



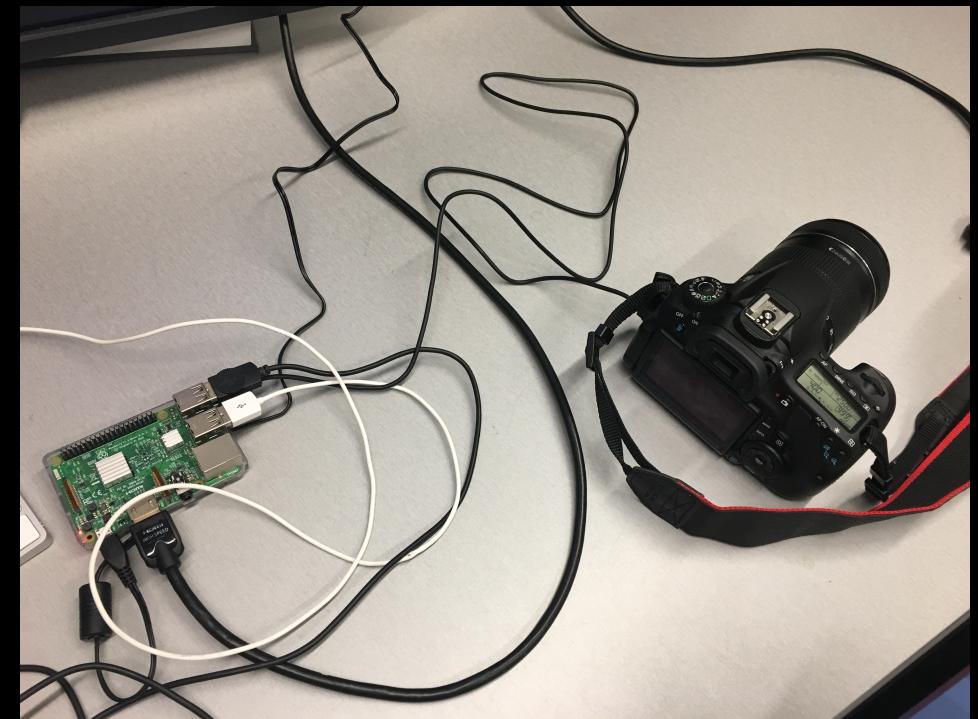
USING IOS APP TO DO SELF-DEFINED LAPSE SHOOTING

Jing Si

jingsi@gwu.edu

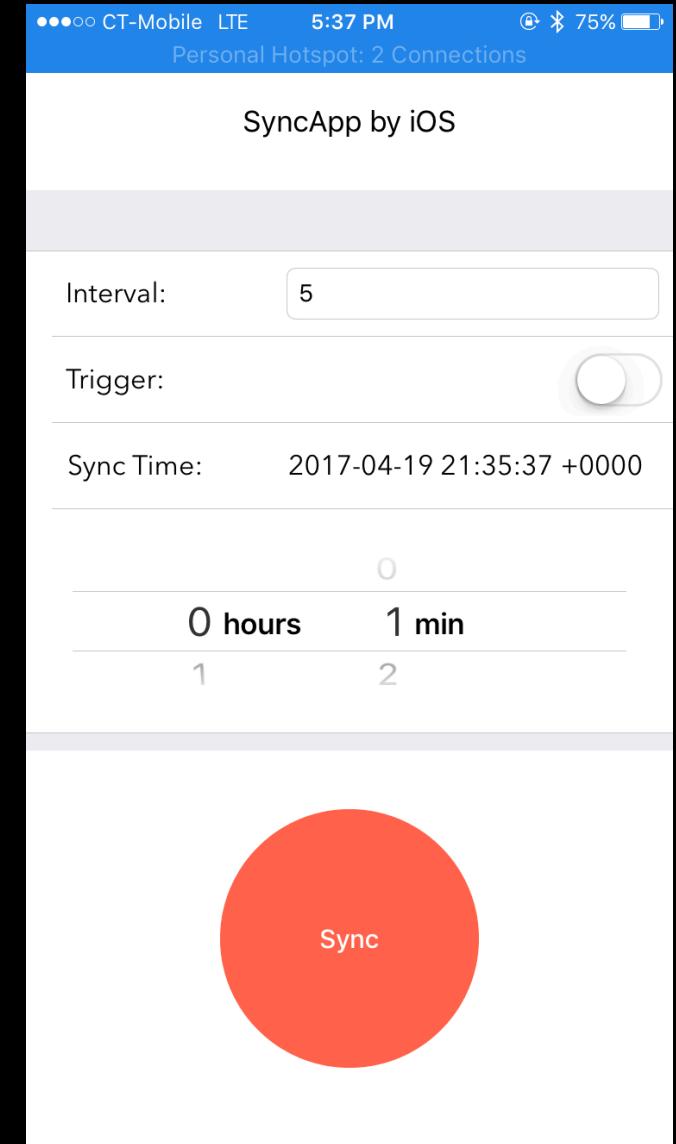
PART 1 CAMERA & RASPBERRY SET UP

- Assembly the basic parts
- Using an USB cable to connect the camera and the raspberry pi



PART 2 DEVELOP AN IOS APP

- this app is used to control the shooting process
- The interval is the sleep time between two pictures
- The time wheel is to set the entire time of the process, and the longest time would be 12hs and 59mins
- When the trigger is ON, the process will began, at this time, user can change the variables as they want, and will not break the process.
- The app intend to change more variables, however, they haven't been added.



REAL-TIME DATABASE

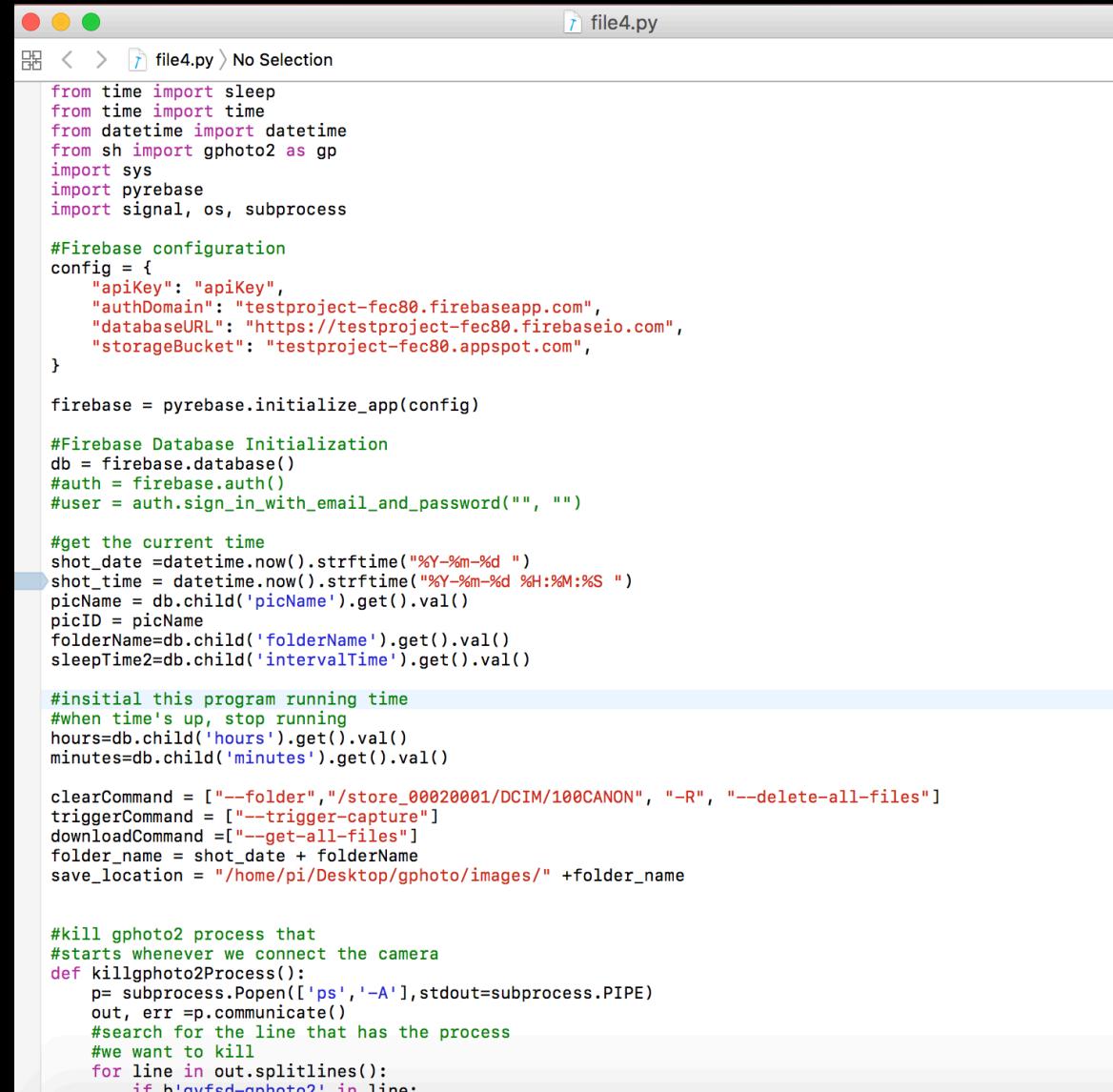
- Using Google's real time database Firebase
- User can change parameters at any time during the shooting process
- without breaking the process

The screenshot shows the Firebase Realtime Database console. At the top, there is a navigation bar with the Firebase logo, the project name 'TestProject', and links for 'Overview', 'Analytics', 'Authentication', 'Database' (which is selected), 'Storage', 'Hosting', 'Functions', 'Test Lab', and 'Crash Reporting'. Below the navigation bar, there are tabs for 'DATA', 'RULES', 'BACKUPS', and 'USAGE'. The main area is titled 'Realtime Database' and shows the URL 'https://testproject-fec80.firebaseio.com/'. A red warning message states: '⚠ Your security rules are defined as public, anyone can read or write to your database' with 'LEARN MORE' and 'DISMISS' buttons. The database structure under 'testproject-fec80' is displayed as follows:

```
testproject-fec80
  - folderName: "setFolderName"
  - hours: 0
  - interval: 5
  - minutes: 1
  - picName: "IMAGE"
  - trigger
    - state: "OFF"
```

PART 3 RASPBERRY AS CONTROLLER

- Using gphoto2 to control the trigger and manage the photos
- Like, trigger capture, download photos from camera to local directory
- Create new folder and rename the photo by users



```
file4.py > No Selection
from time import sleep
from time import time
from datetime import datetime
from sh import gphoto2 as gp
import sys
import pyrebase
import signal, os, subprocess

#Firebase configuration
config = {
    "apiKey": "apiKey",
    "authDomain": "testproject-fec80.firebaseio.com",
    "databaseURL": "https://testproject-fec80.firebaseio.com",
    "storageBucket": "testproject-fec80.appspot.com",
}

firebase = pyrebase.initialize_app(config)

#Firebase Database Initialization
db = firebase.database()
#auth = firebase.auth()
#user = auth.sign_in_with_email_and_password("", "")

#get the current time
shot_date = datetime.now().strftime("%Y-%m-%d ")
shot_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S ")
picName = db.child('picName').get().val()
picID = picName
folderName=db.child('folderName').get().val()
sleepTime2=db.child('intervalTime').get().val()

#initial this program running time
#when time's up, stop running
hours=db.child('hours').get().val()
minutes=db.child('minutes').get().val()

clearCommand = ["--folder","/store_00020001/DCIM/100CANON", "-R", "--delete-all-files"]
triggerCommand = ["--trigger-capture"]
downloadCommand =[ "--get-all-files"]
folder_name = shot_date + folderName
save_location = "/home/pi/Desktop/gphoto/images/" +folder_name

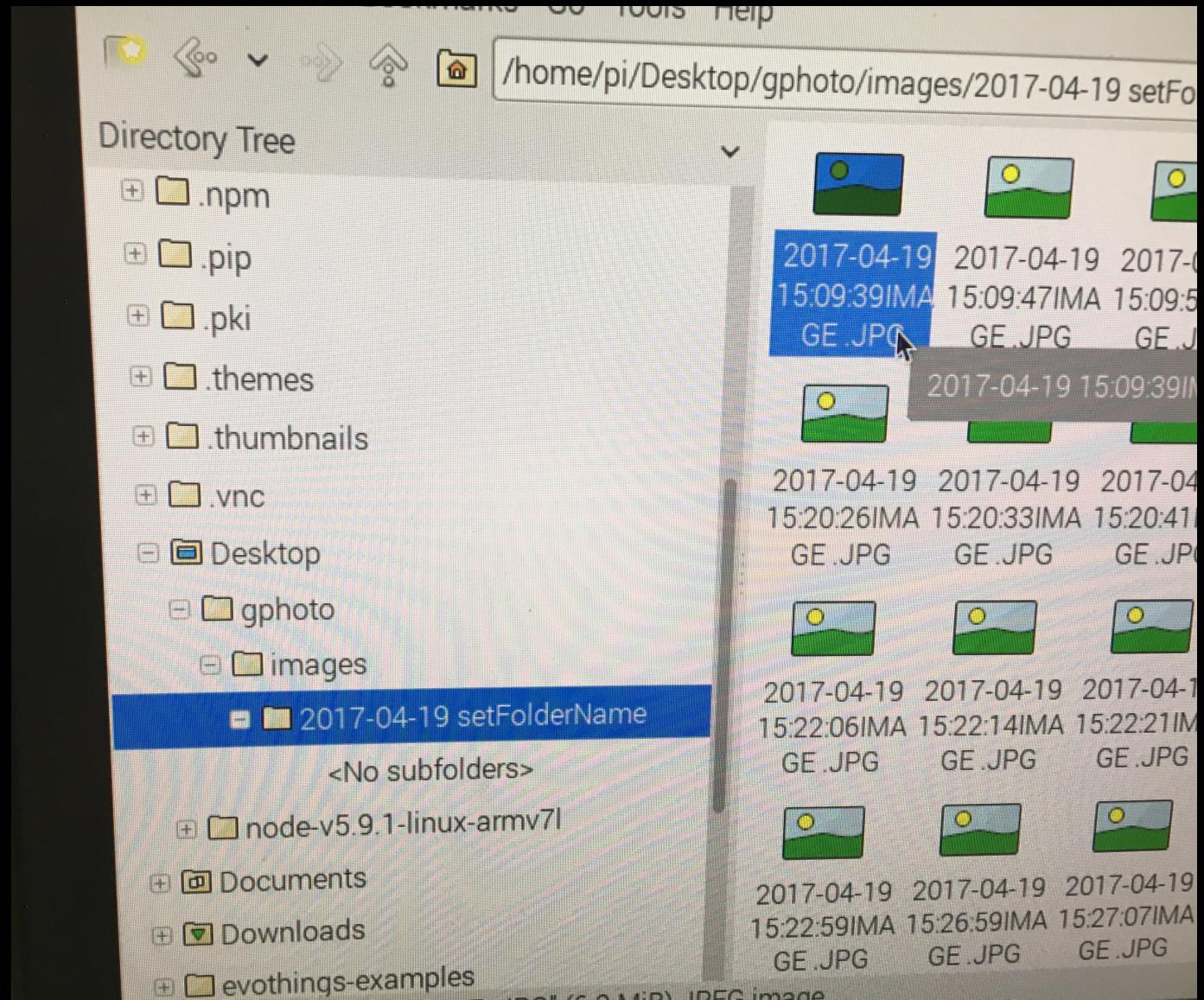
#kill gphoto2 process that
#starts whenever we connect the camera
def killgphoto2Process():
    p= subprocess.Popen(['ps',' -A'],stdout=subprocess.PIPE)
    out, err = p.communicate()
    #search for the line that has the process
    #we want to kill
    for line in out.splitlines():
        if b'gvfsd-gphoto2' in line:
```

STEPS

- Configuration of Firebase: so that the raspberry pi could access the DATA in Firebase
- Obtain from DATA of Firebase
- If the variable "trigger" says ON, start the shooting process, or if "OFF" end this process
- Download pictures from camera to local computer a

STEPS

- User could create a new folder or use the default one
- Download pictures from camera to local computer, and user also rename these pictures.



ADVANTAGE

- All the data is real time, so if the user want to change some variables during the process, he do not need to end the process.
- For example, if the lights become dark, user could extend the exposure time during the shooting period.

