	------	---

Time

1.8519	days	◆
--------	------	---

Buy

TL-D800C (USB)

USB 3.2 Gen2 Type-C high-capacity JBOD storage enclosure

▶ Watch Video



[For NAS](#)

[For PC/Server](#)

[Specification](#)

[Supported NAS](#)

[Compatibility](#)

[Downloads](#)

The TL-D800C JBOD storage enclosure allows you to back up and expand your QNAP NAS. The TL-D800C features eight 3.5-inch SATA 6Gb/s drive bays with USB 3.2 Gen2 (10 Gbps) Type-C connectivity to provide a fast and smooth expansion solution.

Note :

1. TL-D800C is not compatible with USB 3.0, 1.1, and 2.0.





SPECIFICATIONS

Workflow Station (Dock and USB miniHub)

Interface

Dock: USB 3.2 Gen 2 USB miniHub: USB 3.2 Gen 1



Write some Code

```
dodo.py > task_create_json
53     command = f'exiftool -api largefilesupport=1 -m -u -q -q -n -CameraSerialNumber -CreateDate -SourceFile -Duration -FileSize -FieldOfView -
54     if file_dep:
55         yield {
56             'name':path,
57             'file_dep':file_dep,
58             'actions':[command],
59             'targets':[target],
60             'uptodate':[run_once],
61             'clean':True,
62         }
63 @create_after(executed='create_json', target_regex='.*\.json')
64 def task_concat_json():
65
66     def concat(dependencies, targets):
67         data = pd.concat([pd.read_json(dep) for dep in dependencies])
68         data['Bad'] = data['CreateDate'].isna()
69         data['SourceFile'] = data.apply(lambda row: os.path.join(row['Directory'], row['FileName']), axis=1)
70         data['Directory'] = data['SourceFile'].apply(lambda row: os.path.dirname(row), axis=1)
71         data['FileName'] = data['SourceFile'].apply(lambda row: os.path.basename(row), axis=1)
72         data[['ItemPath', 'ItemName']] = data[['Directory', 'FileName']].apply(lambda row: os.path.join(row['Directory'], row['FileName']), axis=1)
73         data['ItemPath'] = data['ItemPath'].apply(lambda row: row.replace('\\', '/'), axis=1)
74         data['ItemName'] = data['ItemName'].apply(lambda row: row.replace('.MP4', '.mp4'), axis=1)
75         data['Duration'] = data['Duration'].apply(lambda row: row if row is not None else '%H:%M:%S', axis=1)
76         data['Duration'] = data.groupby('CameraSerialNumber')['Duration'].cumsum()
77
78     @create_after(executed='concat_json', target_regex='.*\.json')
79     def get_urls():
80         url_pattern = os.path.join(cfg['exifstore'], '.*')
81         urls = glob.glob(cfg['exifname'])
82
83     'file_dep':exiffiles,
84     'actions':[concat],
85     'targets':[get_urls('exifstore')],
86     'uptodate':[True],
87     'clean':True,
88 }
89
90 @create_after(executed='concat_json', target_regex='.*\.json')
91 def task_checkbars():
92     def checkbars(dependencies, targets):
93         data = pd.read_csv(dependencies[0], parse_dates=['CreateDate'])
94         data.Duration = pd.to_timedelta(data.Duration, unit='s')
95         data['CreateEnd'] = data.CreateDate + data.Duration
96         data['CreateStart'] = data.CreateDate
97         cstart = data.groupby('CameraSerialNumber')['CreateStart'].min()
98         cend = data.groupby('CameraSerialNumber')['CreateEnd'].max()
99         stats = pd.concat([cstart, cend], axis=1)
100         if os.path.exists(targets[0]):
101             barnumbers = pd.read_csv(targets[0], parse_dates=['StartDate', 'EndDate']).drop(['CreateStart', 'CreateEnd'], axis=1, errors='ignore')
```

EXPLORER

REPO

- __pycache__
- __init__.py
- .doit.db
- .gitignore
- config.yml
- dodo.py
- gopro_dropcam.pipeline.py
- requirements.txt

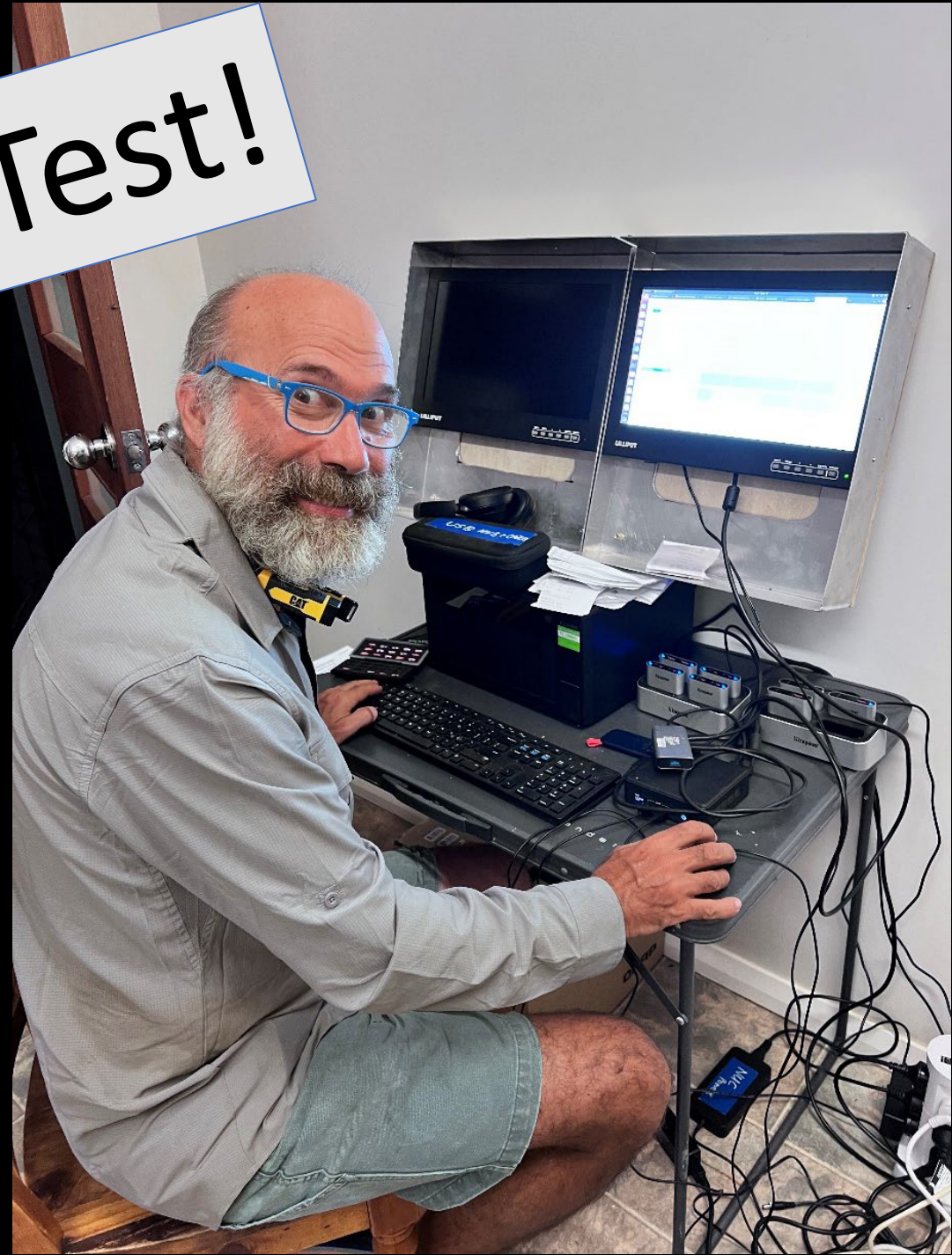
OUTLINE

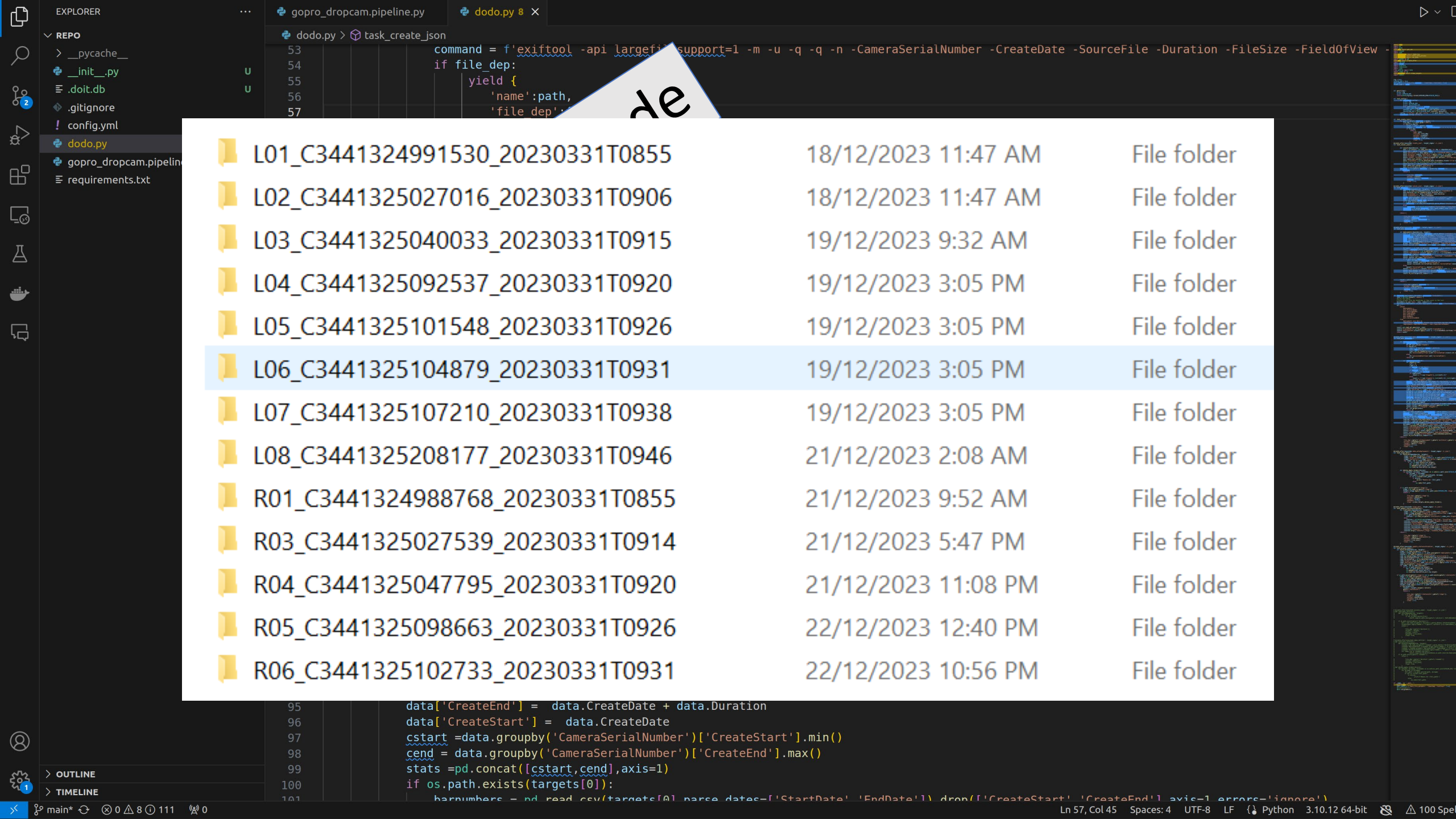
TIMELINE

Terminal

Ln 57, Col 45 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit 100 Spell

Field Test!
















L01_C3441324991530_20230331T0855	18/12/2023 11:47 AM	File folder
L02_C3441325027016_20230331T0906	18/12/2023 11:47 AM	File folder
L03_C3441325040033_20230331T0915	19/12/2023 9:32 AM	File folder
L04_C3441325092537_20230331T0920	19/12/2023 3:05 PM	File folder
L05_C3441325101548_20230331T0926	19/12/2023 3:05 PM	File folder
L06_C3441325104879_20230331T0931	19/12/2023 3:05 PM	File folder
L07_C3441325107210_20230331T0938	19/12/2023 3:05 PM	File folder
L08_C3441325208177_20230331T0946	21/12/2023 2:08 AM	File folder
R01_C3441324988768_20230331T0855	21/12/2023 9:52 AM	File folder
R03_C3441325027539_20230331T0914	21/12/2023 5:47 PM	File folder
R04_C3441325047795_20230331T0920	21/12/2023 11:08 PM	File folder
R05_C3441325098663_20230331T0926	22/12/2023 12:40 PM	File folder
R06_C3441325102733_20230331T0931	22/12/2023 10:56 PM	File folder












```
95 data['CreateEnd'] = data.CreateDate + data.Duration
96 data['CreateStart'] = data.CreateDate
97 cstart = data.groupby('CameraSerialNumber')['CreateStart'].min()
98 cend = data.groupby('CameraSerialNumber')['CreateEnd'].max()
99 stats = pd.concat([cstart, cend], axis=1)
100 if os.path.exists(targets[0]):
101     barnumbers = pd.read_csv(targets[0], parse_dates=['StartDate', 'EndDate'], drop(['CreateStart', 'CreateEnd'], axis=1, errors='ignore'))
```

Write more Code

StageId	StartTime	FinishTime	Collection	Station	Operation	Latitude	Longitude	Depth	VideoStart	VideoEnd	Frame	CameraTime
CSIRO-SB12_20230404T135228									4/04/2023 13:52	5/04/2023 0:29	CSIRO-SB12	20230404T135228
CSIRO-SB09_20230420T085406									20/04/2023 8:54	#####	CSIRO-SB09	20230420T085406
CSIRO-SB09_20230420T085457									20/04/2023 8:54	#####	CSIRO-SB09	20230420T085457
CSIRO-SB14_20230420T090314									20/04/2023 9:03	#####	CSIRO-SB14	20230420T090314
CSIRO-SB14_20230420T090535									20/04/2023 9:05	#####	CSIRO-SB14	20230420T090535
CSIRO-SB10_20230420T090939									20/04/2023 9:08	#####	CSIRO-SB10	20230420T090939
CSIRO-SB11_20230420T091441									20/04/2023 9:14	#####	CSIRO-SB11	20230420T091441
CSIRO-SB12_20230420T091736									20/04/2023 9:17	#####	CSIRO-SB12	20230420T091736
CSIRO-SB16_20230420T092023									20/04/2023 9:20	#####	CSIRO-SB16	20230420T092023
CSIRO-SB10_20230423T101459									23/04/2023 10:14	#####	CSIRO-SB10	20230423T101459
CSIRO-SB08_20230423T104607									23/04/2023 10:45	#####	CSIRO-SB08	20230423T104607
CSIRO-SB13_20230423T105309									23/04/2023 10:53	#####	CSIRO-SB13	20230423T105309
CSIRO-SB02_20230423T105700									23/04/2023 10:57	#####	CSIRO-SB02	20230423T105700
CSIRO-SB11_20230423T110835									23/04/2023 11:08	#####	CSIRO-SB11	20230423T110835
CSIRO-SB12_20230423T111511									23/04/2023 11:15	#####	CSIRO-SB12	20230423T111511
CSIRO-SB06_20230423T112100									23/04/2023 11:21	#####	CSIRO-SB06	20230423T112100
CSIRO-SB05_20230423T112838									23/04/2023 11:27	#####	CSIRO-SB05	20230423T112838
CSIRO-SB14_20230423T113350									23/04/2023 11:33	#####	CSIRO-SB14	20230423T113350

```
data['CreateEnd'] = data.CreateDate + data.Duration
data['CreateStart'] = data.CreateDate
cstart = data.groupby('CameraSerialNumber')['CreateStart'].min()
cend = data.groupby('CameraSerialNumber')['CreateEnd'].max()
stats = pd.concat([cstart, cend], axis=1)
if os.path.exists(targets[0]):
    barnumbers = pd.read_csv(targets[0], parse_dates=['StartDate', 'EndDate'], drop(['CreateStart', 'CreateEnd'], axis=1, errors='ignore'))
```


 BOL-69417_20230329T1230_BAR-11	27/01/2024 2:55 PM
 BOL-69418_20230329T1212_BAR-10	27/01/2024 4:04 PM
 BOL-69419_20230329T1154_BAR-09	27/01/2024 5:15 PM
 BOL-69420_20230329T1136_BAR-16	27/01/2024 1:45 PM
 BOL-69426_20230329T1113_BAR-14	27/01/2024 6:24 PM
 BOL-69427_20230329T1058_BAR-13	27/01/2024 7:37 PM
 BOL-69430_20230329T1424_BAR-02	27/01/2024 8:56 PM
 BOL-69747_20230329T1440_BAR-03	27/01/2024 11:04 PM
 BOL-69760_20230329T1403_BAR-01	27/01/2024 11:58 PM
 BOL-70288_20230329T1504_BAR-04	28/01/2024 12:57 AM
 BOL-70547_20230329T1539_BAR-05	28/01/2024 2:54 AM

-  BOL-69420_20230329T1136_BAR-L16_C3441325506299_0002_01.MP4
-  BOL-69420_20230329T1147_BAR-L15_C3441325495269_0004_10.MP4
-  BOL-69420_20230329T1147_BAR-L15_C3441325495269_0044_01.MP4
-  BOL-69420_20230329T1147_BAR-L15_C3441325495269_0044_02.MP4
-  BOL-69420_20230329T1211_BAR-L15_C3441325495269_0044_03.MP4
-  BOL-69420_20230329T1211_BAR-L16_C3441325506299_0002_02.MP4
-  BOL-69420_20230329T1235_BAR-L15_C3441325495269_0044_04.MP4
-  BOL-69420_20230329T1235_BAR-L15_C3441325495269_0044_05.MP4
-  BOL-69420_20230329T1246_BAR-L15_C3441325495269_0044_06.MP4
-  BOL-69420_20230329T1246_BAR-L16_C3441325506299_0002_03.MP4
-  BOL-69420_20230329T1258_BAR-L15_C3441325495269_0044_07.MP4

Over 9000 videos

Over 100Tb of video data

Ready for analysis

Thank you



<https://github.com/NickMortimer/turtle>

Photo Cindy Bessy!