|  |
| --- |
| 카메라센서 – 임시원 |
| |  | | --- | | 사전설치 및 환경설정 | | **~h264 -> mp4로 변환을 위한 패키지 설치~**  MP4Box(h264->mp4변환.명령어ex)MP4Box -add a.h264 a.mp4)  sudo apt-get install gpac  **~윈도우(라즈베리가 공유폴더에 접근할 수 있도록 환경설정을 해줘야 한다)~**  **1. 윈도우 키 + R을 누르고 다음 명령을 입력하여 실행합니다.**   |  |  | | --- | --- | | control /name [Microsoft.NetworkAndSharingCenter](http://microsoft.networkandsharingcenter/) | [cs](http://colorscripter.com/info#e) |     https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/7ccb1efb-96f2-4d1a-b956-3e2bfd25da4b.png?resizeSmall&width=832  **2. 고급 공유 설정 변경을 선택합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/2097a7cf-7698-4e59-9eb6-0ac4a0b95cf9.png?resizeSmall&width=832  **3.  네트워크 검색 켜기와 파일 및 프린터 공유 켜기를 체크합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/3dc85b37-ddac-4776-a3be-0f80abf89825.png?resizeSmall&width=832  **아래 화면처럼 선택하고 변경 내용 저장을 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/92255db4-d148-416d-95e3-2292e150251a.png?resizeSmall&width=832    **4. 공유할 폴더를 선택하고 마우스 우클릭 후, 메뉴에서 속성을 선택합니다.**  **공유 탭을 선택하고 고급 공유를 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/b338a5b7-d9ea-443c-aaa5-c318a5925850.png?resizeSmall&width=832  **5. 선택한 폴더 공유를 체크하고 권한을 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/84e67a70-f73d-4af0-91ec-02148fb3892e.png?resizeSmall&width=832  **6. 추가를 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/04fe8030-a2bb-44c3-ae15-0470116f4ef0.png?resizeSmall&width=832  **7. 선택할 개체 이름에 라즈베리파이에서 공유폴더에 접근시 사용할 윈도우에 등록된 사용자 이름을 적어주고 확인을 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/91d226f1-8e90-42f1-969d-2e8d4a9da36d.png?resizeSmall&width=832  **8. 방금 추가된 사용자를 선택하고 모든 권한의 허용을 체크해줍니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/9577a562-1aa8-4e28-b1b1-75fc60393d47.png?resizeSmall&width=832  **9. 추가한 사용자외에 다른 사람들은 공유폴더에 접근할 수 없도록 Everyone을 선택하고 제거를 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/f3982a28-af8c-41e1-b6d9-ffedba9f7bf6.png?resizeSmall&width=832  **10. 파일 공유시 방어벽이 문제가 안되도록 설정을 변경해줘야 합니다.**  **제어판에서 시스템 및 보안 > Windows 방어벽에서 앱 허용을 선택합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/2fb877c9-9bd9-4c07-acd1-e1f0c9fc88b1.png?resizeSmall&width=832  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/56f94e30-f5c1-4748-a57e-4dc2982bf846.png?resizeSmall&width=832  **11. 설정 변경을 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/b080ccc3-2769-4ec6-97ce-f53bfb1e9c35.png?resizeSmall&width=832  **12. 목록에서 파일 및 프린터 공유를 찾아서 개인 항목을 체크하고 확인을 클릭합니다.**  https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/15d49b77-b121-4234-a09b-d47afa1c9886.png?resizeSmall&width=832  **~라즈베리(윈도우 공유폴더에 접근해야 한다)~**  **1. Samba를 위해 필요한 패키지를 설치합니다.**   |  |  | | --- | --- | | $ sudo apt-get install samba samba-common | [cs](http://colorscripter.com/info#e) |   **2. 공유폴더를 생성합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo mkdir /media/pi  pi@raspberrypi:~ $ sudo chmod 777 /media/pi | [cs](http://colorscripter.com/info#e) |   **3. /etc/samba/smb.conf 파일을 수정합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo nano /etc/samba/smb.conf | [cs](http://colorscripter.com/info#e) |   **3-1. 노란줄을 추가합니다.**   |  |  | | --- | --- | | [global]    ## Browsing/Identification ###    # Change this to the workgroup/NT-domain name your Samba server will part of    workgroup = WORKGROUP    # Windows Internet Name Serving Support Section:  # WINS Support - Tells the NMBD component of Samba to enable its WINS Server  #   wins support = no  wins support = yes  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |     **3-2. 파일 끝부분에 추가합니다.**   |  |  | | --- | --- | | [raspberrypi]  path=/media/pi  browseable=Yes  writeable=Yes  only guest=no  guest ok = no  create mask=0777  directory mask=0777  public=no | [cs](http://colorscripter.com/info#e) |     **4. 설정을 적용하기 위해 samba 데몬을 재시작합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo /etc/init.d/samba restart  [ ok ] Restarting nmbd (via systemctl): nmbd.service.  [ ok ] Restarting smbd (via systemctl): smbd.service. | [cs](http://colorscripter.com/info#e) |     **5.삼바 계정 패스워드를 생성합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo smbpasswd -a pi  New SMB password:  Retype new SMB password:  Added user pi. | [cs](http://colorscripter.com/info#e) |     **6. smbclient 패키지를 설치해야 합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo apt-get install smbclient | [cs](http://colorscripter.com/info#e) |     **7.공유 폴더를 마운트할 디렉토리도 생성해줍니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo mkdir /media/windows | [cs](http://colorscripter.com/info#e) |     **8. -L 다음에 윈도우 컴퓨터의 이름을 적고, -U 다음에 윈도우 컴퓨터 사용자 계정 이름을 적습니다.**  다음처럼 “protocol negotiation failed: NT\_STATUS\_CONNECTION\_RESET” 에러가 나는 경우 추가 옵션을 적어줘야 합니다.  ( [https://www.novell.com/support/kb/doc.php?id=7019892](https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.novell.com%2Fsupport%2Fkb%2Fdoc.php%3Fid%3D7019892) )  **(+비밀번호 설정은** **제어판\사용자 계정 및 가족 보호\사용자 계정 에서 설정해주시길)**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo smbclient -L webnautes-pc -U webnautes  WARNING: The "syslog" option is deprecated  Enter webnautes's password:  protocol negotiation failed: NT\_STATUS\_CONNECTION\_RESET | [cs](http://colorscripter.com/info#e) |     (+ 컴퓨터사용자계정이름 대신 IP를 적어줘도 됨)   |  |  | | --- | --- | | sudo smbclient -L 192.168.0.133 -U kosta | [cs](http://colorscripter.com/info#e) |   https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/02335ad3-6297-4e58-b31a-bc59bdf8a994.png?resizeSmall&width=832  **9.  앞에서 찾은 윈도우의 공유 폴더를 마운트하려면 cifs-utils 패키지가 필요합니다.**   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo apt-get install cifs-utils | [cs](http://colorscripter.com/info#e) |     **10. 윈도우와 라즈베리를 연동할(이게 마운트인듯) 폴더를 지정한다(라즈베리 킬때마다 해줘야했음..)**   |  |  | | --- | --- | | sudo mount -t cifs -o user=kosta,file\_mode=0777,dir\_mode=0777 //192.168.0.133/BlackBox /media/windows    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/e4faa827-b1d2-4827-b5c5-002484006c30.png?resizeSmall&width=832  **11. 파일을 생성해주면, 윈도우에서 해당 파일이 보입니다**   |  |  | | --- | --- | | pi@raspberrypi:~ $ touch /media/windows/테스트 | [cs](http://colorscripter.com/info#e) |   https://www.evernote.com/shard/s675/sh/167cab89-60c3-4ad5-ad13-e3b0094db428/2f11c9f64bf6b1c7/res/c3b1dc3b-5171-4775-8ef9-754b9908d786.png?resizeSmall&width=832  작업이 모두 끝나면 다음 명령으로 마운트를 해제합니다.   |  |  | | --- | --- | | pi@raspberrypi:~ $ sudo umount /media/windows | [cs](http://colorscripter.com/info#e) | |  |  | | --- | | 소스 | | start.py(카메라모듈& 플라스크 통신)   |  |  | | --- | --- | | #!/usr/bin/env python  from flask import Flask, request, render\_template, Response  from camera import Camera  from multiprocessing import Process  import time    app = Flask(\_\_name\_\_)  cam = Camera()    @app.route('/rec')                             #IP:5000/rec으로 접근 시, 녹화 시작  def index():      return render\_template('index.html')      #비디오 스트리밍 html(/templates/안의 index.html)  @app.route('/stop')                            #IP:5000/stop으로 접근 시, 녹화종료 후 플라스크 종료  def stop():      cam.stopRec()                               #녹화 종료      shutdown\_server()                           #플라스크 종료      print("camera recording end...")      return '\* FLASK SERVER CLOSED \*'         #html에 플라스크서버 종료 메시지 띄우기    @app.route('/video\_feed')  def video\_feed():      #Request후에 받는 Response정보(정확히는 Response Headers에 들어가는 정보인듯. 크롬에서 F12로 확인가능?)      return Response(gen(cam), mimetype='multipart/x-mixed-replace; boundary=frame')    def gen(camera):                                #비디오 스트리밍을 생성한다      while True:          frame = cam.get\_frame()          #바이너리로 처리한다는 설정값인듯          yield (b'--frame\r\n'                 b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')      print("preview start...")    def shutdown\_server():                          #플라스크 서버를 종료한다      func = request.environ.get('werkzeug.server.shutdown')  #종료 명령      if func is None:          raise RuntimeError('Not running with the Werkzeug Server')      func()    if \_\_name\_\_ == '\_\_main\_\_':      server = Process(target=app.run(host='0.0.0.0', debug=False, threaded=True))    #멀티프로세스 사용(플라스크 서버 종료를 구현하기 위해 사용)      server.start()      server.terminate()      #join은 지워도 될라나      server.join()      [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   camera.py(카메라 미리보기 & 녹화)   |  |  | | --- | --- | | import time,io,threading,picamera,subprocess    class Camera(object):      thread = None      frame = None      last\_access = 0      now = time.localtime()      s = "day\_%04d\_%02d\_%02d\_time\_%02d%02d%02d" % (now.tm\_year, now.tm\_mon, now.tm\_mday, now.tm\_hour, now.tm\_min, now.tm\_sec)      path = '/media/pi/h264/'      cpPath = '/media/windows/'      str = s + '.h264'      cpStr = s + '.mp4'      path += str      cpPath += cpStr        def initialize(self):          if Camera.thread is None:              Camera.thread = threading.Thread(target=self.\_thread)              Camera.thread.start()              while self.frame is None:                  time.sleep(0)        def get\_frame(self):          Camera.last\_access = time.time()          self.initialize()          return self.frame        def \_thread(self):          self.camera = picamera.PiCamera()          self.camera.resolution = (480,318)#720,480          self.camera.hflip = True          self.camera.vflip = True            self.camera.start\_preview()                              #미리보기 시작          self.camera.start\_recording(self.path, format='h264')   #블랙박스 녹화 시작          print("camera recording start...")            time.sleep(2)          stream = io.BytesIO()                                   #바이너리스트림 오픈          for foo in self.camera.capture\_continuous(stream, 'jpeg',use\_video\_port=True):              stream.seek(0)              self.frame = stream.read()              stream.seek(0)              stream.truncate()              if time.time() - self.last\_access > 10:                  break          self.thread = None        #rec종료      def stopRec(self):          self.camera.stop\_recording()          print("camera recording end...")          try:              time.sleep(1)              test = 'MP4Box -add ' + self.path + ' ' + self.cpPath              print(test)              # 파이썬코드에..비밀번호를 쓰는건 좀...다른방법도 찾아보자ㅠㅠ              #라즈베리와 연동된 윈도우 계정 로그인(윈도우 공유폴더에 접근하기 위함)              subprocess.call('sudo mount -t cifs -o user=kosta,password=kosta,file\_mode=0777,dir\_mode=0777 //192.168.0.133/mp4 /media/windows',shell=True)              time.sleep(5)                print("업로드 시작...")              #h264파일을 mp4로 변환 후, 윈도우 공유폴더 경로에 저장              subprocess.call('MP4Box -add ' + self.path + ' ' + self.cpPath, shell=True)              print("영상 변환 성공")          except:              print("영상 변환 실패")          return  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) | |  |  | | --- | | 회로도 | |  |  |  | | --- | | 시연사진 | | 플라스크 서버 시작    192.168.0.132:5000/rec에 접속하여 실시간 관제 & 녹화 시작    192.168.0.132:5000/stop에 접속하여 관제 및 녹화 종료 & 플라스크 서버 종료    녹화 종료 후, 저장된 블랙박스파일 확인 | |