# 기타 연계 프로그램

기타 연계 프로그램

## 캡쳐

```
import os
import cv2
import datetime, time
from pathlib import Path

archive_path = os.path.join(os.path.dirname(os.path.realpath(__file__)), 'archive')

def capture_and_save(im):
    s = im.shape
    # Add a timestamp
    font = cv2.FONT_HERSHEY_SIMPLEX
    bottomLeftCornerOfText = (10,s[0]-10)
    fontScale = 1
    fontColor = (20,20,20)
    lineType = 2

    cv2.putText(im,datetime.datetime.now().isoformat().split(".")
[0],bottomLeftCornerOfText,font,fontScale,fontColor, lineType)

    m = 0
    p = Path(archive_path)
    for imp in p.iterdir():
        if imp.suffix == ".png" and imp.stem != "last":
            num = imp.stem.split("_")[1]
            try:
                num = int(num)
                if num>m:
                    m = num
            except:
                print("Error reading image number for",str(imp))
    m +=1
    lp = Path(archive_path + "/last.png")
    if lp.exists() and lp.is_file():
        np = Path(archive_path + "/img_{}.png".format(m))
        np.write_bytes(lp.read_bytes())
    cv2.imwrite(archive_path + "/last.png",im)

if __name__=="__main__":
    capture_and_save()
    print("done")
```

# 환경 설정

```
from pathlib import Path

p = Path("logs")
if not p.exists():
    p.mkdir()

dictConfig = {
    'version': 1,
    'disable_existing_loggers': True,
    'formatters': {
        'standard': {
            'format': '%(asctime)s [%(levelname)s] %(name)s:: %(message)s',
        },
    },
    'handlers': {
        'default': {
            'level': 'DEBUG',
            'formatter': 'standard',
            'class': 'logging.StreamHandler',
            'stream': 'ext://sys.stdout',
        },
        'file': {
            'class': 'logging.handlers.RotatingFileHandler',
            'level': 'DEBUG',
            'formatter': 'standard',
            'filename': 'logs/logfile.log',
            'mode': 'a',
            'maxBytes': 5_242_880,
            'backupCount': 3,
            'encoding': 'utf-8',
        },
    },
    'loggers': {
        '__main__': {
            'handlers': ['default','file'],
            'level': 'DEBUG',
            'propagate': False,
        },
        'camera': {
            'handlers': ['default', 'file'],
            'level': 'DEBUG',
            'propagate': False,
        },
    }
}
```

## 싱크로나이징

```
import os
import datetime

class Lock_Manager:
        def __init__(self, name):
                self.locks = os.path.dirname(os.path.realpath(__file__)) + "/locks"
                self.name = name + ".lock"
                self.path = os.path.join(self.locks, self.name)

                if not os.path.exists(self.locks):
                        os.makedirs(self.locks)

        def set(self):
                if not os.path.exists(self.path):
                        print("Setting {} lock".format(self.name))
                        timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")

                        with open(self.path, 'w+') as f:
                                f.write(timestamp)

        def remove(self):
                if os.path.isfile(self.path):
                        os.remove(self.path)

        def get_lock_time(self):
                for filename in os.listdir(self.locks):
                        with open(self.locks + "/" + filename) as f:
                                return f.read()

                return None
```

## 파워정보 조회

```
from vcgencmd import Vcgencmd

vcgm = Vcgencmd()

class PowerStatus(object):
    def __init__(self):
        print("> vcgm.version()=" + vcgm.version())

    def getPowerStatus(self):
        output=vcgm.get_throttled()
        print("raw_data=" + output['raw_data'])
        print("binary=" + output['binary'])
        return "raw_data=" + output['raw_data'] + ", binary=" + output['binary']
```

# Data Util

```
from datetime import datetime

class Util:
        def log(source, msg):
                print(datetime.now().strftime("%Y%m%d_%H%M%S") + " | " + source + " |
" + str(msg))
```

---

Revision #1
★Created 10 June 2023 13:09:33 by Hyeon Su Ryu
Updated 10 June 2023 13:24:51 by Hyeon Su Ryu