#4월 2일 보강 update upgrade xrdp vinagre

xrdp : X window remote desktop protocol / 설치하면 윈도우에서 원격 접속 가능

sudo apt-get update : 프로그램 설치 시 라즈베리파이 서버 혹은 미러링 사이트에서 라즈베리파이의 최신 버전을 갖고 있는 파일 위치를 업데이트. 업데이트 후 업그레이드 하게 되면 파악된 위치에서 최신 버전 다운 가능. 라즈베리파이 서버의 파일 저장 리스트들(예를 들면 그림판 파일, 익스플로러 등등의 최신 버전 위치 리스트들) 받아 옴. 즉 Install 파일들을 리눅스에선 package라고 부름. 즉 update는 설치 되어 있는 패키지들의 새로운 버전이 있는지 확인하는 것이고 upgrade는 그 최신 버전에 따라서 패키지들을 업그레이드 해줌.

라즈베리파이 임베디드 컴퓨터

> 임베디드 보드(시스템) 특징 : low cost(저비용), minimal(소형), networked(네트워크에 연결되어 있어야 함), mobility(이동성)

소형pc라즈베리파이는 usb, 이더넷, hdmi 출력, 영상출력, 사운드 출력 등의 기능을 갖고 있는 하나의 작은 컴퓨터. 가격이 저렴하고 작은 크기이지만 성능에 문제가 없는 완전한 컴퓨터. 더욱이 단일 마이크로세서와 같이 입출력 신호를 제어할 수 있는 포트가 있음. 범용적인 목적으로 입/출력을 담당하는 GPIO를 갖고 있으며, SPI 통신, I2C(I square C), UART 통신 (Serial 통신= 직렬통신) 갖추고 있음. 특히 SPI 통신(Serial Peripheral Interface 직렬 주변 통신)으로 ADConvert 하면서 고급 센서 쓸 수 있게 됨.

라즈베리는 하나의 임베디드 소형 pc, 아두이노는 마이크로 컨트롤러 보드. 라즈베리는 os가 들어가서 자유도가 높음. 범용적으로 여러 프로그램들을 설치해서 이용 가능. 아두이노는 칩에 프로그램 한번 입력하면 계속 그 프로그램만 사용 가능. 고정되어 있는 프로그램이라 해서 펌웨어(Firm ware) 라고 부르고 유동성이 떨어진다. 하지만 아두이노는 PWM 이나 AD convertor, DA convertor 내장 되어 있어서 센서나 액추에이터(actuator)를 다루기 특화되어 있음. 라즈베리는 AD convertor 부수적으로 달아야 함. 이 두개는 serial 통신을 사용해서 통신 가능함.

sudo = super user(=root, 윈도우로 치면 administrator), apt-get(업데이트 사이트에서 받아서 파일들을 업데이트 해라))

sudo apt-get install xrdp : xrdp 설치

ifconfig : 연결된 IP주소 등 정보 보기

IP주소 복사하기 : 드래그 후 마우스 휠 누르기

sudo apt-get install vinagre : vinagre = 클라이언트(Remote Desktop Viewer), xrdp = 서버

설치 완료 후 윈도우에서 원격 데스크톱 연결(IP주소 입력)

#4월 7일 화면캡쳐 scrot 활용

Scrot : 화면 캡쳐. 캡쳐된 사진 저장은 /home/pi

Scrot &, Scrot : 전체 화면 캡쳐

Scort –u : 활성화된 윈도우 캡쳐

Scrot –s : 드래그해서 캡쳐

Scrot –d 5 : 5초뒤에 전체화면 캡쳐

Scrot –q 50 : 전체 화질 50프로 이하로 저하. (용량 저하)

#4월 9일 samba 1부

Samba : 리눅스/유닉스 서버와 윈도우 기반 클라이언트를 호환하게 해주는 middle ware.

둘 사이에 파일 교환이 이질감없이 양방향으로 되게끔 하는 체계. (interoperability)

리눅스는 passwd 쓸 때 타이핑해도 화면에 아무것도 안 나타남, 두 번 물어봄 -> 리눅스 보안 기본

Leafpad : 리눅스의 메모장 같은 역할.

sudo apt-get install samba samba-common-bin (삼바 설치)

sudo smbpasswd -a pi (삼바 유저 아이디, 패스워드 설정)

('Share' 폴더 만듦 /home/pi/Share)

sudo leafpad /etc/samba/smb.conf

smb.conf : samba의 설정 파일, 관리자 권한으로만 열림.

[pi]
comment = rpi samba server
path = /home/pi/Share //윈도우와 리눅스를 연결하는 Share라는 공유 폴더 쓰겠다.
valid user = pi //그것을 허용하는 사람이 pi
writable = yes //write 가능
browseable //더블클릭해서 직접 실행 가능. 그 권한을 pi에게 주겠다.

sudo service smbd restart : samba restart는 안됨. smbd라고 해야 함.

hostname –I : ifconfig 보다 심플하게 IP주소 알려줌.

설치 완료 되면 윈도우에서 '실행' 열어서 '₩₩ ip주소' 입력 share 폴더 접근 가능.

#4월 14일 samba 2부

sudo apt-get install samba-common-bin –y : samba 설치

ls –al /etc/samba/smb.conf : 권한 확인. (-rw-r--r-- 1 root root ...)

- ➔ **sudo chmod 666 /etc/samba/smb.conf** : 파일 속성을 666(모두 다 읽고 쓰기 가능)이용해서 바꿈. (-rw-rw-rw- : 게스트들도 파일 바꿀 수 있음)
- ➔ **sudo chmod 644 /etc/samba/smb.conf** : 파일 속성을 644(관리자만 변경 가능)이용해서 원래대로 바꿈.

#4월 16일 apache1

라즈베리파이 : 임베디드 컴퓨터

임베디드 보드(시스템) 특징 : low cost(저비용), minimal(소형), networked(네트워크에 연결되어 있어야 함), mobility(이동성)

IoT(Internet of Things) : 사물인터넷, 웹 서버를 통해서 정보를 주고 받고 가능.

LAMP (Linux, Apache, My SQL, PHP)

-apache : 웹 서버 만들어주는 프로그램

-My SQL : 데이터베이스 프로그램

-PHP : 전처리기(html, 동영상이나 이미지의 처리나 인터페이스에 부족->보완 위해 나옴)

-이 셋을 합쳐 AMP 혹은 APM 이라고 함. 윈도우 경우 리눅스 서버 설치 안해도 APM setup 하면 오픈 소스여서 웹페이지 제작 가능.

Apache : 웹 페이지를 제공해주는 웹 서버 어플리케이션. 웹 서버를 잘하려면 통신을 잘해야 하고 여러가지 웹 브라우저와 서버의 종류에 호환이 되어야 함. 클라이언트(웹 브라우저, languae) 와 서버(리눅스, 윈도우 서버 등등)의 미들웨어(중간매개체) 역할을 해줌. 일종의 매니저. static(정적인) HTML 파일을 HTTP 통해서 제공받을 수 있음. 추가적인 모듈들이 있고 다이나믹한 웹페이지들은 (HTML5 나오기 전에는) PHP, ASP 같은 전처리 언어로 보완해서 제공함.

(+HTTP : Hyper Text Transfer Protocol, Hyper Text=홈페이지를 만드는 언어)

sudo apt-get install apache2 -y : apache 설치. 설치 끝나면 라즈베리파이는 웹 서버가 됨.

와이파이 주소가 웹 서버 주소가 됨. (hostname -I, ifconfig 로 주소 파악 가능)

http://localhost or http://192.168.1.1(해당 ip 주소) 치면 apache2 기본 홈페이지 나옴.

/var/www/html/intex.html 이 파일이 화면에 뜨는 파일, 이걸 웹브라우저로 읽으면 됨.

(홈페이지 고치고 싶으면 해당 파일 내용 다 지우고 Hello name 등등 입력하면 됨)

#4월 21일 apache2

localhost : 자신의 ip주소를 http가 읽어 옴.

ls : 폴더 안에 어떤 파일들이 있는지

ls -al : 해당 폴더 안의 파일의 크기나 생성날짜 보려면 입력.

cat 파일이름 : 파일의 내용을 보기 위한 명령어

sudo cp index.html index.html.bak : index.html 파일을 관리자 권한으로 복사해서 백업파일 만들기

홈페이지 고치기 코드

```
<html>
hello kanghee Lee
</html>
```

sudo chmod 666 index.html : 관리자 아니어도 write 할 수 있는 권한.

#4월 23일 php 1부

PHP : preprocessor(전처리기), 전처리 전용 언어. 사용자가 서버에 요청을 했을 때, 서버에 가기 전에 php가 먼저 요청을 받아서 처리를 해 줌. 처리 나온 결과를 서버에 요청, 홈페이지에 결과 보여줌. 페이지에 보여줄 필요가 있는 것을 먼저 실행하고 동작 시킨 다음에 그 브라우저에 보냄.

(어떤 요청을 했을 때 그 결과를 서버에 보내는데 요청 결과를 서버에서 최종적으로 제대로 처리를 못하면 PHP에서 먼저 처리 후 결과를 서버에 보내주면 서버는 이 결과를 웹브라우저로 보내줌 -> 기존의 static HTML 보다 다이나믹 HTML 효과를 누릴 수 있음.)

(웹브라우저(클라이언트)) (Apache 서버) (php 프로그램) 있을 때 웹브라우저가 html만 요청하면 바로 그 결과를 서버가 보내주지만, 클라이언트에서 php프로그램을 짜서 주면 apache는 그것을 수행 못하므로 php에게 넘김, php가 결과를 apache에게 넘겨주면 apache가 결과를 출력해줌. 결국은 웹브라우저를 대면하는 것은 apache 서버지만 apache는 정적 html파일만 결과를 보내줄 수

있기 때문에 동적 html 들어오면 php에게 맡김. 클라이언트는 apache서버가 다한다고 느낌)

+초창기 때 이랬고 현재는 동적인 계산 가능한 html5가 발전됨.

가장 대표적인 PHP example : wordpress, facebook, Wikipedia -> 다 데이터베이스 필요로 함.

index.php 에는 html, php 다 들어갈 수 있음. 상위호환임.

phpInfo() : php정보 불러오는 함수.

#4월 23일 php 2부

PHP = Personal Homepage Tools 였는데 발전하면서 Personal Hypertext Preprocessor

sudo apt-get install php 7.3 -y (원래 7.0으로 소개됐으나 7.3버전이 최신)

/var/www/html/ 에 index.html 파일 있으므로 index.php 만듦.

sudo touch index.php (touch 명령어 -> 빈 파일 만듦)

sudo chmod 666 index.php (편집 가능하게 해줌, 아래 코드 작성)

```
<?php
        echo "Hellp PHP7.3!!!";   //echo=printf
        phpInfo(); //php정보 불러오는 함수
?>
```

(저장 후*) sudo service apache2 restart* (새로운 php 적용)

(index.html 파일과 index.php 파일 함께 있으면 html 파일을 먼저 인식하기 때문에 수정한 결과로 안 나옴 -> index.html을 제거해줘야 함)

sudo mv index.html index.html.original (mv 명령어로 파일 옮겨 줌)

한번더 *sudo service apache2 restart*

+192.168.0.25/test.html 하면 test.html 만든 거로 읽어 옴 (강의에서 a+b=10)

php로 a+b 함수 짜면

```
<?php
        $a = 3;
        $b = 7;
        $sum = $a + $b;
        echo "a + b = $sum";
?>
```

html로 하드코딩하던, php로 실제 계산을 했던 사용자는 모름.

+마지막에 sudo mv 해도 index.html.orignal 파일을 html로 읽어오기 때문에 아예 제거해줌.

sudo rm index.html index.html.original

#4월30일 MySQL

my SQL : php와 호환이 잘되는, 인터랙션 잘 해줌.

sudo apt-get install mysql-server php7.3-mysql

- ➔ 오류 떠서 ***sudo apt-get install mariadb-server-10.0*** *-y* (mariadb : mysql에서 쓰는 DB 형태의 일종.)

sudo apt-get install php7.3 -mysql (원래 코드 작성하면 설치하는 것이 업데이트 되어 mairadb 10.0 설치하고 php mysql을 따로따로 설치함.)

sudo mysql_secure_installation

password랑 id 둘다 root 로 입력. 암호바꾸겠습니까? 등등 질문 나오면 n 입력

Thanks for using MariaDB! 뜨면 설치 완료.

sudo mysql -uroot -p (mysql의 유저가 root, 엔터치고 비밀번호 root 입력. mariadb에 접속 완료)

(wordpress라는 웹 저작 도구 이용해서 홈페이지 만들기 위해 DB를 먼저 생성)

MariaDB [(none)]> create database wordpress; (wordpress란 데이터 베이스 하나 만듦)

MariaDB [(none)]> show databases; (database 정보 보여달라)

crtl+z 해서 나옴.

sudo service apache2 restart (php가 mysql 다루니까 php부분이 업데이트 되어서 apache도 업데이트해줌)

(mysql 설치 완료)

#5월 5일 wordpress로 종합홈페이지 만들기

cd /var/www/html/ → *sudo rm ** (폴더 안의 내용 싹 다 지워라)

sudo wget http://wordpress.org/lastest.tar.gz (터미널을 이용해서 어떤 파일을 다운받고 싶을 때 wget 사용, 최신으로 압축된 파일 다운(tar:묶어주는 것, gz:압축파일))

sudo tar xzf lastest.tar.gz (압축파일 해제, 하고 나면 wordpress 라는 폴더째로 풀림)

(wordpress 폴더 안에 있는 내용들 html 폴더 안으로 다 옮김. index.php가 wordpress 가 아닌 html 폴더 밑에 있어야 함.)

sudo mv worldpress/ .* (worldpress안에 있는 모든 파일들(*)을 현재 폴더(.)로 옮겨라)

sudo rm -rf wordpress latest.tar.gz (-rf : recursive하게 폴더 안의 모든 내용들 강제로 삭제, wordpress폴더 지우고 lastest 압축 파일도 삭제)

(wordpress 폴더 안에 있던 파일들은 오픈 소스여서 파일의 소유자가 nobody, 소유그룹이 nogroup 임. 이걸 접속이 가능하도록 소유자와 소유그룹 바꿔야함.)

sudo chown -R www.data: . (chown은 나중에 알려주심, 소유자와 소유그룹을 www.data로 바꿔줌)

sudo mysql 로 mysql 실행

MariaDB [(none)]> ***GRANT ALL PRIVILEGES ON wordpress. * TO "root"@"localhost" IDENTIFIED BY "root";*** (mysql 문법이어서 설명 생략, 엔터쳤을 때 Query OK 뜨면 제대로 된 거, mysql에 외부 접근 권한을 추가한 것 localhost 의 root에게 wordpress의 파일 DB에 접근할 수 있는 권한 부여)

MariaDB [(none)]> quit 해서 나옴.

브라우저 실행(http://localhost) → 영어로 설정

→ 1. Database name = wordpress 2. Database username = root 3. Database password = root 기억 후 접속 → 각각 입력해주고 submit → All right~ 뜨면 Run the installation → Welcome 창에서 Site Title, username, Password, your email 작성 후 Install WordPress → Success! 뜨면 wordpress 플랫폼을 만들어 준 것임. username과 password 입력 후 로그인 하면 wordpress 홈페이지 만들 수 있음. → Dashboard에서 Quick Draft 에서 Title 입력 후 Save Draft → Customize Your Site → 여러 설정들 바꾸고 publish 누르면 Site Title을 가진 홈페이지가 만들어짐.

(정리하면 : **처음 리눅스 상에서 apache2, php 설치(php가 양방향으로 움직임, apache에서 할 수 없는 동적인 데이터 전처리해주고 데이터베이스 연결해 줌), mysql설치 하여 데이터베이스 사용. wordpress 다운받고, 마리아 DB와 wordpress를 연결해서 root에서 접근할 수 있게끔 해줌)**

#5월 12일 Basic Linux

리눅스는 유닉스에서 시작됨. 유닉스를 pc로 옮긴 것이 리눅스.

리처드 스톨만의 "GNU project" : 오픈 소스의 시초

#5월 14일 vi gcc

vi : 아주 low level의 텍스트에디터. 유닉스 시절 마우스도 흔치 않았기에 키보드로만 할 수 있게 한 방식.

Starting Vi : sudo vi filename(새로 파일을 만들거나 기존 파일 열기)

Vi mode

-Command Mode : esc 키 눌러서 명령어를 주는 것. 진입, 저장, 검색, 이동, 나가기 등

-Insert Mode : a (append)나 i(insert) 키 눌러서 글자 입력. 평상시에 쓰는 mode.

Exiting the Vi Editor

(esc 키 누르고 : 누르기)

:q quit the editor :q! 파일 저장 없이 나가기

:wq 저장하고 나가기 :w 저장만 :wq! 강제로 저장하고 나가기

esc + dd : 한 줄 전체 지우기 esc + 2dd : 두 줄씩 지우기

esc + i or a: 글자 입력. (i는 그 자리에서, a는 한 칸 옆에서)

esc + x : 문장 맨 뒤에서 누르면 백스페이스, 문장 중간에서 누르면 delete키 기능

esc + l : 오른쪽 방향키, h : 왼쪽 방향키, j : 아래쪽 방향키, k: 위쪽 방향키

esc + u : undo esc +p: 붙여 넣기 esc + yy : 한 줄을 복사

:(숫자) 그 숫자 라인넘버로 이동. /(단어) : 커서 뒤부터 그 단어 탐색

?(단어) : 커서 앞부터 그 단어 탐색 n 누르면 단어 계속 탐색

권한 읽기

-rwxr--r-- 1 root root 64 June 14 10:01 address.txt

- rwx r-- r-- 1 root root 64 June 14 10:01 address.txt

끊어서 읽기 (d면 directo ry -면 그냥 파일, user, 그룹, 게스트 접근권한, 소유권한)

쓸데없이 만들어진 파일은 sudo rm -rf filename 으로 삭제

#5월 19일, 21일 리눅스 기본 명령어

date : 현재 날짜 출력

time ls : ls 명령어를 수행한 시간 계산

cal : 현재 월의 달력 출력 cal -y : 올해 12개월의 달력 출력

man(manual) : 특정 명령어의 설명서를 출력. ex. man ls : ls 명령어에 대한 설명서 출력.

history : 히스토리 출력. 여태까지 쳤던 명령어들 순차적으로 보여줌.

whereis : 실행파일, 소스, man페이지의 위치를 알려줌.

useradd : 새로운 사용자를 생성.

useradd -g icqa21 -d /home/icqa21 linux (기본 그룹은 icqa21 , 홈디렉토리는 /home/icqa21 인 사용자를 새로 만듦.

userdel : 사용자를 삭제. userdel -r linux

(linux라는 사용자를 삭제하되 사용자 메일과 홈디렉토리까지 모두 삭제)

usermod : 사용자 정보를 수정.

passwd : 사용자의 패스워드를 변경. 현재 로그인한 자기 자신의 패스워드 변경

passwd linux : linux 패스워드 변경

ls(list) : 파일 및 디렉토리 리스트 출력

ls -al /etc : /etc 디렉토리의 파일 및 디렉토리를 출력하되 숨김파일까지 자세히 출력 (a는 숨김파일까지, l은 디렉토리나 파일들의 자세한 정보 출력)

cat : 파일 안의 내용 출력.

file : 특정 파일이 어떤 종류의 파일인지 알려줌.

more : 파일의 내용을 한 화면씩 끊어서 출력. cat과 비슷

(ls -l /etc | more : /etc 디렉토리의 리스트들을 한 화면씩 출력.

cp : 파일 및 디렉토리 복사 (cp -R /etc . : /etc 하위 디렉토리를 포함한 파일 및 디렉토리를 현재 디렉토리로 복사. -R 옵션 중요) (sudo cp name1.txt work/name2.txt 하면 name1을 work 폴더로 새롭게 복사) (sudo cp (현재파일) (복사할파일or위치) 이 때 절대경로를 직접 쳐도 되고 ../name.txt 이런 식으로 상대 경로 쳐도 됨. 해당 폴더 안의 모든 파일을 복사하고 싶으면 cp ../* 로 가능 파일 전체를 지칭할 땐 *)

mv : 파일 및 디렉토리 이동

rm : 파일 및 디렉토리 삭제 (rm -rf : 전체 파일 다 삭제, 디렉토리 단독으로 삭제불가능해서)

find : 특정 파일 검색. (find /etc -name passwd -print : /etc 디렉토리 하위에서 passwd라는 이름을 가진 파일 및 디렉토리를 모두 찾아서 화면에 출력.)

. (현재 디렉토리) .. (상위 디렉토리)

pwd : 내가 지금 있는 위치(현재 작업 디렉토리) 출력

mkdir : 디렉토리 생성　　rmdir : 디렉토리 삭제 (rmdir -rf 하면 전체 파일 다 삭제)

cd : 작업 디렉토리 위치 변경.

cd /home : 루트 기준으로 처음부터, cd home : 현재 경로에 있는 것부터.

cd .. : 상위 디렉토리로 이동

grep : 특정 파일 내에서 정규표현식을 사용하여 문자열을 검색.

(grep lee name.txt 하면 name.txt 안에 있는 lee 찾아서 표시.)

ls -al | grep root : 출력될 값들 중 root가 포함된 것 찾아서 표시.)

df : 하드디스크 용량 출력

du : 특정 디렉토리의 용량 출력

ifconfig : 네트워크 정보 출력

ping : 상대방 컴퓨터와 네트워크 통신이 가능한지 체크함.

chmod : read, write, excute 모드를 바꾸는 명령.

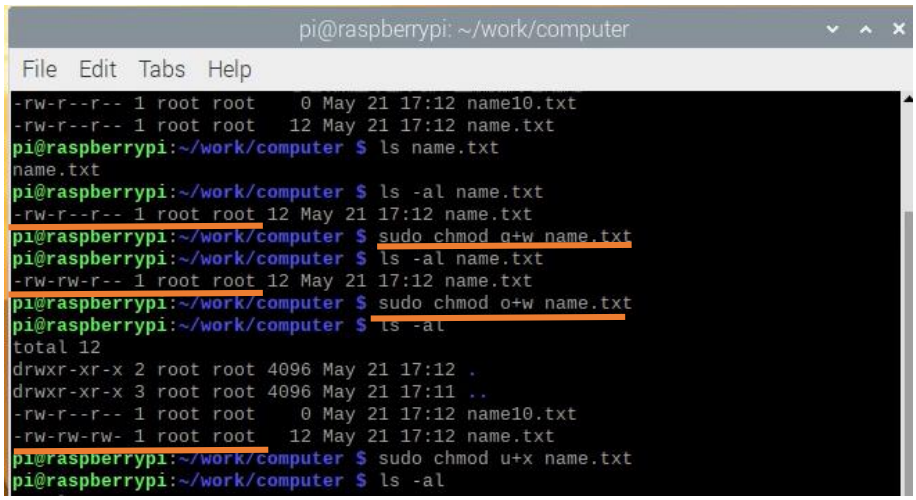
문자열 모드

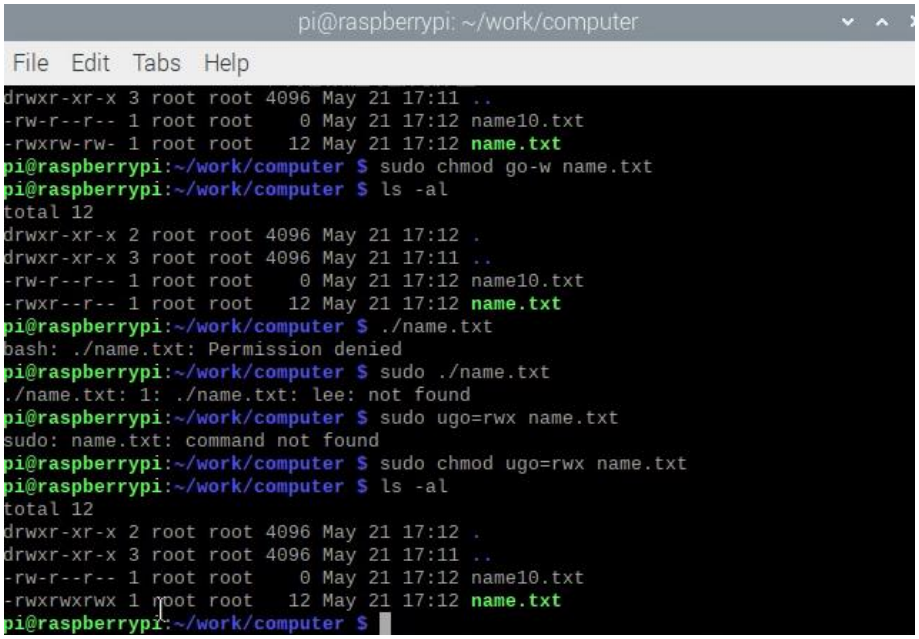
레퍼런스	클래스	설명
u	사용자	파일의 소유자
g	그룹	그 파일의 그룹 멤버인 사용자
o	다른 사람들	그 파일의 소유자나 혹은 그 그룹의 멤버가 아닌 사용자
a	모든 사람	위의 셋 모두, "ugo"와 같다

chmod 프로그램은 파일의 모드들이 어떻게 조정될 수 있는지를 명시하기 위해서 연산자를

연산자	설명
+	지정된 모드들은 지정된 클래스들에 더한다
-	지정된 클래스들로부터 지정된 모드들은 지운다
=	지정된 클래스들을 위해서 지정된 모드들이 정확한 모드들로 만들어지게 된다

모드	이름	
r	읽기 (read)	파일을 읽거나 디렉터리 안 내용물의 리스트를 보여준다
w	쓰기 (write)	파일이나 디렉터리에 쓴다
x	실행하기 (excute)	파일을 실행하거나 디렉터리 트리로 되돌아간다





8진법 모드

rwx = 1 1 1 (=7)

sudo chmod 755 test.txt → 755 이면 u =7(rwx), g = 5(r-x), o =5(r-x)

chown : 파일이나 디렉토리의 소유권 변경

sudo chown pi sample.txt → root root 에서 pi root로 변경

chgrp : 파일이나 디렉토리의 소유 그룹 변경.

sudo chgrp pi sample.txt → pi root에서 pi pi로 변경

touch : 빈 파일 생성.

#5월26일 제어기활용 GPIO

GPIO = General Purpose Input Output port (범용 인풋아웃풋 포트(=핀들의 집합 단위), 핀들이 input 될 수도 있고 output 될 수도 있음.)

라즈베리파이는 3.3v 단위 output, input 의 경우 센싱을 받아들이는 것. 같은 핀이 input output 적용 가능.

아두이노는 센싱, 모터 돌리기에 특화, 웹서핑이나 os 까는 것들은 라즈베리가 우위.

소형pc라즈베리파이는 usb, 이더넷, hdmi 출력, 영상출력, 사운드 출력 등의 기능을 갖고 있는 하나의 작은 컴퓨터. 가격이 저렴하고 작은 크기이지만 성능에 문제가 없는 완전한 컴퓨터. 더욱이 단일 마이크로세서와 같이 입출력 신호를 제어할 수 있는 포트가 있음. 범용적인 목적으로 입/출력을 담당하는 GPIO를 갖고 있으며, SPI 통신, I2C(I square C), UART 통신 (Serial 통신= 직렬통신)

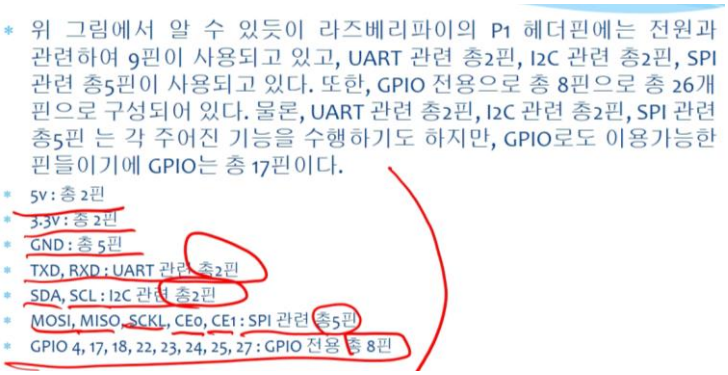
갖추고 있음. 특히 SPI 통신(Serial Peripheral Interface 직렬 주변 통신)으로 ADConvert 하면서 고급 센서 쓸 수 있게 됨.

현재 라즈베리파이 모델 B의 경우 외부핀은 26핀으로 구성. 전원(5V(아두이노 2번, 4번), 3.3V(라즈베리 1번, 17번)), GND(0V, 6번, 9번, 14번, 20번, 25번) 어딨는지 파악.

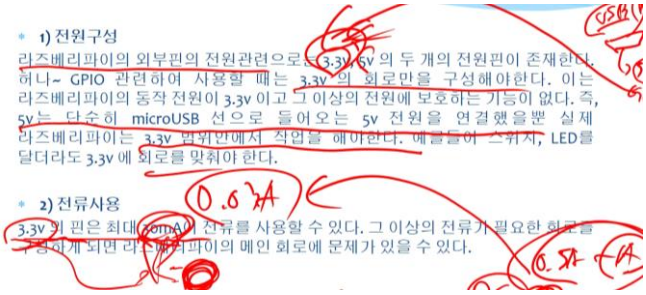
GND(GROUND) : 0V, 어떤 전압이 통한 뒤 빠져나갈 구멍이 반드시 있어야 함. 그래야 전류가 높은 곳에서 낮은 곳으로 흐름.

SPI 통신 : ADC 연결해야 고급 센서 사용 가능. MOST(19번), MISO(21번), SCKL(23번), CE0(24번), CE1(26번) 필요

이렇게 26개 중에 전원, GND, 통신에 필요한 핀들 빼고 나면 8개의 범용핀(GPIO)이 남음. 웬만하면 통신핀들 말고 남아있는 핀들을 써라. (라즈베리파이 3부터 40개로 전체 핀 늘어남)



주의사항(전원구성, 전류사용, 과전압과전류, 버전별 핀배열)



#5월26일 라즈베리파이DRAM관점에서 컴퓨터메모리구조 이해

DRAM : 반도체, 트랜지스터(TR)의 집합체

-데이터를 저장하는 Memory cell array와 I/O 등의 컨트롤러로 구성

-Refresh를 위한 회로 필요 → 전반적인 회로 구성이 복잡.

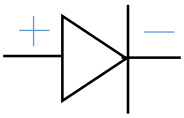
-그럼에도 저렴한 가격으로 제작 가능

-무어의 법칙 : 24개월 동안 연산처리 능력이 2배씩 늘어남.

-가장 단순한 형태의 Memory : 1 Tr(트랜지스터) + 1 Cap(Capacitor, 충전지)

#5월28일 output 테스트 (LED 출력, 파이썬)

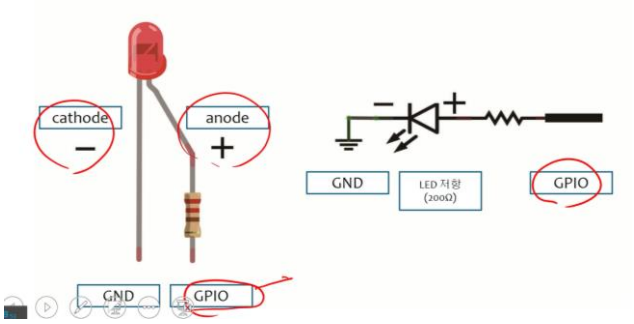
LED 전구 위에서 봤을 때 안 깎인 거 +, 깎인 거 - / 선이 긴 게 +, 짧은 게 -



+ 를 Anode, - 를 Cathode

Light Emitting Diode : 어떤 한 방향으로 전류를 유도해서 흐르게 하는 것. 역방향으로 흐르는 걸 막아 줌.

LED 2개, 저항 220~330옴을 헤더핀 16, 18번에 연결. 회로도는 GPIO의 output 값을 high(3.3v가 라즈베리파이에서 LED로 나가서 그라운드로 돌아오는 것. 전류가 한 번 순환하는 것)로 하면 LED On, Low(0v가 나가는 것)로 설정해주면 Off.



다이오드는 양단 전압이 0.7V로 고정 되어있는데 저항 없이 3.3V 그대로 넣으면 타 버림. 전류를 줄여주는 역할을 저항이 함.

1. 파이썬 프로그램 설치

2. 프로그램 작성 → sudo leafpad output.py 처럼 만듦.

```
import RPi.GPIO as GPIO //GPIO 쓰게끔 라이브러리 import
import time

GPIO.setmode(GPIO.BCM) //GPIO setup

print "Set up LED pins as outputs"

GPIO.setup(23, GPIO.OUT)
GPIO.output(23, False) //False 주는 것 = low = 0v , 꺼져 있음.
GPIO.setup(24, GPIO.OUT)
GPIO.output(24, False)
```

GPIO.output(23, True) //True = high = 3.3v, 켜짐
GPIO.output(24 True)

time.sleep(1) //1초가 지나면

GPIO.output(23, False) //다시 꺼짐
GPIO.output(24, False)

raw_input('press enter to exit program') //프로그램을 빠져나감
GPIO.cleanup() //GPIO 값들을 clear.

3. 루트 권한으로 실행해줘야 함. sudo python output.py

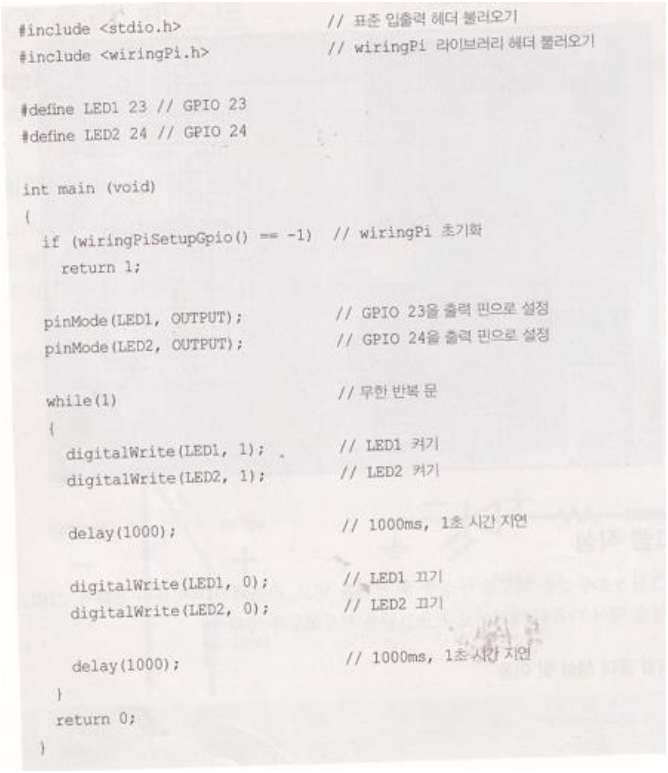
#5월28일 output 테스트 (LED출력, c)

2점식 스위치 안쓰고 4점식 쓰는 이유

1. 브래드보드에 고정.
2. 병렬 회로 구현 가능.

코드 작성

1. gpio -v : gpio 깔렸는지 확인. 있으면 wiringPi 설치 필요x
2. sudo mkdir gpio → cd gpio 해서 작업 폴더 생성
3. sudo leafpad output.c → 프로그램 작성
4. 코드 작성



```c
#include <stdio.h>                  // 표준 입출력 헤더 불러오기
#include <wiringPi.h>               // wiringPi 라이브러리 헤더 불러오기

#define LED1 23 // GPIO 23
#define LED2 24 // GPIO 24

int main (void)
{
  if (wiringPiSetupGpio() == -1)  // wiringPi 초기화
    return 1;

  pinMode(LED1, OUTPUT);          // GPIO 23을 출력 핀으로 설정
  pinMode(LED2, OUTPUT);          // GPIO 24을 출력 핀으로 설정

  while(1)                        // 무한 반복 문
  {
    digitalWrite(LED1, 1);        // LED1 켜기
    digitalWrite(LED2, 1);        // LED2 켜기

    delay(1000);                  // 1000ms, 1초 시간 지연

    digitalWrite(LED1, 0);        // LED1 끄기
    digitalWrite(LED2, 0);        // LED2 끄기

    delay(1000);                  // 1000ms, 1초 시간 지연
  }
  return 0;
}
```

5. sudo gcc -o output.exe output.c -lwiringPi → 프로그램 빌드
6. sudo ./output.exe → 실행

(회로도는 노트에 그렸음)

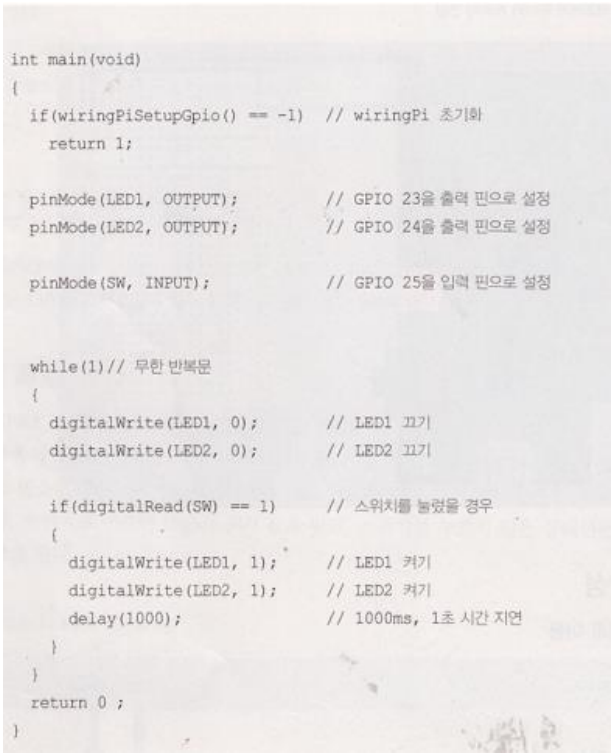
#6월2일 input 테스트 (LED 출력, SW 입력, c, python)

저항의 역할

1. 완충 작용.
2. 전류의 통로 역할.
3. 전압이 걸리게 하는 역할.

코드

```c
#include <stdio.h>
#include <wiringPi.h>
#define LED1 23
#define LED2 24
#define SW 25 //GPIO 25번 연결
```



```
int main(void)
{
  if(wiringPiSetupGpio() == -1)  // wiringPi 초기화
    return 1;

  pinMode(LED1, OUTPUT);         // GPIO 23을 출력 핀으로 설정
  pinMode(LED2, OUTPUT);         // GPIO 24을 출력 핀으로 설정

  pinMode(SW, INPUT);            // GPIO 25을 입력 핀으로 설정


  while(1)// 무한 반복문
  {
    digitalWrite(LED1, 0);       // LED1 끄기
    digitalWrite(LED2, 0);       // LED2 끄기

    if(digitalRead(SW) == 1)     // 스위치를 눌렀을 경우
    {
      digitalWrite(LED1, 1);     // LED1 켜기
      digitalWrite(LED2, 1);     // LED2 켜기
      delay(1000);               // 1000ms, 1초 시간 지연
    }
  }
  return 0 ;
}
```

sudo gcc -o input.exe input.c -lwiringPi → 프로그램 빌드

sudo ./input.exe → 실행

#6월2일, 4일 SPI 통신 1부

SPI 통신 : 라즈베리파이에선 A/D convertor 존재하지 않기 때문에 이를 SPI 통신과 연결해서 외부 센서를 사용.

ADC 칩 MCP3208 (ADC 8채널 12비트) : 아날로그 값을 디지털로. 아날로그 값을 동시에 8개까지 받을 수 있다.(=8채널) 양자화 레벨이 2의 12승.(=12비트, 3.3v/4096 per sample 이므로 정밀하게 측정 가능)

라즈베리파이는 GPIO와 SPI 통신, I2C 통신, UART 통신을 갖고 있지만 ADC 기능이 없어서 센서 사용 불가능. 그래서 MCP3208을 이용해 외부 ADC기능을 사용하도록 함. 이를 위해서는 라즈베리파이와 MCP3208 간에 SPI 통신 이용하여 ADC 데이터 받아야 함.

MCP3208은 8개의 ADC 포트 중에 (CH0~CH7) CH0에 가변저항을 달아 ADC값 테스트

가변저항(=Potential meter, =Variable resistor) : 이 세상에 존재하는 아날로그 센서.(=output으로 전압을 내보내는 센서)와 등가 대치가 가능, 가변 저항도 전압을 바꿔주는 역할. 대표적인 예가 볼륨. 움직임이나 회전에 따라 인가된 저항값이 달라짐.

주의할 점 : 센서, 디지털 회로도 대부분이 5V 기준, 라즈베리는 3.3V로 동작함. 칩도 5V로 동작

하지만 센서에서 나온 것을 직접 칩에 연결하고 이 칩이 라즈베리에 연결이 되는데 자칫하면 라즈베리가 타버릴 수 있음. 그래서 전압 변환 회로인 레벨 컨버터가 필요.

MCP3208의 핀 배치 : 홈이 있는 게 칩의 윗부분. 통상적으로 윗부분의 양 옆이 1번 핀일 확률 높음. 센서 전압이 들어가는 핀들에 여러 센서 넣어도 가능. 대신에 디지털화된 변환된 값 받아들이는 핀은 12번 D out 1개. VDD=VCC .

AGND, DGND(아날로그, 디지털 그라운드)는 접지. 아날로그, 디지털 따로 있는 이유는 디지털의 경우 주로 5V 레벨인데 아날로그는 크고 다양한 전압들 가질 수 있기 때문. 보통 두개 따로 0V를 넣어주지만 큰 의미X, 두개를 묶어주는 게 더 안전할 수 있다.

Dout이 MISO로 들어가고 MOSI가 Din으로 들어감.(라즈베리에서 3.3V로 가는 것이라 문제x)

CLK(Serial Clock=아날로그는 8개 들어올 수 있지만 디지털은 1개 밖에 못 들어오기 때문에 순차적으로 이를 관리할 관리자역할이 필요함. 라즈베리파이와 호흡이 맞아야함.)

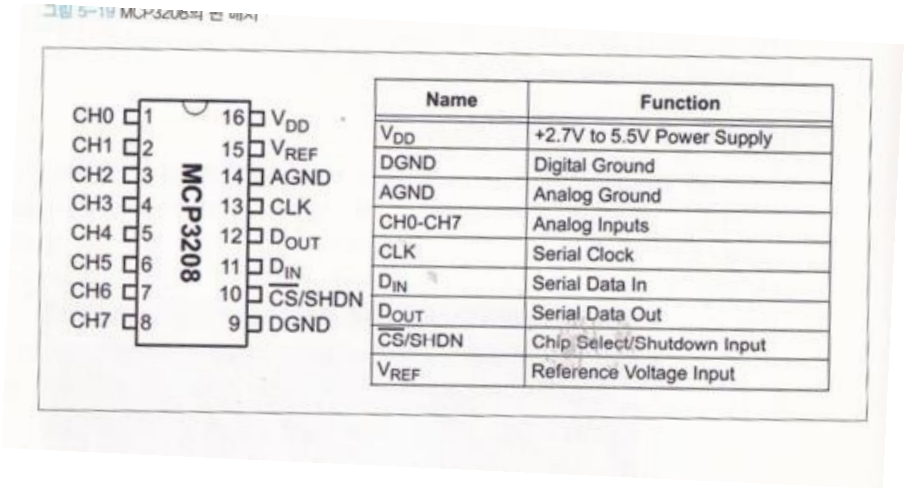
CS(Chip Selector) : 라즈베리파이에서 CS에다가 0V를 주는 순간 AD converting 시작.

CE(Chip Enable)

MISO(Master Input Slave Output) : 하나의 마스터(라즈베리)에 여러개의 slave(mcp) 할 수 있게함.

(Slave에서 출력을 내보내면 master에서 받음=Full duplex 방식)

MOSI(Master Output Slave Input) : 마스터(라즈베리)가 3.3V 내보내면 : MCP3208 Din이 받아들임.



Name	Function
V_{DD}	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
D_{IN}	Serial Data In
D_{OUT}	Serial Data Out
$\overline{CS}$/SHDN	Chip Select/Shutdown Input
V_{REF}	Reference Voltage Input

아두이노에서 ADC하는 함수는 analogread(), 하지만 라즈베리는 없어서 불가능.

그림 5-20 MCP3208의 회로 구성

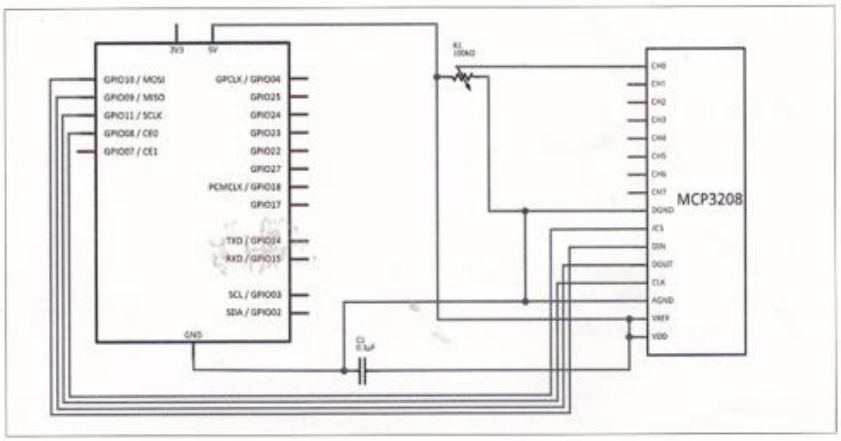
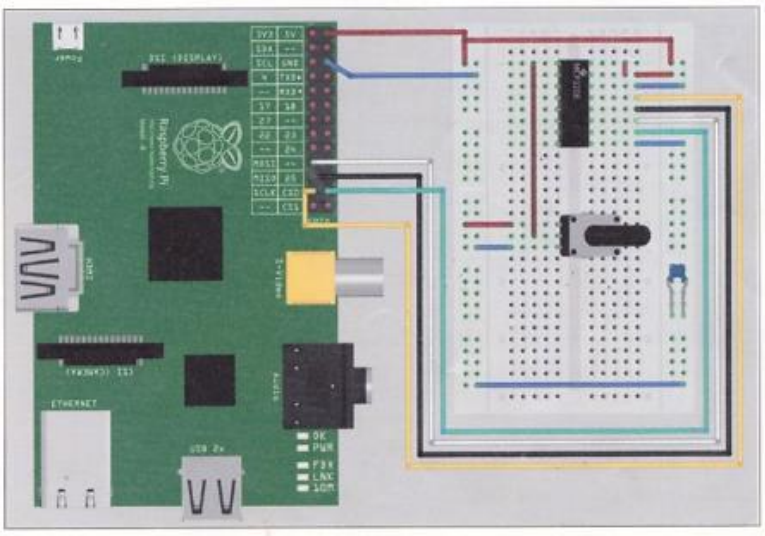


그림 5-21 라즈베리 파이와 MCP3208의 연결



raspi ←-----→ MCP3208 (raspi에서 신호보내면 MCP에서 받음)

CE0	CS
SCLK	CLK
MISO	DOUT
MOSI	DIN
GND	AGND & DGND

센서에서 AOUT <->ch0

문제에서 이거 연결하라하면 회로 연결해서 각각 인덱싱.

<코드 - 16진수를 받아와서 최종적으로 10진수로 변경>

#define CS_MCP3208 8 //CE(chip enable)에 연결된 chip을 동작시킴.

#6월4일 SPI 통신 2부

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringPiSPI.h>

#define CS_MCP3208 8

#define SPI_CHANNEL 0
#define SPI_SPEED 1000000

int read_mcp3208_adc(unsigned char adcChannel)
{
unsigned char buff[3];
int adcValue = 0;

//라즈베리에서 MCP로 buffer이용해서 자료 보냄. 버퍼들은 8비트씩.
buff[0] = 0x06 | ((adcChannel & 0x07) >> 2); // (0x06=16진수)  0x0→4비트, 6→4비트
buff[1] = ((adcChannel & 0x07) << 6);
buff[2] = 0x00;

digitalWrite(CS_MCP3208, 0);

wiringPiSPIDataRW(SPI_CHANNEL, buff, 3); //MCP에서 라즈베리로 보냄.

//라즈베리에서 buffer값 읽는 것.
buff[1] = 0X0F & buff[1];
adcValue = ( buff[1] << 8) | buff[2];

digitalWrite(CS_MCP3208, 1);
return adcValue;
}
```

```
int main (void)
{
int adcChannel = 0;
int adcValue = 0;
if(wiringPiSetupGpio() == -1)
{
fprintf(stdout, "Unable to start wiringPi:  %s₩n", strerror(errno));
return 1;
}
if(wiringPiSPISetup(SPI_CHANNEL, SPI_SPEED) == -1)
{
fprintf(stdout, "wiringPiSPISetup Failed: %s₩n", strerror(errno));
return 1;
}
pinMode (CS_MCP3208, OUTPUT);

while(1)
{
adcValue = read_mcp3208_adc(adcChannel);
printf("adc0 Value = %u₩n", adcValue);
}
return 0;
}
```