

9월 업무일지

서울시립대학교 컴퓨터과학부

2017920036 양다운

인턴 활동 / 업무일지

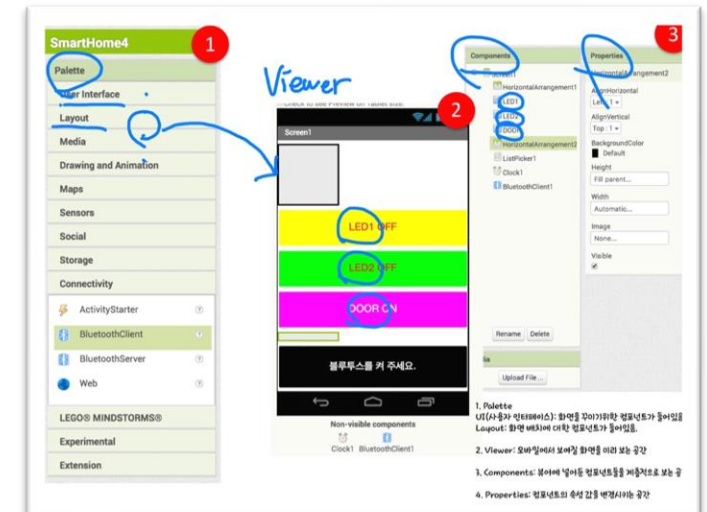
9/1 ~ 9/2

1. 청라중학교 강의 영상 제작

1. Adobe Premiere Pro 사용하기
2. 촬영 및 녹음하기 - 노트북은 Xbox 프로그램, 아이패드에는 RECGO 앱 다운로드
3. 아이패드-굿노트로 ppt 설명하기
4. 노트북-실습 과정 설명하기
5. 2,3강 각각의 20분 정도의 강의 제작
6. 저작권 문제를 야기하는 폰트나 음원 사용하지 않기

2. 2강. App Inventor를 활용한 원격제어 코딩

1. 아두이노에 포함된 함수들 요약
2. code.org에 접속하여 코딩공부를 다양하게 가능
3. Appinventor 실행하기 위한 환경 설정
4. Appinventor 프로그램 사용하는 방법
5. smart_home 프로젝트 제작-Designer와 Block 부분



3. 3강. App Inventor를 이용한 GET방식의 연동 실습

1. smart_home 프로젝트의 Block 부분 코드 설명하기
 1. ListPicker1
 2. LED1, LED2
 3. DOOR
2. GET방식과 POST 방식 비교하기, GET방식을 이용한 이유
3. GET방식을 이용하고자 필요한 프로그램 소개
4. Google chart api 활용
5. Google_home_chart.html 코딩
 1. Get방식을 이용하여 값 가져오기
 2. 온도, 습도 센서 가져오기
 3. 센서 값을 구글 차트에 적용하기
6. Google_home_chart.html 적용하기-get방식 데이터 전달
7. Appinventor, 중학생들 실습하기
8. 안드로이드폰이나 애물레이터 프로그램으로 제작한 앱 실행하기

** 데이터를 클라이언트(Client)에서 서버(Server)로 보내는 방식 2가지

1. GET 방식

- 데이터를 주소에 묻혀서 보내짐. -> 보안성 취약.
- 게시판 글 조회나 검색 같이 정보를 가져올 필요성이 있을 때 사용.
- 전송가능한 데이터의 최대 크기는 브라우저별로 다르지만, 크기가 정해져 있음.(2048byte)

http://emotionreport.co.kr/google_chart.html?t=12&h=23&t1=45&h1=45&t2=14&h2=98

주소 부분

파일 이름

GET 방식 데이터 전달 (key= value)

2. POST 방식

- 데이터를 주소에 묻혀 보내지 않고, 전송 객체의 메세지안에 담아서 전달. -> 보안성에 강함.
- 비밀번호나 주민번호 등 private한 데이터를 전송할 때 사용.
- 데이터 양의 제한이 없음. -> 대량의 데이터 전송 가능.

#. GET 방식이 POST 방식보다 좋은 이유?

- 속도가 빠르다. 하지만 이유를 아는 것이 중요하다.
- GET 방식의 요청은 캐싱이 있기 때문이다.
- 캐싱: 한번 접근한 후, 재요청 시 빠른 접근을 위해 데이터를 저장시켜 놓는 것

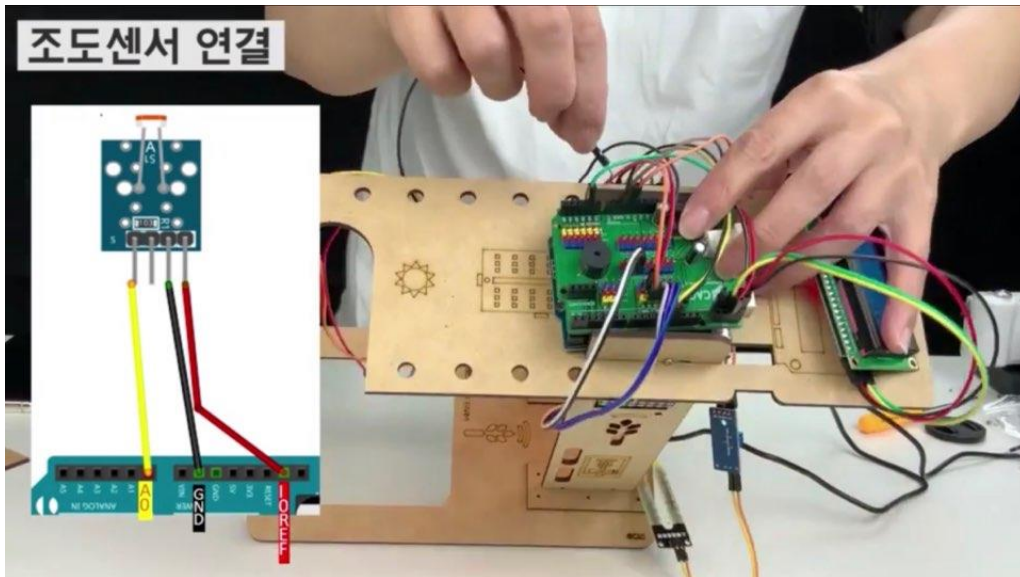
9/7 ~ 9/11

1. SmartFarm 제작 영상 촬영

1. 이영옥팀장님과 아두이노 실드를 이용하여 smartfarm KIT 제작하기

2. SmartFarm 영상 편집

1. Adobe Premiere Pro 사용하기
2. KIT 부품 check와 smartfarm KIT 제작과정



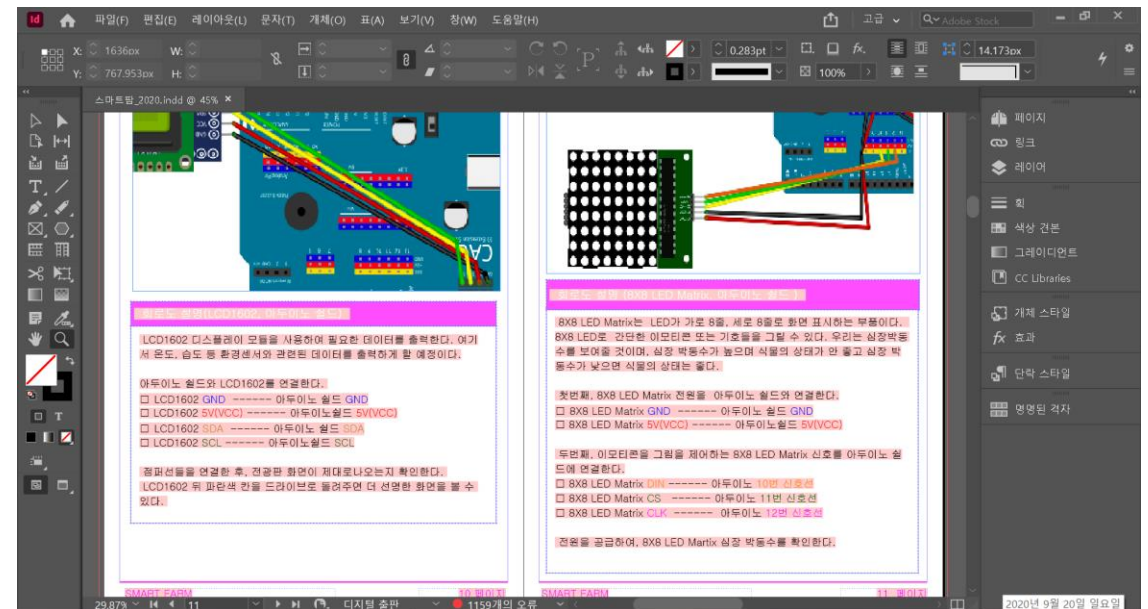
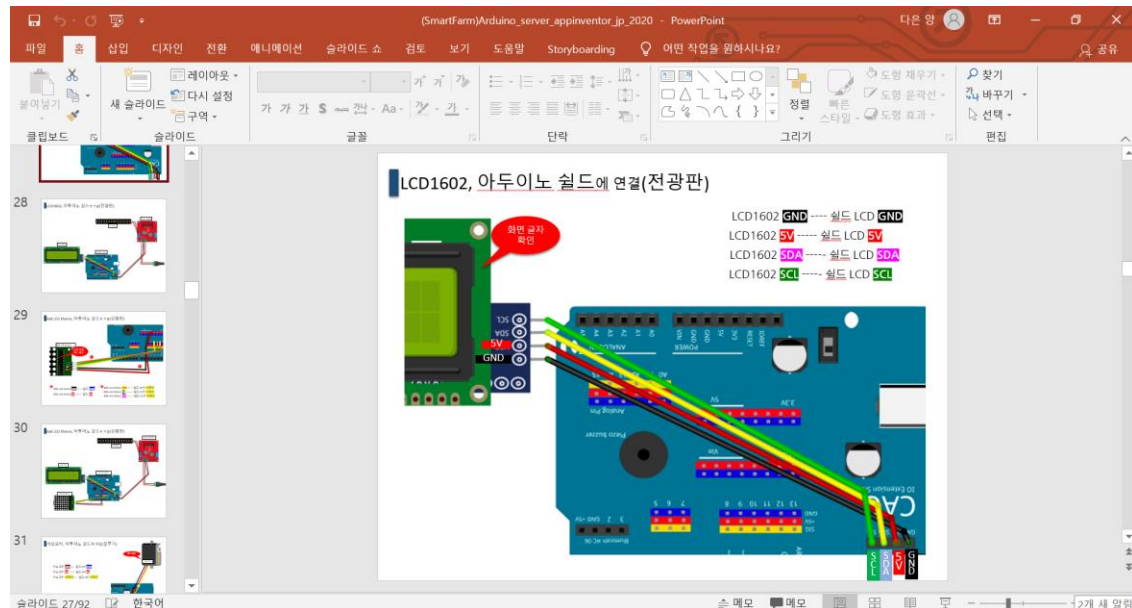
9/7 ~ 9/11

3. SmartFarm 강의자료 업데이트

1. Fritzing 프로그램 이용하여 회로도 변경하기
2. smartfarm KIT 제작과정(회로도 포함) 자료 수정하기

4. SmartFarm 도서 회로도 업데이트

1. Adobe InDesign 사용하기
2. smartfarm KIT 제작과정(회로도, 제작단계 설명 포함) 자료 수정하기



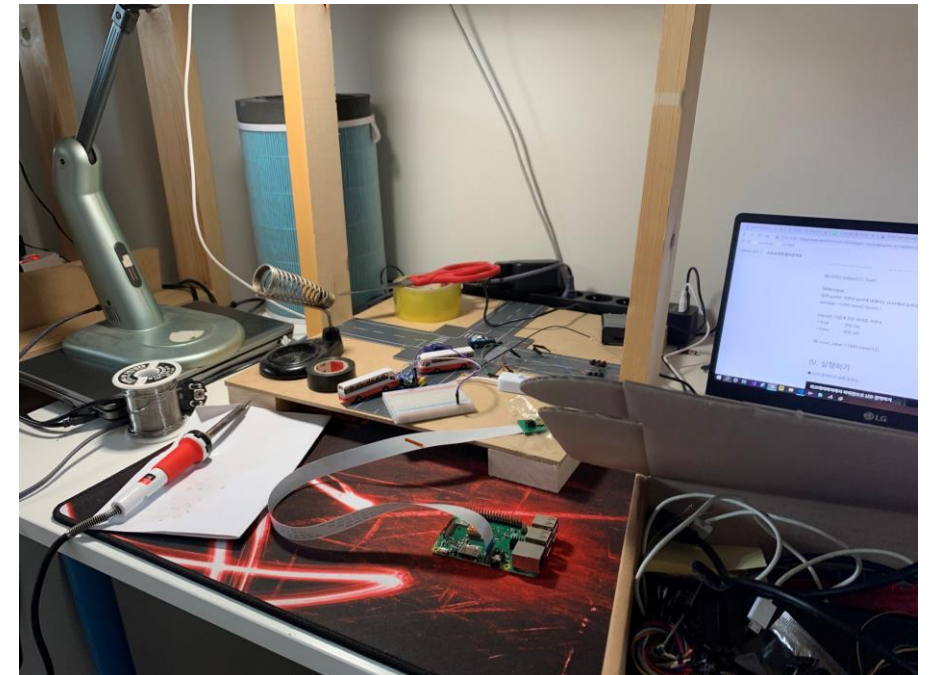
9/14 ~ 9/18

1. Raspberry Pi 작동법

1. 전원 on -> cd Desktop/ free camera/ (파일 경로 변경) -> python3 ~~.py -> 프로그램 실행 -> ctrl+C 마침
2. Tab 클릭하면 자동완성
3. raspistill -o ~.jpg (카메라 프로그램 실행)
4. ls (파일 찾기)
5. sudo reboot (라즈베리파이 리셋)

2. 자동차를 인식할 카메라 위치 선정하기

1. 위에 설치
 1. Dataset안에 자동차 윗면 이미지가 없어 자동차를 인식하지 못한다.
 2. 실제로 하늘에 신호등을 달 수 없고, 비효율적이다.
2. 사거리 중앙에 설치
 1. 최소 두대 이상의 카메라를 설치해야 한다.
3. 옆면에 설치, 현 신호등 위치
 1. 자동차 옆면 이미지를 제대로 파악한다.
 2. 효율적인 위치
 3. 결정!

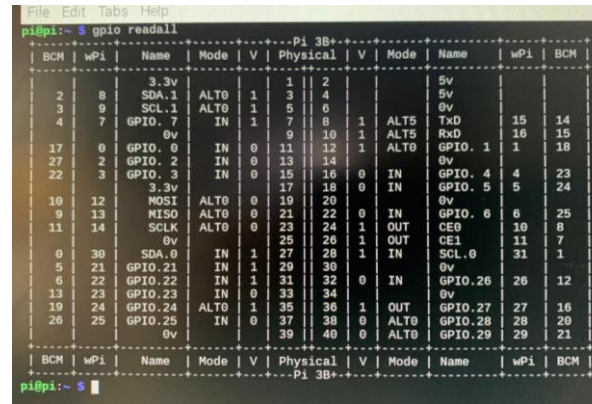
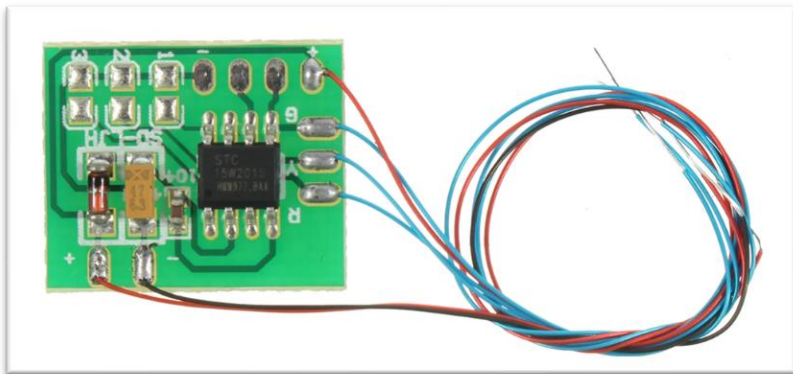


3. 카메라를 옆면에 설치할 때, 야기될 문제점

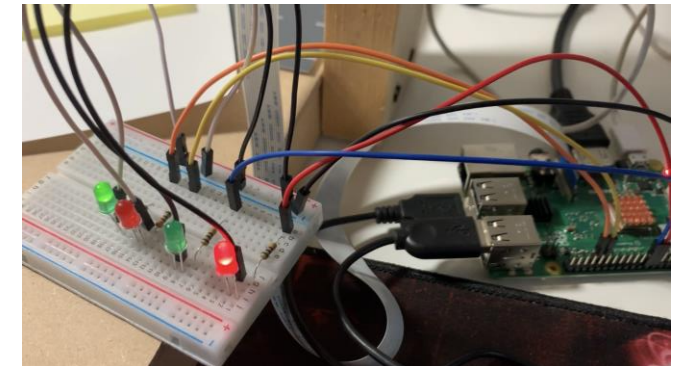
1. 1차선인 경우는 크게 무리 없지만, 2차선 이상일 경우 카메라와 가까운 차선의 자동차만 인식 가능
2. 빠르게 움직이는 자동차는 인식하지 못한다. -> 정차된 상태에 자동차 수 측정하기
3. 자동차 색깔, 크기에 따라 인식될 정확도가 달라진다. -> 밝은 색과 뚜렷한 자동차 모형만 사용하기
4. 대표님 왈, 중요한 점은 알고리즘을 제작하는 것이다. 세세한 단점들은 이후에 보완할 점들이다.

4. 신호등 제작하기

1. 신호등 변환 효과를 제어하는 마이크로 회로 기판 사용하고자 했으나, 코딩으로 신호등 제어가 불가능하여 포기
2. 수작업 납땜하여 신호등 on/off 확인하기
3. 빨간색, 초록색 LED 센서 각 두 개씩 이용하여 브레드보드에 설치하기(저항, 점퍼선, 스위치 등등 사용)
4. 라즈베리 파이와 연결하면 핀번호가 BCM/Board 마다 다르다는 점을 알게 되었다. -> BCM의 GPIO번호 사용

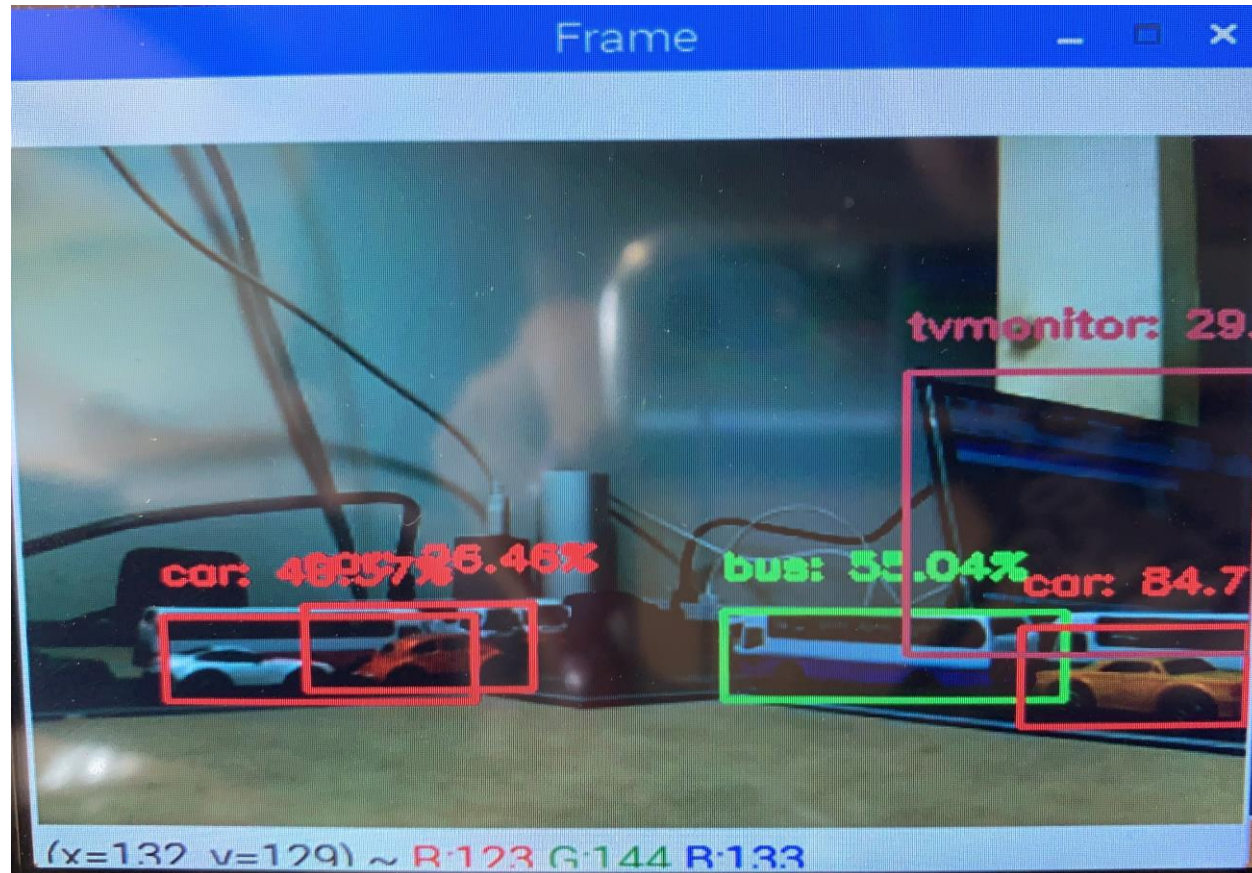


BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
-----Pi 3B+-----										
2	8	3.3v		1	1	2		5v		
3	9	SDA.1	ALT0	1	3	4		5v		
4	7	SCL.1	ALT0	1	5	6		0v		
		GPIO. 7	IN	1	7	8	1	ALT5	TxD	15
					9	10	1	ALT5	RxD	16
17	0	GPIO. 0	IN	0	11	12	1	ALT0	GPIO. 1	1
27	2	GPIO. 2	IN	0	13	14		0v		18
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4
					17	18	0	IN	GPIO. 5	5
					19	20		0v		23
10	12	MOSI	ALT0	0	19	20		0v		24
9	13	MISO	ALT0	0	21	22	0	IN	GPIO. 6	6
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	8
					25	26	1	OUT	CE1	11
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31
5	21	GPIO. 21	IN	1	29	30		0v		1
6	22	GPIO. 22	IN	1	31	32	0	IN	GPIO. 26	26
13	23	GPIO. 23	IN	0	33	34		0v		12
19	24	GPIO. 24	ALT0	1	35	36	1	OUT	GPIO. 27	27
26	25	GPIO. 25	IN	0	37	38	0	ALT0	GPIO. 28	28
					39	40	0	ALT0	GPIO. 29	29
										21
-----Pi 3B+-----										
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM



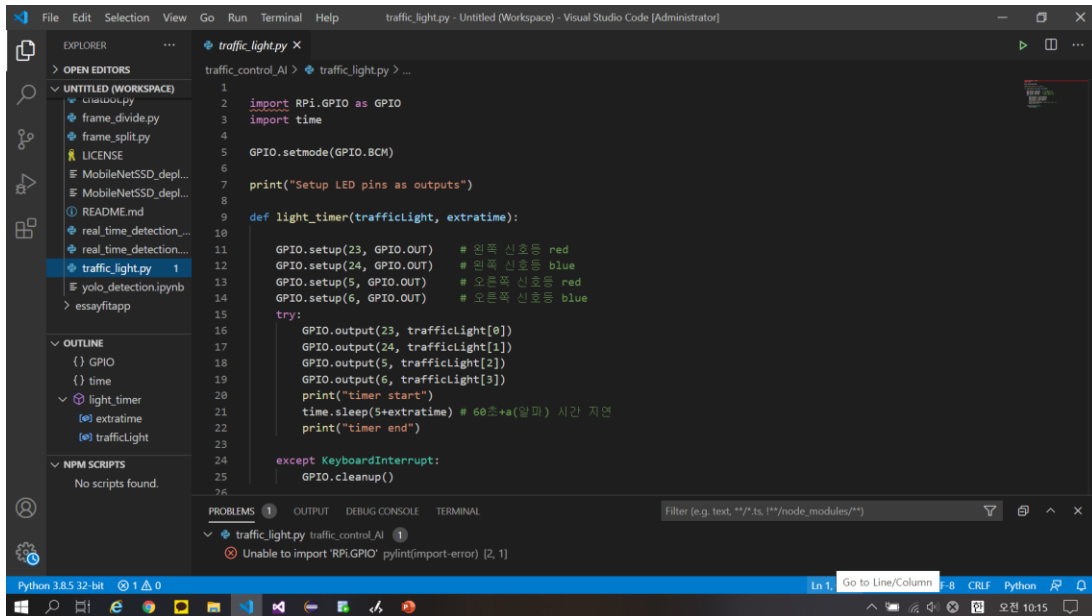
5. 카메라로 측정된 Frame 분할하기

1. 왼쪽, 오른쪽 각각의 차선에 자동차 수를 측정하기 위해 화면을 반으로 분할하고자 한다.
2. 처음엔 빨간 신호에 따라 정차된 자동차 수를 count하고자 했으나, 측정되는데 시간이 지연되는 문제점 발생하여 포기
3. 측정된 자동차의 (startX, startY) 좌표와 (endX, endY) 좌표 값을 출력하기 -> frame을 나눌 기준선(width, height) 잡기



6. Raspberry Pi-OpenCV 신호등 알고리즘 제작

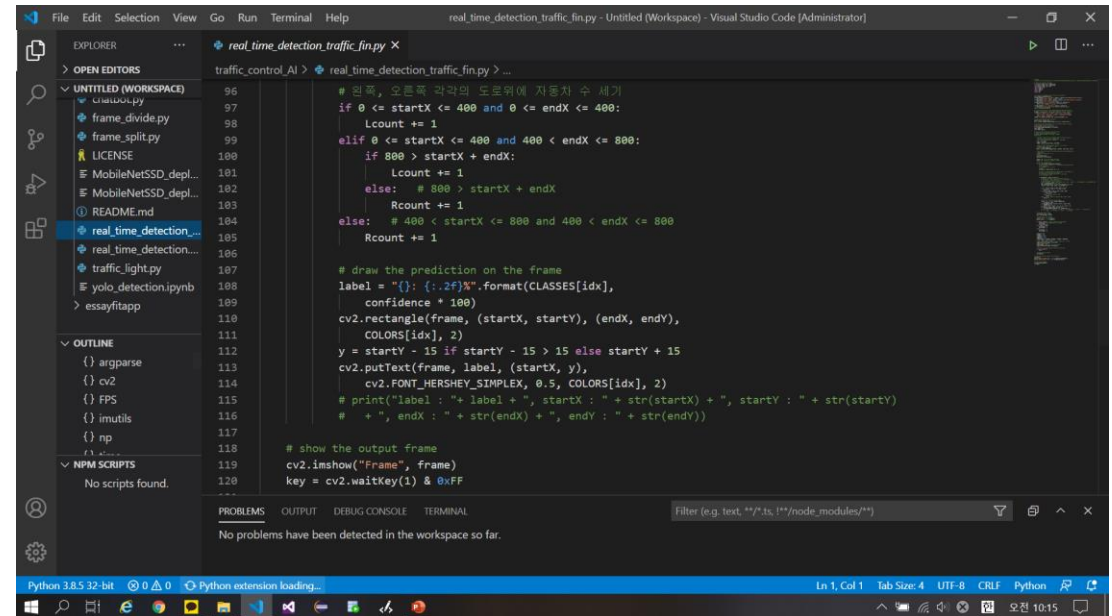
1. Raspberry Pi-OpenCV 신호등 알고리즘 제작
2. OpenCV를 이용하여 카메라로 자동차 인식하기-적절한 카메라 위치 선정&class 인식가능한 코딩
3. 왼쪽, 오른쪽 자동차 수를 각각 count하여 비교하기-Lcount, Rcount를 비교하여 초록색 LED 센서 지연 시간 (extratime) 발생하는 코딩
4. 지연되는 신호등 시간을 LED센서에 적용하기-traffic_light.py의 light_timer 함수를 이용하여 LED sleep 시간 조절하는 코딩



```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5
6 print("Setup LED pins as outputs")
7
8 def light_timer(trafficLight, extratime):
9
10     GPIO.setup(23, GPIO.OUT) # 왼쪽 신호등 red
11     GPIO.setup(24, GPIO.OUT) # 왼쪽 신호등 blue
12     GPIO.setup(5, GPIO.OUT) # 오른쪽 신호등 red
13     GPIO.setup(6, GPIO.OUT) # 오른쪽 신호등 blue
14
15     try:
16         GPIO.output(23, trafficLight[0])
17         GPIO.output(24, trafficLight[1])
18         GPIO.output(5, trafficLight[2])
19         GPIO.output(6, trafficLight[3])
20         print("timer start")
21         time.sleep(5+extratime) # 60초+(일파) 시간 지연
22         print("timer end")
23
24     except KeyboardInterrupt:
25         GPIO.cleanup()
26

```



```

96 # 왼쪽, 오른쪽 각각의 도로위에 자동차 수 세기
97 if 0 <= startX <= 400 and 0 <= endX <= 400:
98     Lcount += 1
99 elif 0 <= startX <= 400 and 400 < endX <= 800:
100     if 800 > startX + endX:
101         Lcount += 1
102     else: # 800 > startX + endX
103         Rcount += 1
104 else: # 400 < startX <= 800 and 400 < endX <= 800
105     Rcount += 1
106
107 # draw the prediction on the frame
108 label = "{}: {:.2f}%".format(CLASSES[idx],
109                             confidence * 100)
110 cv2.rectangle(frame, (startX, startY), (endX, endY),
111              COLORS[idx], 2)
112 y = startY - 15 if startY - 15 > 15 else startY + 15
113 cv2.putText(frame, label, (startX, y),
114            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
115 # print("label : " + label + ", startX : " + str(startX) + ", startY : " + str(startY)
116 # + ", endX : " + str(endX) + ", endY : " + str(endY))
117
118 # show the output frame
119 cv2.imshow("Frame", frame)
120 key = cv2.waitKey(1) & 0xFF
121

```

9월 18일 금요일

1. 10:10 ~ 12:10

1. Raspberry Pi-OpenCV 신호등 알고리즘 제작
2. Python 으로 코딩 완성!
 1. traffic_light.py
 2. real-time-detection-fin.py
3. Desktop/free camera/ 파일안에 real-time-detection-fin.py
4. car와 bus의 개수를 count하여 왼쪽, 오른쪽 신호등 지연시간 조절하기

2. 12:10 ~ 12:50

1. 점심식사-'미스사이공'에서 베트남볶음밥

3. 13:00 ~ 14:30

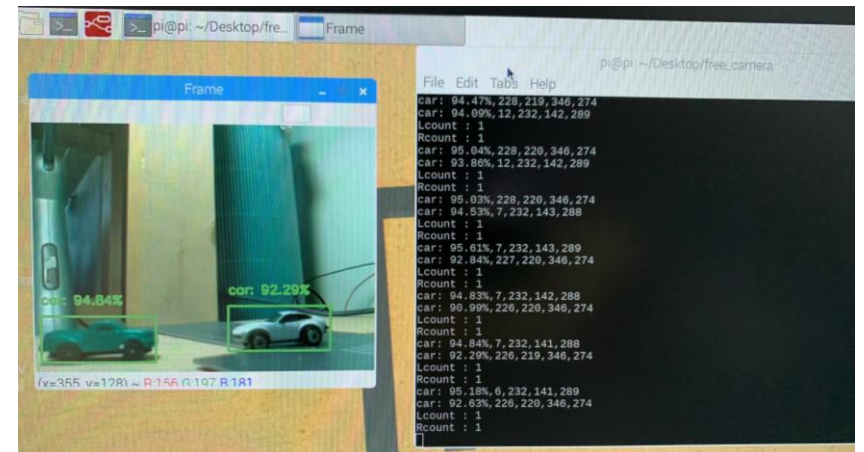
1. cardboardcollege.com 자료실에 있는 smart_farm.pdf 수정하기
2. 브레드보드를 포함한 키트 설명에 부품이름 추가하기

4. 15:00 ~ 16:30

1. 대표님께 구현한 사거리 신호등 설명하기

5. 17:10 ~ 18:40

1. 다음주 보조강사에 사용할 unity에 대해 공부하기



9월 21일 월요일

1. 9:50 ~ 10:30

1. 이사님과 업무회의

1. 신호등 알고리즘-yolo이용, 차 갯수에 따른 자동 제어 (완료)
2. 축구공 + 녹색판 인식하기-yolo, custom Dataset으로 설정
3. 흙 + 좌표값 -> 포크레인 작동-yolo, custom
4. Unity - 야광봉 따라잡기
5. 서버 - NLP 하나씩 upload
6. Unity - touchdesigner/madmapping

7. (YOLO)_Training YOLO v3 for Objects Detection with Custom Data 파일에서 별(*)표 있는 강의영상 위주로 공부

2. 10:40 ~ 12:30

1. Yolo custom 학습하기

3. 13:10 ~ 13:40

1. 점심식사-구내식당

9월 22일 화요일

1. 8:20 ~ 8:40

1. 동인천고 진로캠프 강의 시 유의사항과 필수사항 숙지

2. 8:50 ~ 11:40(1학년-1분반)

1. VR원리체험 수업 진행-보조강사

2. Unity hub, 2018.4.27fl unity, VirtualAqua 등 프로그램과 필요 파일 모든 노트북(20개이상)에 설치

3. Cardboard VR 제작 및 VR 참여

4. Unity 기본 동작법과 Asset Store 활용하기

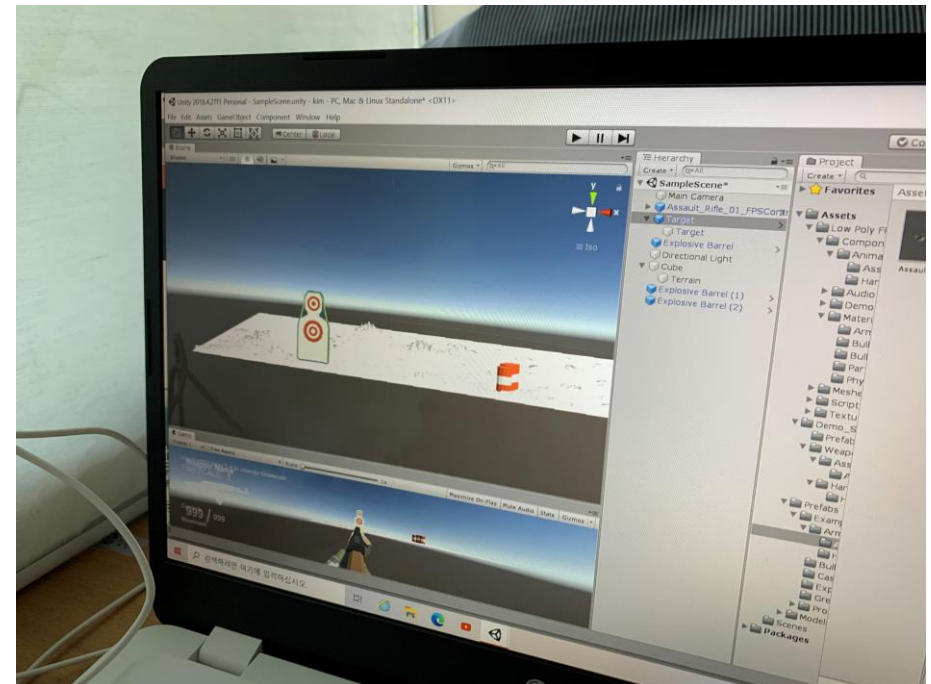
5. VirtualAqua 파일 열어서 함께 물고기 활용하기

3. 11:50 ~ 12:40(1학년-2분반)

4. 12:40 ~ 13:30

1. 점심시간-면사무소

5. 13:30 ~ 15:20(1학년-2분반)



9월 23일 수요일

1. 아버지가 보건소가서 코로나검사를 받아야하기 때문에, 고등학교로 보조강사로 활동하기 어려워 일단 대기
2. 9월 월차 1회 사용
3. 아버지 음성판정. 그래도 고등학교로 출근 못하고 내일 회사로 다시 출근

9월 24일 목요일

1. 9:50 ~ 10:30

1. 화재경보기 설치ing

2. 10:30 ~ 11:30

1. 대표님과 프로젝트 진행

1. essayfitapp 프로그램제작

2. ai_showingtelling.py - 주어진 문장이 묘사(showing)인지, 설명(telling)인지 분류

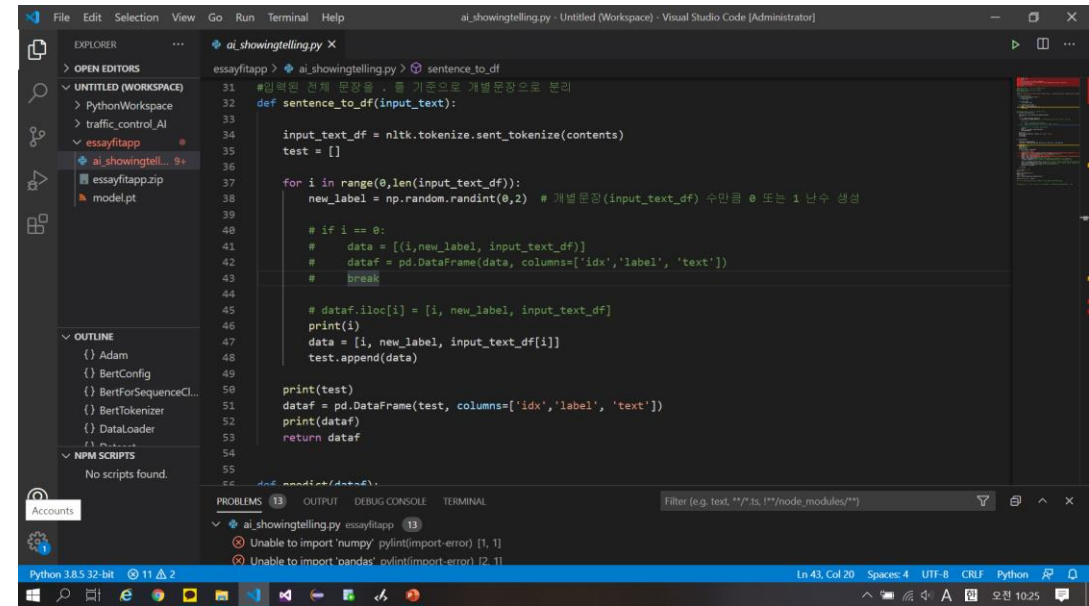
3. Label: 0 -> showing / 1 -> telling

3. 11:40 ~ 12:30

1. 점심식사-구내식당

4. 12:40 ~ 15:20

1. ai_showingtelling.py - 입력된 전체 문장(contents)을 . 를 기준으로 개별문장(input_text_df)으로 분리



```
File Edit Selection View Go Run Terminal Help
ai_showingtelling.py - Untitled (Workspace) - Visual Studio Code [Administrator]

EXPLORER
  OPEN EDITORS
  UNTITLED (WORKSPACE)
  PythonWorkspace
  traffic_control_AI
  essayfitapp
  ai_showingtelling.py
  essayfitapp.zip
  model.pt

OUTLINE
  Adam
  BertConfig
  BertForSequenceCl...
  BertTokenizer
  DataLoader
  NPM SCRIPTS
  No scripts found.

ai_showingtelling.py
31 #입력된 전체 문장을 . 를 기준으로 개별문장으로 분리
32 def sentence_to_df(input_text):
33
34     input_text_df = nltk.tokenize.sent_tokenize(contents)
35     test = []
36
37     for i in range(0, len(input_text_df)):
38         new_label = np.random.randint(0,2) # 개별문장 (input_text_df) 수만큼 0 또는 1 난수 생성
39
40         # if i == 0:
41         #     data = [(i, new_label, input_text_df[i])]
42         #     dataf = pd.DataFrame(data, columns=['idx', 'label', 'text'])
43         #     break
44
45         # dataf.iloc[i] = [i, new_label, input_text_df[i]]
46         print(i)
47         data = [i, new_label, input_text_df[i]]
48         test.append(data)
49
50     print(test)
51     dataf = pd.DataFrame(test, columns=['idx', 'label', 'text'])
52     print(dataf)
53     return dataf
54
55
56 def main():
57     contents = "The cat sat on the mat."
58     dataf = sentence_to_df(contents)
59     print(dataf)
```

9월 25일 금요일

1. 9:55 ~ 12:30

1. 포크레인 프로젝트 진행

1. 흐름 + 좌표값 -> 포크레인 작동-yolo, custom
2. YOLO custom 학습하기

2. 12:30 ~ 14:00

1. 점심식사-쭈꾸미볶음

3. 14:10 ~ 18:30

1. YOLO custom 학습하기

2. vscode에 OpenCV 설치하기(완료X)

9월 28일 월요일

1. 9:50 ~ 11:20

1. 포크레인 프로젝트 진행
2. YOLO custom training 개발환경 설치하기

2. 11:30 ~ 12:20

1. 점심식사-김밥천국

3. 12:20 ~ 18:30

1. YOLO custom training 개발환경 설치하기
2. vscode에 OpenCV 설치하기(완료X) -> 동작은 가능
3. YOLO / YOLO custom 어느정도 숙지
4. Ubuntu 설치할 usb 메모리 가져오기
5. Train/ Test / Validation set 차이 -> 축구공 + 녹색판 이미지(또는 영상)을 Validation set에 담기
6. 내일 오전까지 스마트팜 아이디어 구상하기

9월 29일 화요일

1. 10:00 ~ 10:30

1. 이사님과 업무회의 - 포크레인 프로젝트

1. Raspberry Pi-OpenCV 이용하여 person과 background 인식
2. person -> 기계동작 멈춤 / background -> 기계동작 실행
3. 서브모터 4개 - 로봇 팔 움직임 구현 가능
4. 기어모터 4개 - 포크레인 본체 바퀴 이동 구현 가능
5. I602 LCD 센서d와 Raspberry Pi 연결하기 -> 작동 체크

2. 10:30 ~ 12:40

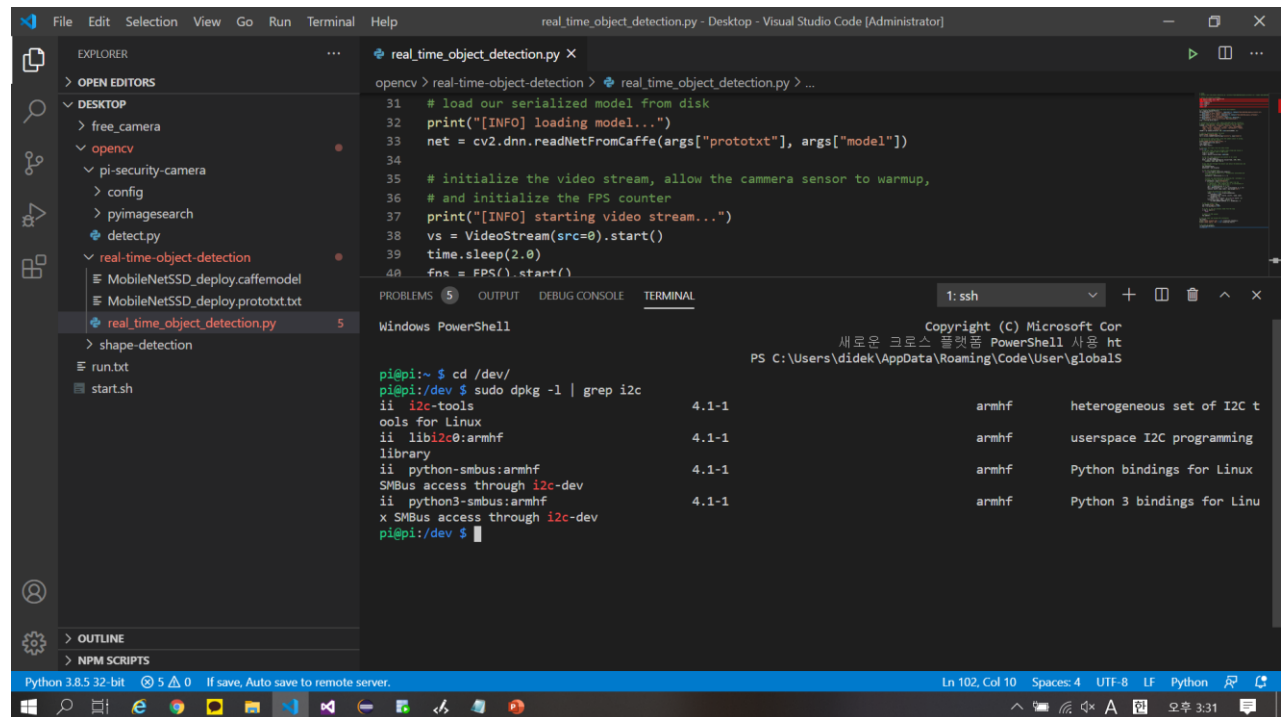
1. I602 LCD 센서d와 Raspberry Pi 연결하기

3. 13:00 ~ 14:40

1. 점심식사-제육볶음

2. 성훈이사님과 대화하기

1. 업무일지 피드백
2. 일을 하면서 제대로 역할 잡기



```
File Edit Selection View Go Run Terminal Help
real_time_object_detection.py - Desktop - Visual Studio Code [Administrator]

EXPLORER
> OPEN EDITORS
> DESKTOP
  > free_camera
  > opencv
    > pi-security-camera
    > config
    > pyimagesearch
    > detect.py
    > real-time-object-detection
      MobileNetSSD_deploy.caffemodel
      MobileNetSSD_deploy.prototxt.txt
      real_time_object_detection.py 5
    > shape-detection
      run.txt
      start.sh
  > OUTLINE
  > NPM SCRIPTS

real_time_object_detection.py X
opencv > real-time-object-detection > real_time_object_detection.py > ...
31 # load our serialized model from disk
32 print("[INFO] loading model...")
33 net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
34
35 # initialize the video stream, allow the camera sensor to warmup,
36 # and initialize the FPS counter
37 print("[INFO] starting video stream...")
38 vs = VideoStream(src=0).start()
39 time.sleep(2.0)
40 fns = FPS().start()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: ssh
Copyright (C) Microsoft Corporation
새로운 크로스 플랫폼 PowerShell 사용
PS C:\Users\didek\AppData\Roaming\Code\User\globalS

Windows PowerShell

pi@pi:~$ cd /dev/
pi@pi:/dev$ sudo dpkg -l | grep i2c
ii i2c-tools                    4.1-1      armhf      heterogeneous set of I2C t
ools for Linux
ii libi2c0:armhf               4.1-1      armhf      userspace I2C programing
library
ii python-smbus:armhf          4.1-1      armhf      Python bindings for Linux
SMBus access through i2c-dev
ii python3-smbus:armhf        4.1-1      armhf      Python 3 bindings for Linu
x SMBus access through i2c-dev
pi@pi:/dev$
```

9월 29일 화요일

4. 15:00 ~ 18:00

1. 1602 LCD 센서와 Raspberry Pi 연결하기

1. 사용중인 라즈베리파이 모델: RPI3 모델 B+

2. 1602 LCD 센서의 SDA와 SCL 핀을 알맞은 Raspberry Pi gpio 핀번호에 연결하기

1. `gpio readall`

gpio 상태 확인

2. `sudo apt-get install python3-smbus`

python3-smbus 라이브러리 설치

3. `i2cetect -y l`

i2c 장치 연결 확인

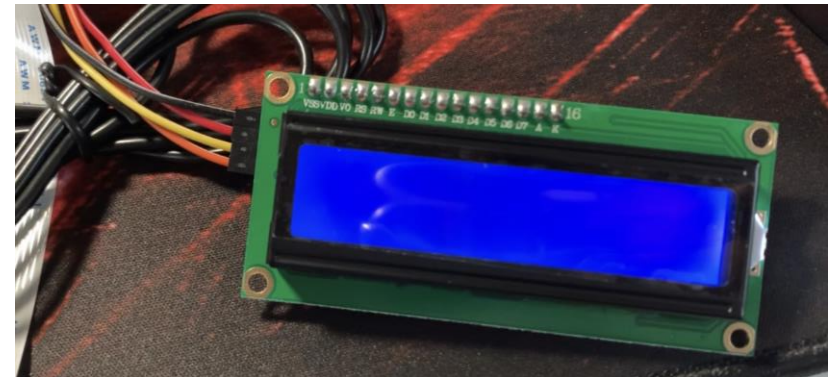
3. Vscode와 라즈베리파이 연결방법

1. View -> command palette -> ftp-simple:Config-FTP connection setting -> host 변경하기(24번)

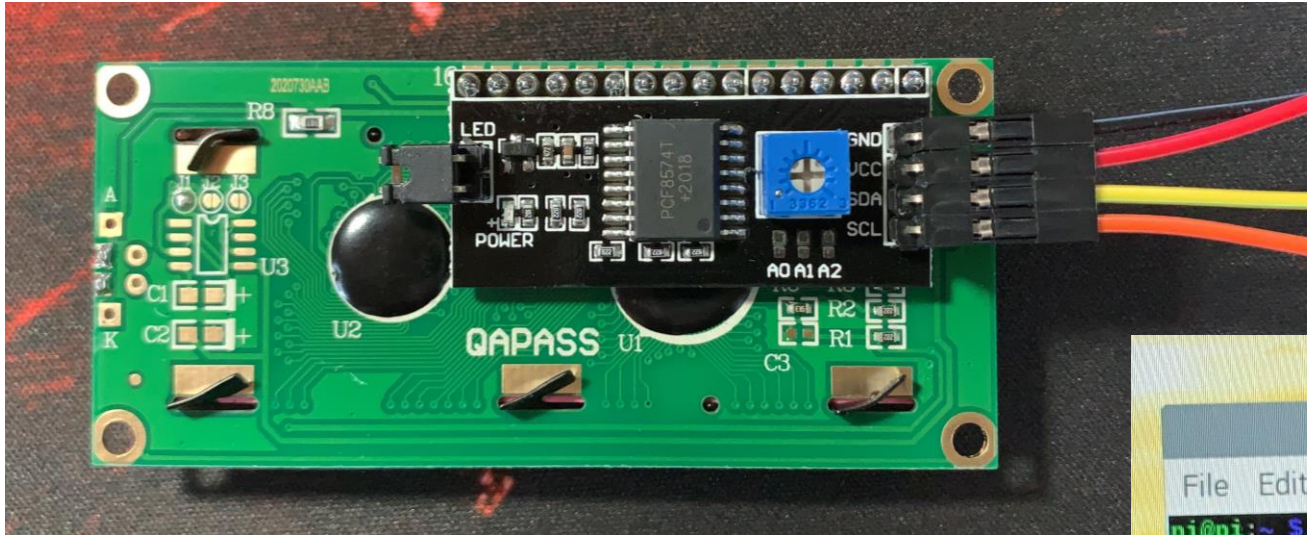
2. View -> command palette -> ftp-simple:Remote directory ~~ -> pi

3. Terminal -> New Terminal -> ssh pi@168.192.0.24 -> password:1234

4. 파일 RPi_I2C_LCD_driver 다운하기 -> ...흠 연결은 되었지만 LCD 화면에 출력 안됨



9월 29일 화요일



```
pi@pi: ~/Desktop/RPi_I2C_LCD_driver/original_example
File Edit Tabs Help
pi@pi:~$ ls
camp.jpg  Documents  MagPi  Pictures  Templates
Desktop  Downloads  Music  Public  Videos
pi@pi:~$ cd Desktop/
pi@pi:~/Desktop$ cd R0i
bash: cd: R0i: No such file or directory
pi@pi:~/Desktop$ cd RPi_I2C_LCD_driver/
pi@pi:~/Desktop/RPi_I2C_LCD_driver$ python3 example.py
python3: can't open file 'example.py': [Errno 2] No such file or directory
pi@pi:~/Desktop/RPi_I2C_LCD_driver$ cd or
bash: cd: or: No such file or directory
pi@pi:~/Desktop/RPi_I2C_LCD_driver$ cd original_example/
pi@pi:~/Desktop/RPi_I2C_LCD_driver/original_example$ python3 examples.py
```