



3G 6:59

Embedded Systems

Log In

2018.9.13

임베디드 시스템

컴퓨터공학과 이병문

강의 일정

01 강의소개, 강의일정소개, 평가소개

02 사물인터넷, 라즈베리파이3 설치/구축

- 임베디드시스템/사물인터넷
- Raspbian 설치, ECMA Script 7

03 임베디드 액츄레이터/센서 제어 1

- Raspbian 환경설정
- Node.js 프레임워크 구축
- LED 1/3/7 color, Button 제어

04 임베디드 액츄레이터/센서 제어 2

- Buzzer ■ Light sensor
- Touch sensor

05 임베디드 액츄레이터/센서 제어 3

- Relay제어, FAN제어
- wiringPi GPIO API, 인터럽트제어
- PWM제어

06 임베디드 액츄레이터/센서 제어 4

- 초음파센서(거리측정) 제어
- 사운드센서, 온습도센서 제어

07 임베디드 액츄레이터/센서 제어 5

- 홀마그네틱센서, 레이저출력제어,
- Tracking센서제어

08 중간고사

임베디드 시스템 / 사물인터넷

■ IoT 서비스 구조

☑ 사물인터넷(IoT)의 3대 구성요소

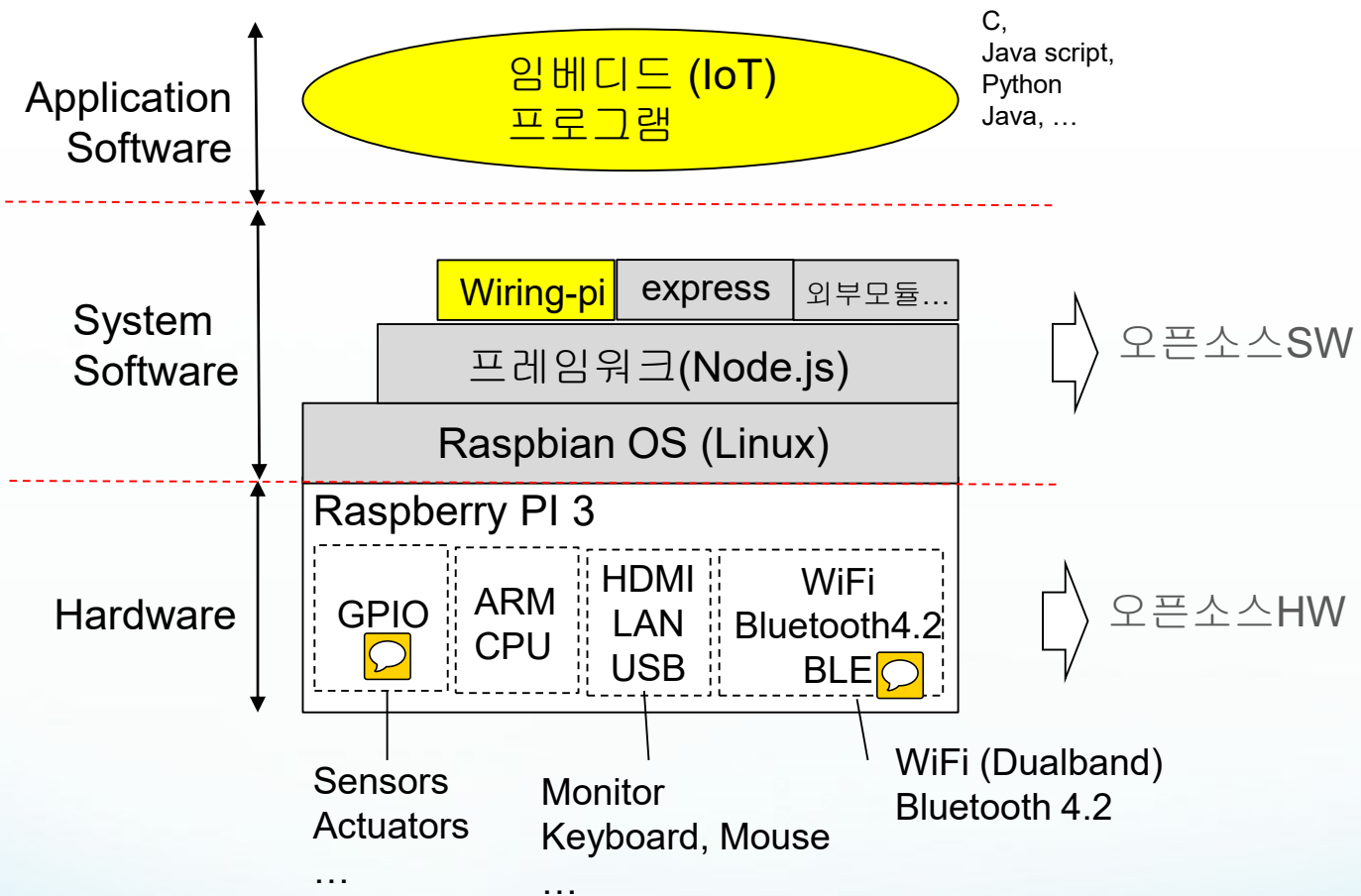
- IoT기기, IoT 네트워크플랫폼, IoT 서버



임베디드 시스템 / 사물인터넷

■ 실습환경(개발환경)

☑ 하드웨어 + OS + 응용프로그램



임베디드 시스템 / 사물인터넷

■ 오픈소스 HW(개방형 하드웨어 플랫폼 ▶ IoT하드웨어플랫폼)

- ☑ (하드웨어 설계도가 공개된) 소형 단일보드형태의 마이크로 컨트롤러
- ☑ 성능 개선과 활용도의 증가로 최근 2~3년간 판매량 급증 추세
- ☑ 아두이노와 라즈베리파이 각각 출시 이후 누적 100만대 이상의 판매실적 기록
- ☑ 오픈소스 하드웨어 플랫폼개발에 인텔, 퀄컴, 삼성전자 등 경쟁(예, '갈릴레오 보드')
- ☑ IoT 기기의 표준 하드웨어로 경쟁제품 다변화

아두이노	라즈베리 파이	비글본 블랙	갈릴레오 보드
			
'05년 이탈리아에서 시작 Atmel의 AVR CPU 탑재 가장 널리 활용되고 있음	영국 라즈베리 파이 재단 이 '12년 정식 출시한 초 소형 싱글 보드 컴퓨터	초기 비글보드를 거쳐 '13 년 Texas Instrument 출시 가장 뛰어난 성능	인텔-아두이노 합작 개발 저전력 소형 코어 제품인 Quark 칩 장착

출처: Wikipedia

임베디드 시스템 / 사물인터넷

■ Single Board Computer

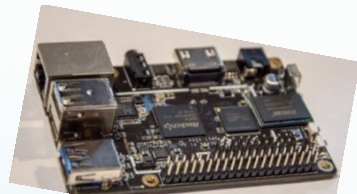
☑ Raspberry Pi 의 대체보드들



1위, Asus Tinker Board S



2위, Odroid-XU4



3위, Rock64 Media Board

Asus Tinker
Board S

Giga Ethernet, 2GB RAM, GPU,

Rockchip RK3288 (4x
Cortex-A17 @ 1.8GHz)

Mali-T760

2GB LPDDR3
RAM

\$89

Odroid-XU4

Samsung Exynos5422 (4x
Cortex-A15 @ 2.0GHz and
4x Cortex-A7 @ 1.4GHz)
Ubuntu, Android4.4, 5.0,7.1,
USB 3.0 and Gigabit Ethernet

Mali-T628
MP6

2GB LPDDR3
RAM; optional
up to 64GB
eMMC

\$59

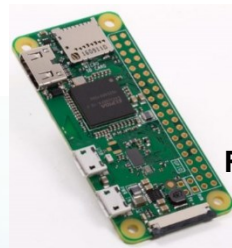
Rock64 Media
Board

Rockchip RK3328 (4x
Cortex-A53 @ 1.5GHz)

Mali-450
MP2

1GB, 2GB, or
4GB DDR3L;
empty eMMC
slot

\$24.95
(1GB),
\$34.95
(2GB), or
\$44.95
(4GB)



Raspberry pi 0 W

임베디드 시스템 / 사물인터넷

■ IoT 제품 사례

☑ 스마트 IoT 온도조절기

- 네트워크와 연결, 냉난방 온도 조절뿐만 아니라 에너지 사용량 점검과 내·외부의 공기 질 모니터
- 인공지능(AI) 플랫폼과 연동되는 스마트 기능



Amazon 이코비4

https://www.youtube.com/watch?v=Y_doUXfQVYo



<https://www.youtube.com/watch?v=EWZ4xGjEJ2Y>



Google Nest



MS CLAS



<https://www.youtube.com/watch?v=WQ0FZrc2IRA>

임베디드 시스템 / 사물인터넷

■ IoT 제품 동향분석

☑ 블루투스제어 무드등, 스피커



임베디드 시스템 / 사물인터넷

■ IoT 제품 동향분석

☑ 천식환자용 스마트 호흡기 (아스마폴리스)

- 천식환자의 호흡기에 **GPS**, 블루투스를 탑재해 스마트처리기능
- 호흡기를 사용할 때마다 서버로 위치정보를 전송하여 전국의 천식위험도 맵(지도)을 자동생성
- 스쿨버스 노선변경, 천식위험 지역경고알림을 제공하여 천식발병률을 최소화



천식 맵 서버

■ IoT 제품 동향분석

☑ 스마트 약병

- 약먹을 시간이 되면 약병의 **LED**불빛으로 알려줌
- 약을 먹지 않을 경우에 스마트폰이나 약병으로 알려줌
- 복용정보가 약복용 관리서버에 저장되고 관리됨

약 복용관리 Server



IoT device



IoT App server

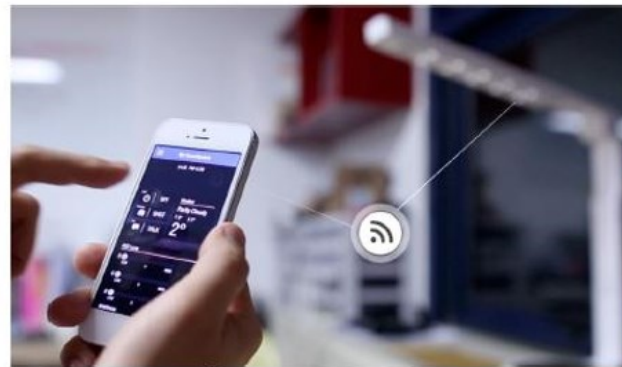
임베디드 시스템 / 사물인터넷

■ IoT 제품 동향분석

☑ 스마트 데스크 램프

- 스탠드(램프)기능, 소리모니터링,
- 스마트폰과 연동하여 설정된 상황을 감시하고 보고하는 기능

<https://youtu.be/eHQ75-ul5KY>



Raspbian 설치

■ 설치방법

1. 준비물

1. Raspberry Pi 3 B+
2. MicroSD 메모리카드 (8GB, 16GB, 32GB)
3. HDMI 모니터, USB키보드, USB마우스
4. 케이블(LAN, HDMI), **USB to Serial Cable**
5. USB 5V 전원어댑터
6. PC 또는 노트북 (인터넷)
7. Breadboard
8. 센서모듈킷
9. 점퍼케이블



USB 키보드 & 마우스세트



UHS Speed Class (UHS bus only)	Speed Class	Min. Constant Write Speed*
UHS Speed Class 3 U3		30MB/s (240Mbps)
UHS Speed Class 1 U1	Speed Class 10 10 Class 10	10MB/s (80Mbps)
	Speed Class 6 6 Class 6	6MB/s (48Mbps)
	Speed Class 4 4 Class 4	4MB/s (32Mbps)
	Speed Class 2 2 Class 2	2MB/s (16Mbps)

Raspbian 설치

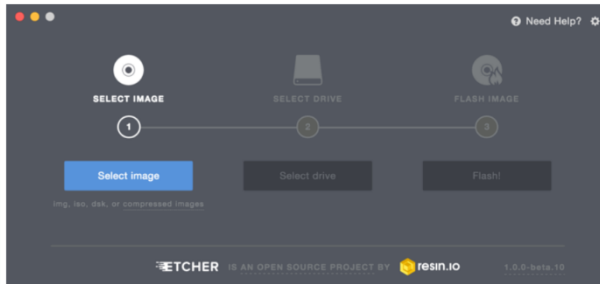
■ 설치방법

2. 필요SW 다운로드

크롬브라우저로

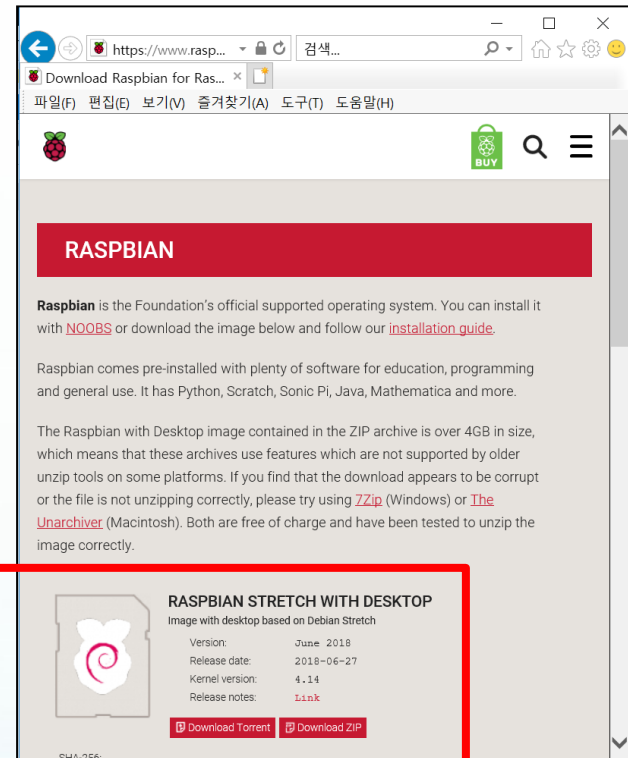
<http://www.etcher.io> 접속하여 “Download for Windows x64” **etcher** 다운로드 V1.4.4-x64

<https://www.raspberrypi.org/downloads/raspbian> 최신버전 다운로드 2018-06-27-raspbian-stretch



- Etcher 사이트 접속오류가 나면
- Chrome 브라우저 사용

*.zip 파일을 다운로드한다.
(압축은 풀지 않는다.)

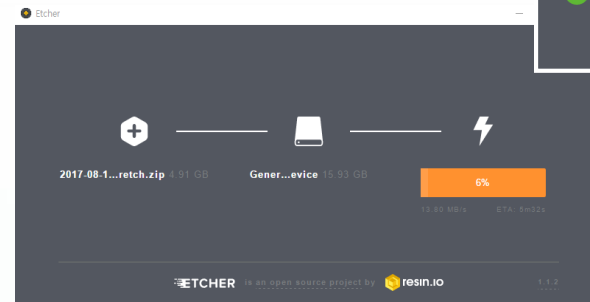
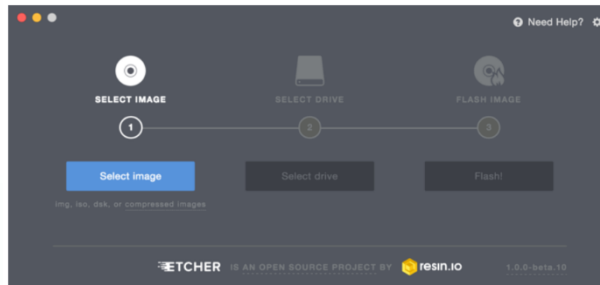


Raspbian 설치

■ 방법 2) 설치방법

3. Etcher 실행 & MicroSD메모리카드 쓰기

- 3.1) MicroSD메모리 카드를 PC에 넣고, (용량확인!, 실습에서는 약16GB)
- 3.2) etcher 를 실행한다.
- 3.3) “Select Image” 클릭하여 raspberry pi 이미지를 선택한 후 flash 한다.
(압축파일을 풀지 않고, 그대로 flash 한다), (다소 시간이 소요됨...)
(혹시라도, Validating 후 오류가 나면 다시 시도할 것!)



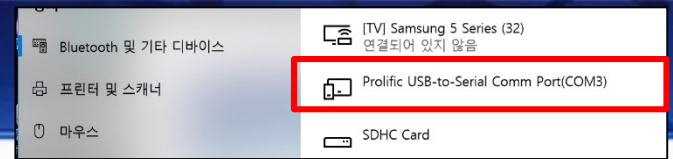
설치하고 나면 windows 에서 인식 불가한다며, 포맷하여야 한다고 나오는데, 포맷하면 안됨!!! (메시지를 무시할 것)

4. 라즈비안(Raspbian)의 시리얼통신(UART) 기능을 활성화시킴

노트북의 모니터/키보드/마우스를 라즈베리파이에서 활용하기 위한 목적

- 4.1) MicroSD카드를 빼었다가, 노트북에 다시 끼운다. (포맷하면 절대안됨!!)
- 4.2) boot 디스크의 폴더를 연다.
- 4.3) 워드패드로 config.txt 파일을 열어서, 파일 맨 끝부분에 다음 내용을 입력하고, 저장한 뒤, 종료한다.

```
enable_uart=1
```



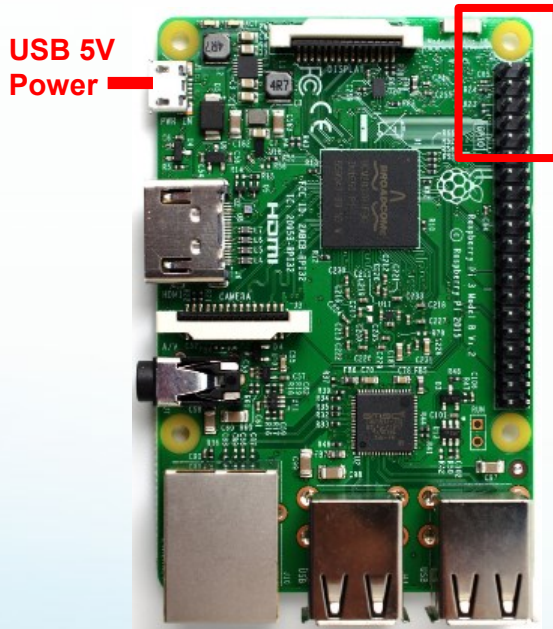
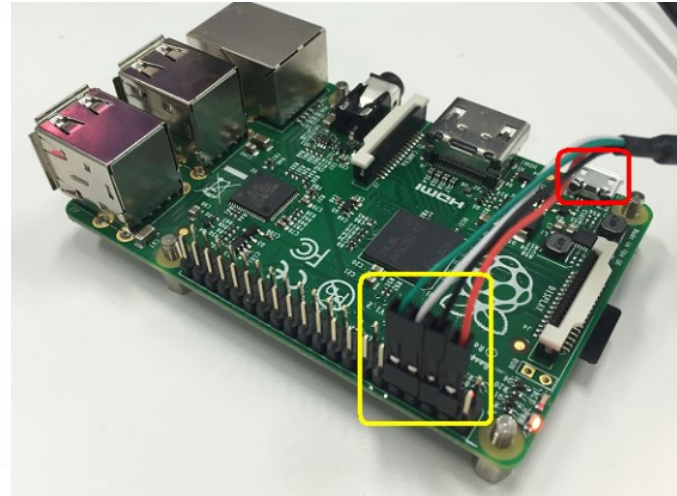
6.4

5. MicroSD메모리카드를 라즈베리파이3 의 메모리슬롯에 정확히 삽입한다.

주의! 삽입할때, 거꾸로 삽입하지 않도록 주의깊게 확인한후 삽입한다.

6. Serial to USB 케이블의 연결

- 6.1) 케이블을 rPi3에 그림처럼 연결한다.
- 6.2) 노트북: 설정->장치를 띄운 뒤
- 6.3) USB를 노트북에 연결한다.
- 6.4) 장치제어기가 자동으로 설치된다.
- 6.5) 다음페이지



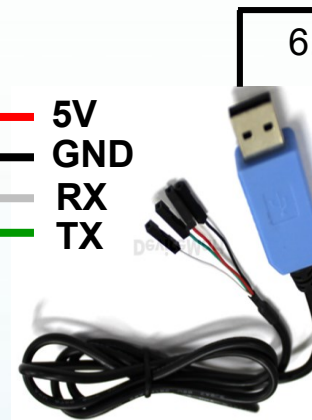
USB 5V
Power



6.1

- | | | |
|----|----|-----|
| 4 | 빨강 | 5V |
| 6 | 검정 | GND |
| 8 | 흰색 | RX |
| 10 | 초록 | TX |

6.3



putty.exe

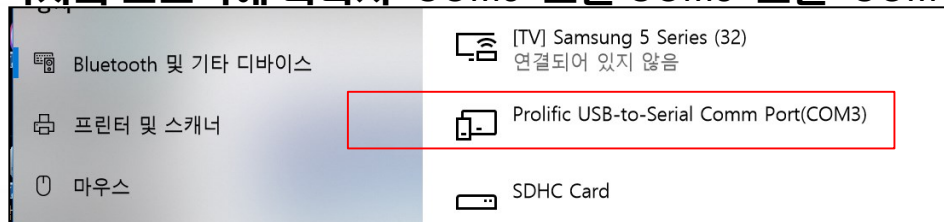


7. 노트북에서 Serial 통신의 설정 및 적용

7.1) 장치제어기가 설치되면 Com Port 가 생성된다.

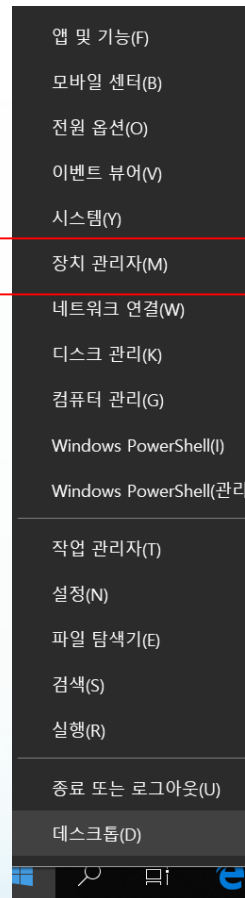
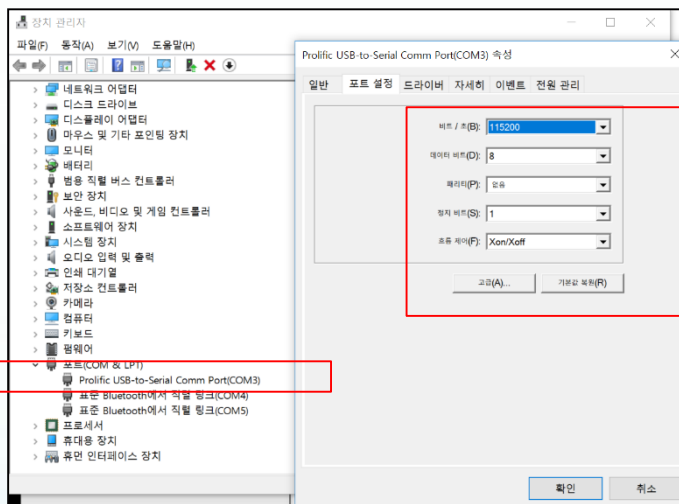
설정->장치 창에서 Prolific USB-to-Serial Comm Port (**COM?**) 을 확인한다.

각자의 노트북에 따라서 COM3 또는 COM8 또는 COM? 으로 되므로 번호를 잘 기억해둔다.



7.2) USB의 장치관리자에서 **COM?** 포트를 아래와 같은 값으로 **포트설정**한다.

- 통신속도(Baud rate) : 115200
- Data bit : 8 bit
- Stop bit : 1 bit
- Parity : None
- Flow control : XON/XOFF



7. 노트북에서 Serial 통신의 설정 및 적용

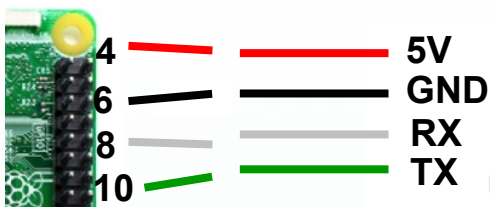
7.3) 네이버에서 putty.exe 를 검색해 바탕화면에 다운로드후 실행시킨다.

7.4) 7.2 에서 설정한 통신초기값과 똑 같이 putty 에 설정한다.

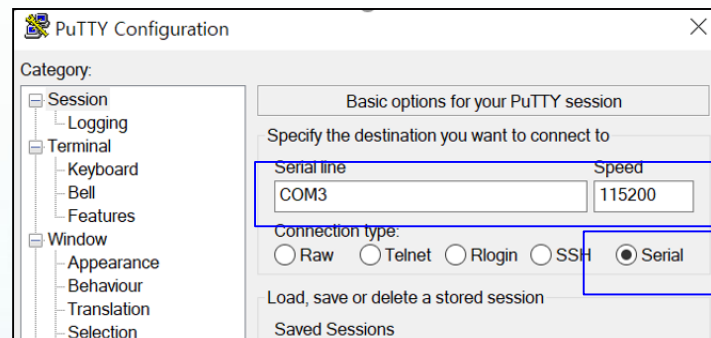
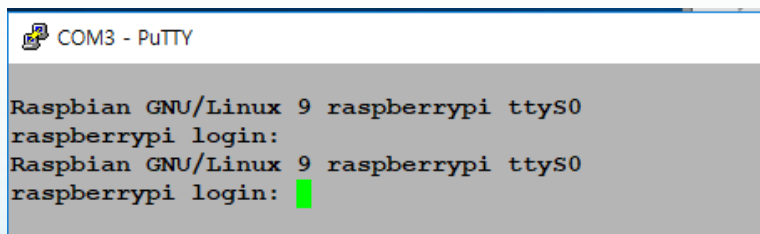
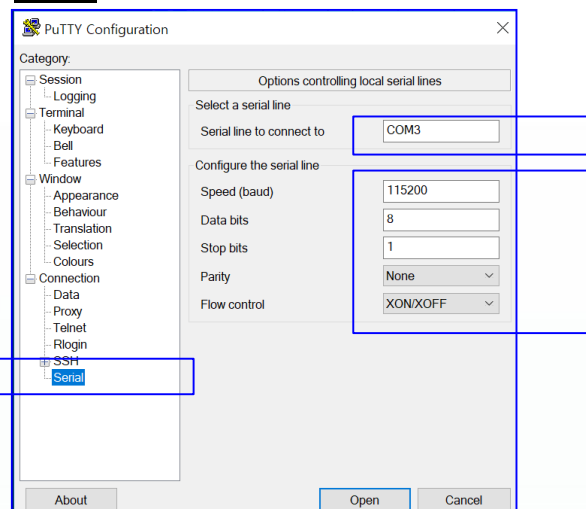
7.5) 설정이 끝나면 putty 에서 시리얼로 라즈베리파이에 접속한다

7.6) 엔터키를 2~3번 치면 접속 로그인 화면이 아래와 같이 뜬다

(만약, 잘 안된다면 엔터키를 여러 번 쳐주면 된다.
그래도 안되면, USB를 빼었다 끼운 후 7.6) 을 다시 한다.)



putty.exe



8. 라즈베리파이 부팅후 로그인 !

8.1) 라즈베리파이에 LAN케이블을 연결한다

8.2) 로그인화면이 나오면 로그인한다 (아이디: pi 초기암호: raspberry)

LAN케이블



USB to Serial



9. 유선LAN설정 !

9.1) 유선LAN을 설정한다.

```
$ sudo raspi-config
```

9.2) 원격접속(ssh)기능을 활성화시킨다.

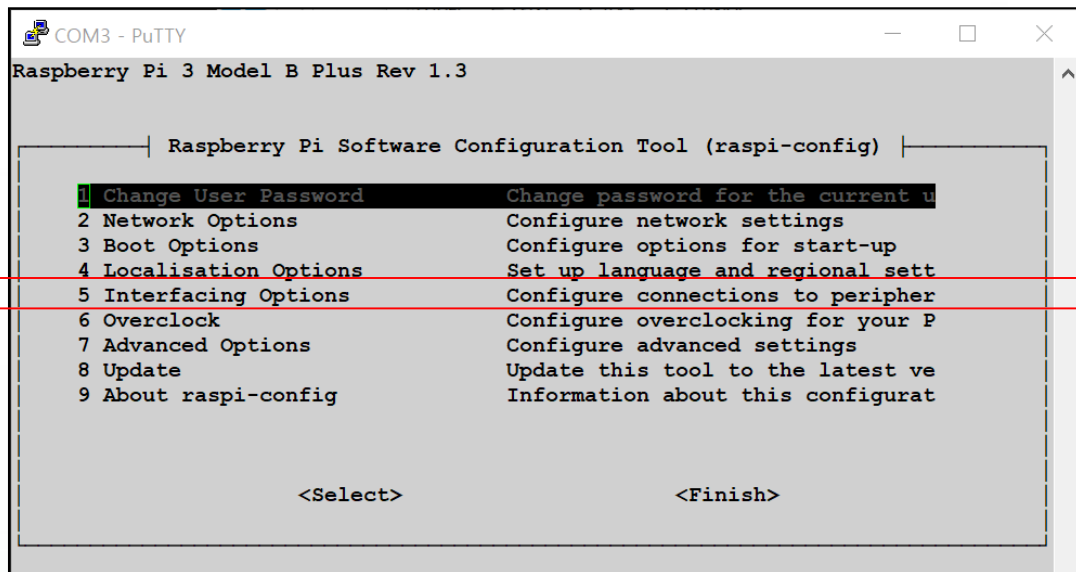
Interface Options -> SSH Server
Enable -> OK

9.3) 원격접속을 하기 위해 필요한

IP주소를 확인하고, 메모할것!!!

```
$ ifconfig eth0
```

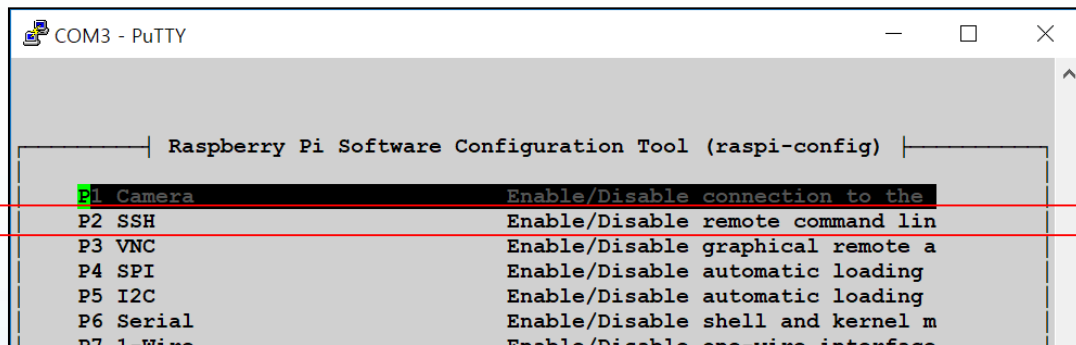
(각자 IP주소가 다르다.
시간이 좀 걸릴 수 있음!)



자신의
라즈베리파이의 IP주소를
메모!!

```
pi@raspberrypi:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.55.95 netmask 255.255.255.0 broadcast 192.168.55.255
    inet6 fe80::a4ef:bac2:f07f:a04b prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:05:2e:35 txqueuelen 1000 (Ethernet)
    RX packets 618 bytes 64370 (62.8 KiB)
    RX errors 0 dropped 149 overruns 0 frame 0
    TX packets 73 bytes 10472 (10.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```



9. 로그인 & 유선LAN설정 !

9.4) putty를 종료하고
 노트북에서 USB를 뺀다!
 (그러면, rPi3의 전원차단됨)
 rPi3에서 시리얼케이블을 제거!

9.5) USB 5V 전원케이블을 연결한다.

9.6) putty 를 다시실행한후
 메모한 IP주소로 ssh 접속하여
 8.2) 과 같이 로그인한다.
 (아이디 pi
 암호:)



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.55.95's password:  
Linux raspberrypi 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Sep  9 07:31:08 2018  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
Wi-Fi is disabled because the country is not set.  
Use raspi-config to set the country before use.  
  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $
```

임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차

```
$ sudo raspi-config
```

The screenshot shows the Raspberry Pi Software Configuration Tool (raspi-config) interface. The main menu lists options 1 through 9. Option 1, 'Change User Password', is highlighted. A callout points to this option with the text 'pi 암호변경 (초기값: raspberry)'. Option 2, 'Network Options', is also highlighted, with a callout pointing to it and the text '라즈베리파이 호스트명 변경 WiFi 설정(SID, 암호)'. Below the main menu, a secondary menu is shown with options I1 through I4. Option I1, 'Change Locale', is highlighted, with a callout pointing to it and the text '다음 페이지 참조'. Option I2, 'Change Timezone', is also highlighted, with a callout pointing to it and the text 'Asia/Seoul'. Option I3, 'Change Keyboard Layout', is highlighted, with a callout pointing to it and the text 'US'. Option I4, 'Change Wi-fi Country', is highlighted, with a callout pointing to it and the text 'US'. A callout points to the bottom of the screen with the text '자동설정됨! 또는'.

Raspberry Pi 3 Model B Plus Rev 1.3

Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password	Change password for the current u
2 Network Options	Configure network settings
3 Boot Options	Configure options for start-up
4 Localisation Options	Set up language and regional sett
5 Interfacing Options	Configure connections to peripher
6 Overclock	Configure overclocking for your P
7 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest ve
9 About raspi-config	Information about this configurat

Raspberry Pi Software Configuration Tool (raspi-config)

I1 Change Locale	Set up language and regional
I2 Change Timezone	Set up timezone to match you
I3 Change Keyboard Layout	Set the keyboard layout to m
I4 Change Wi-fi Country	Set the legal channels used

pi 암호변경 (초기값: raspberry)

라즈베리파이 호스트명 변경
WiFi 설정(SID, 암호)

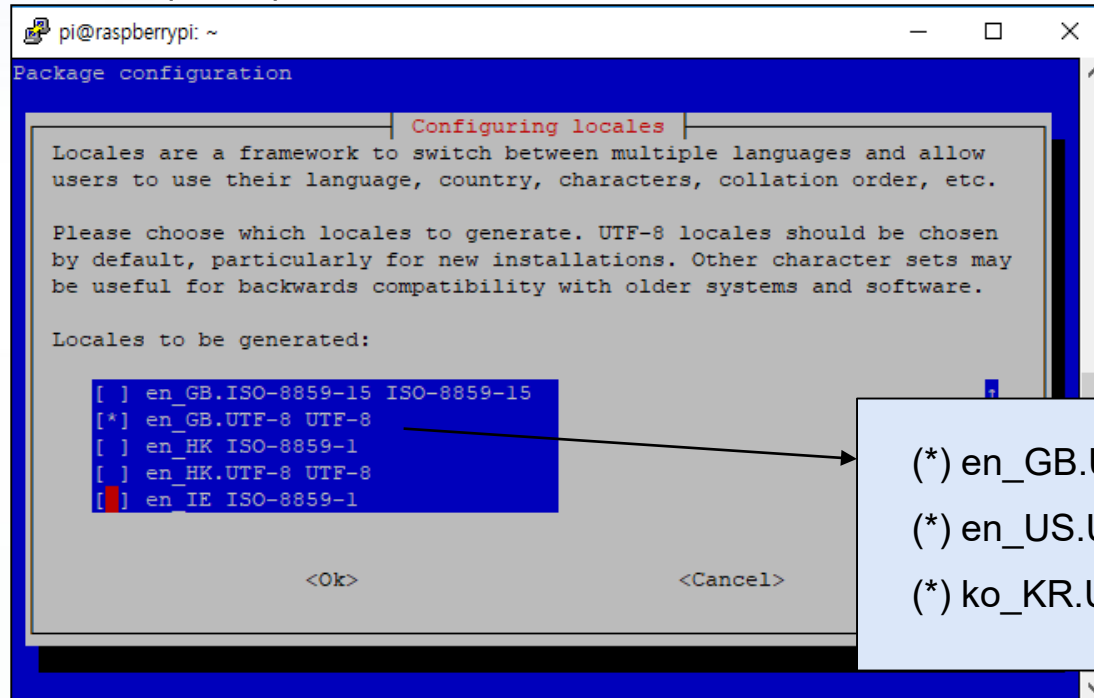
Desktop/CLI 선택
B1 Console 선택

다음 페이지 참조
Asia/Seoul
US

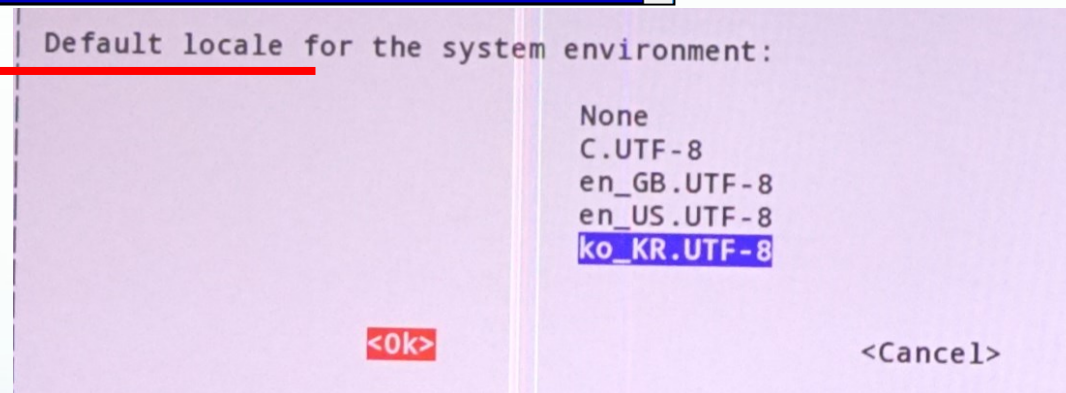
자동설정됨! 또는
Generic 105 key (intl) -> other -> Korean -> Korean (101/104) key Compatible
-> The default ... -> No composite -> No

임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차

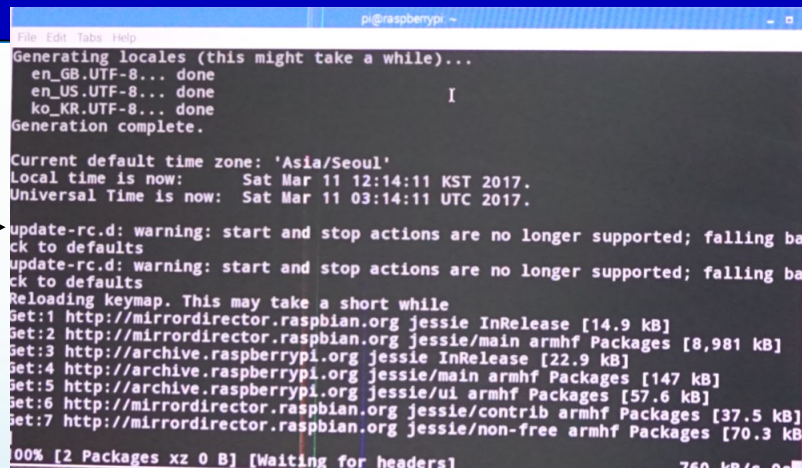
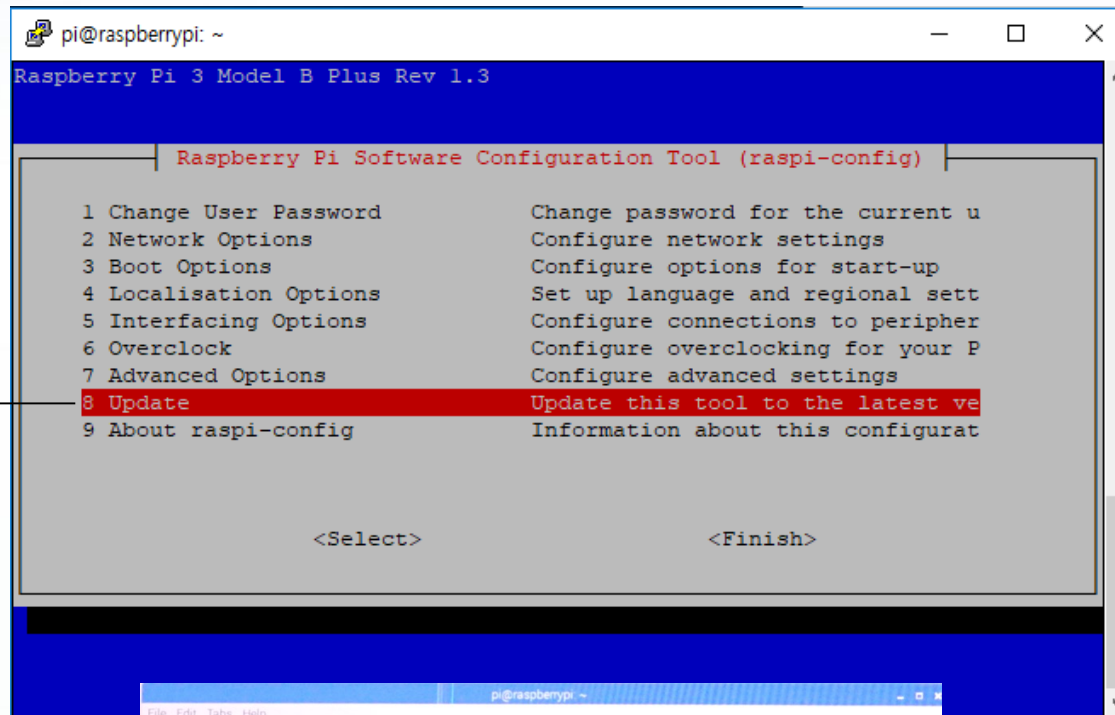


(*) en_GB.UTF-8 UTF-8
(*) en_US.UTF-8 UTF-8
(*) ko_KR.UTF-8 UTF-8



임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차



임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차

☑ 한글설정 방법

1) 한글 SW패키지 설치

```
$ sudo apt-get install ibus  
$ sudo apt-get -y install ibus-hangul  
$ sudo apt-get -y install fonts-unfonts-core
```

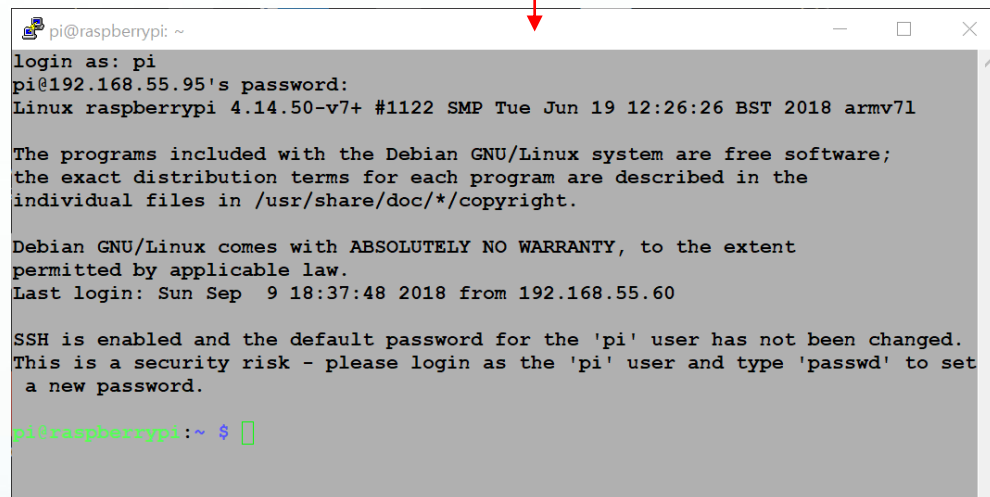
2) 리부팅후, putty 로 다시 로그인

```
$ sudo reboot
```

→ putty 원격 로그인

3) 터미널창에서 한글을 입력한다.

(한영키)



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.55.95's password:  
Linux raspberrypi 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Sep  9 18:37:48 2018 from 192.168.55.60  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
pi@raspberrypi:~ $
```

임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차

☑ 실습계정생성/관리

1) **root** 암호변경 추후 root 권한이 필요한 경우 (예, sudoers 파일오류가 발생하면)

```
$ sudo passwd root
새 UNIX 암호:
$
```

gachon654321

2) 일반사용자 계정생성

```
$ sudo useradd -m -s /bin/bash 학번
$ sudo passwd 학번
```

학번, 예) 20160000

암호, 예) gachon654321

3) 일반사용자의 권한설정

```
$ sudo vi /etc/sudoers
```

```
#
# This file MUST be ....
#
....
```

```
20160000 ALL=(ALL) NOPASSWD: ALL
```

정확히 타이핑하여야 함!
만약에 오타가 날 경우에
추후 sudo 명령이 실행 오류가 남!!

예) 20160000

```
$
```

<ESC> :wq!

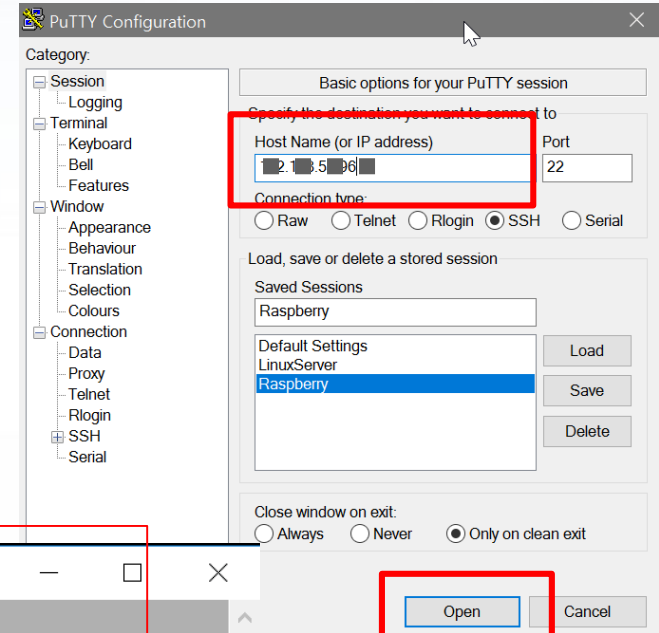
임베디드 시스템 개발환경

4) putty 에서 pi 로그아웃

\$ exit

5) 다시 putty 로 원격로그인

로그인은 방금전에 등록했던 자신의 학번 계정
(예, 20160000 gachon654321)



```
20160000@raspberrypi: ~  
login as: 20160000  
20160000@192.168.55.95's password:  
Linux raspberrypi 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
20160000@raspberrypi:~ $
```

임베디드 시스템 개발환경

■ 실습(개발)환경 구축절차

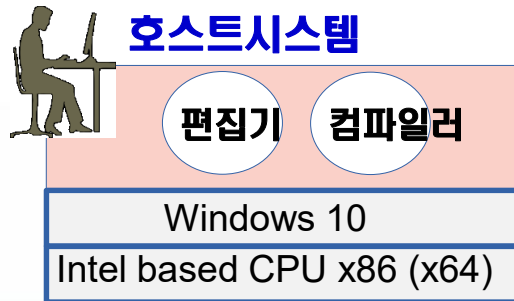
☑ 네트워크 설정

```
20160000@raspberrypi: ~  
20160000@raspberrypi:~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:10:6a:8d  
          inet addr:192.9.45.149  Bcast:192.9.47.255  Mask:255.255.252.0  
          inet6 addr: fe80::e115:c6bc:5ec3:648a/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1269 errors:0 dropped:23 overruns:0 frame:0  
          TX packets:119 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:86184 (84.1 KiB)  TX bytes:20345 (19.8 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:200 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:200 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:16656 (16.2 KiB)  TX bytes:16656 (16.2 KiB)  
  
wlan0     Link encap:Ethernet  HWaddr b8:27:eb:45:3f:d8  
          inet6 addr: fe80::84b9:66f1:11a9:cfb/64 Scope:Link  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:4 errors:0 dropped:4 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:768 (768.0 B)  TX bytes:0 (0.0 B)  
  
20160000@raspberrypi:~ $
```

개발도구의 활용

■ 임베디드SW의 개발방법

☑ 개발시스템 ≠ 타겟시스템



▲ 개발환경

교차컴파일링
Cross Compiling

타겟시스템



◀ 운영환경

타겟보드(ARM based CPU)

☑ 개발시스템 = 타겟시스템

타겟시스템

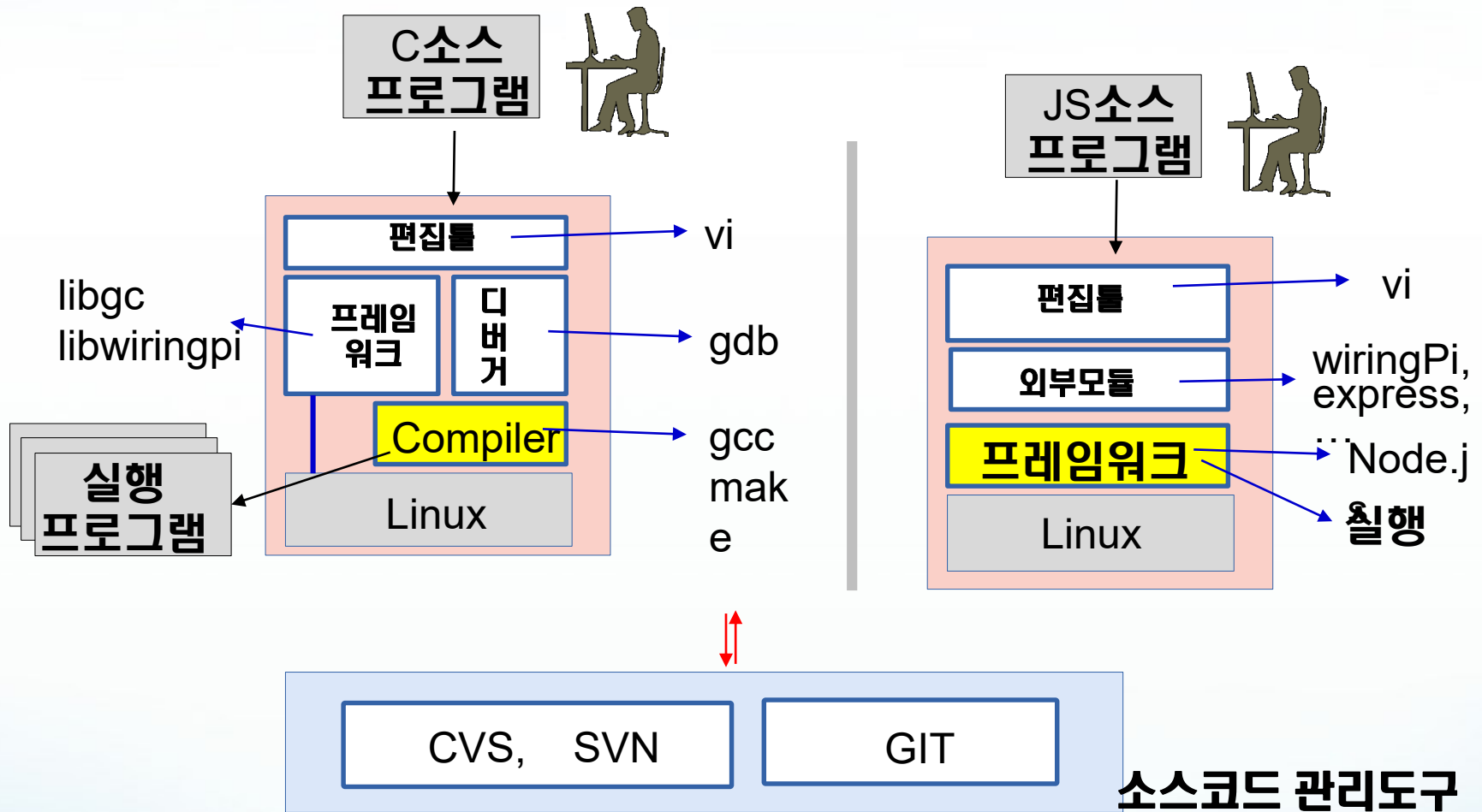


편집기
컴파일러

▲ 개발환경, 운영환경

개발도구의 활용

■ 임베디드SW의 개발도구



- 리눅스 명령 기초

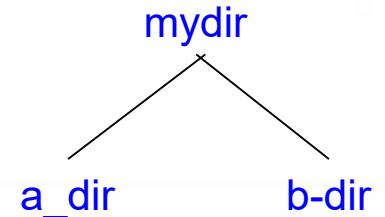
임베디드 시스템 개발환경

■ 실습(개발)환경 명령어

☑ 디렉터리 관리 명령어

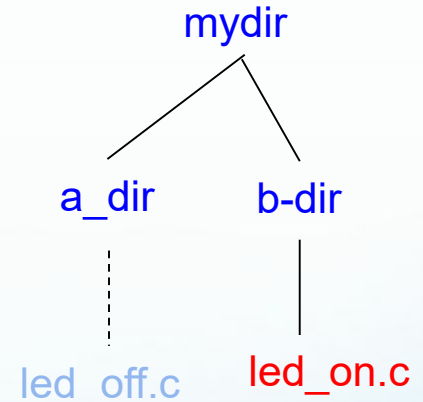
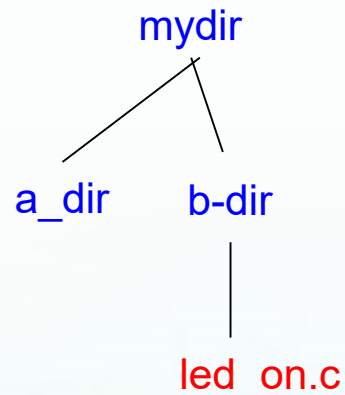
```
$ mkdir myproject  
$ cd myproject  
$ pwd  
$ cd ..  
$ rmdir myproject
```

```
$ mkdir mydir  
$ cd mydir  
$ mkdir a_dir b-dir
```



☑ 파일관리명령어


```
$ vi led_on.c  
$ cat led_on.c  
$ cp led_on.c led_off.c  
$ mv led_off.c ../a_dir  
$ rm ../a_dir/led_off.c
```



임베디드 시스템 개발환경

■ 실습(개발)환경 명령어

☑ 시스템정보 확인/관리 명령어

```
$ uname -a   
$ cat /proc/cpuinfo  
$ cat /proc/meminfo  
$ free  
$ df -h  
$ date
```



```
$ ifconfig  
$ iwconfig  
$ who -a  
$ route  
$ netstat -t  
$ sudo reboot  
$ sudo shutdown -h now
```

☑ 프로세스 관리 명령어

```
$ ps -ef  
$ top  
$ ps -ef | grep sshd
```

```
$ sudo systemctl stop ssh.service  
$ sudo systemctl start ssh.service
```

- 자바스크립트 기초
- C언어 기초

프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 자료형

- ☑ 숫자 ^{number} 27, 4, 123, 61.98, 0.5,
 - ☑ 문자열 ^{string} 'Hello', "Hello World", 'Oh my god', "Good!", 'This is a "father"'
 - ☑ 불리언 ^{boolean} true, false, !false (19 > 7, 45 > 651)
-
- ☑ 함수 코드의 집합 (선언적 함수, 익명함수)
 - ☑ 배열 여러 종류의 자료형을 내부에 넣는 자료형, 인덱스로 구분
 - ☑ 객체 여러 종류의 자료형을 내부에 넣는 자료형, 키워드로 구분

■ 변수 (전역변수: **var**, 지역변수: **let**, 상수: **const**) ☞ ECMAScript 6 에서 let, const 도입

- ☑ 변수선언 ... **var** pi; **const** mypi=3.14; **let** str= 'Hello'; **var** flag=true;
- ☑ 변수사용 ... pi = 3.14159; hours = 5; str = "test A"; flag = false;

자료형/변수

연산

제어흐름

함수

객체

■ 연산자

☑ 산술연산 + - * / % ++ --

☑ 문자열연산 + 'Oh' + 'my god'

☑ 비교연산 >= <= > < == !=

☑ 논리연산 ! || &&

■ 예

☑ `sum = sum + 5;`

☑ `code = 'initial' + 'bmlee';`

☑ `32 >= 45`

프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 조건문 (if, switch)

```
if (result > 10) {  
    console.log('10초과');  
}
```

```
if (result > 10) {  
    console.log('10초과');  
}  
else {  
    console.log('10이하');  
}
```

```
if (result > 10) {  
    if (flag) {  
        console.log('10초과');  
    }  
    else {  
        console.log("good");  
    }  
}  
else {  
    console.log('10이하');  
}
```

```
if (3 <= month && month <= 5) {  
    console.log('봄');  
}  
else if (6 <= month && month <= 8) {  
    console.log('여름');  
}  
else if (9 <= month && month <= 11) {  
    console.log('가을');  
}  
else {  
    console.log('겨울');  
}
```

```
switch (result) {  
    case 1: result++;  
            break;  
    case 2: result--;  
            break;  
    default:  
        console.log('other');  
        break;  
}
```

자료형/변수

연산

제어흐름

함수

객체

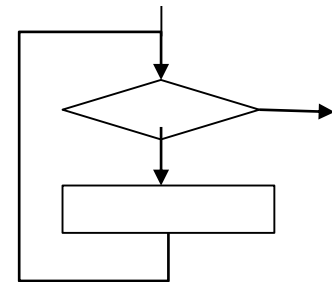
■ 반복문 (while)

```
var flag = true;
var sum = 10;

while (flag) {
    sum = sum + 10;
    flag = sum < 20;
}
```

```
var index = 0;
var array = ['가', '나', '다'];

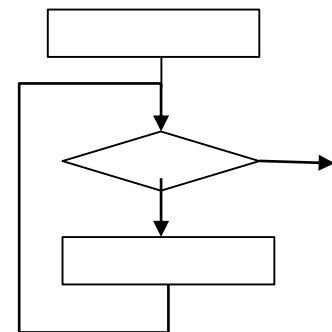
while (index < array.length) {
    console.log(array[index++]);
}
```



■ 반복문 (for)

```
var array = ['가', '나', '다'];

for (var index=0; index < 3; index++) {
    console.log(array[index]);
}
```



프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 함수선언

☑ 선언적함수 vs 익명함수

```
function 함수이름 ( ) {  
    ;  
};
```

```
function ( ) {  
    ;  
};
```

```
const Action_code = function ( ) {  
    ;  
};
```

익명함수를 호출하려면
변수를 이용

■ 함수호출

```
함수이름 ( );
```

```
Action_code( );
```

```
const print_str = function ( ) { console.log("A") };  
function print_str ( ) { console.log("B") };  
print_str( );
```

결과는 ?

프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 매개변수 & 반환값

```
function 함수이름 ( 매개변수1, 매개변수2, ... ) {  
    return 반환값  
};
```

```
var addition = function (k, p) {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

■ Callback 함수 매개변수로 전달되는 함수

```
function Calltentimes ( callback ) {  
    for (var c=0; c < 10; c++) {  
        callback();          // 함수호출  
    }  
};  
  
Var callback = function ( ) {  
    console.log("함수호출됨");  
}  
  
Calltentimes(callback); // 함수호출
```

Callback 함수사용

```
function Calltentimes ( callback ) {  
    for (var c=0; c < 10; c++) {  
        callback();          // 함수호출  
    }  
};  
  
Calltentimes ( function () {  
    console.log("익명으로 함수호출");  
} );
```

익명 Callback 함수사용

프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 다른 함수표현 법

```
function 함수이름 ( 매개변수1, 매개변수2, ... ) {  
    return 반환값  
};
```

```
var addition = function (k, p) {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

```
( 매개변수1, 매개변수2, ... ) => {  
  
};
```

```
addition = (k, p) => {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

프로그래밍기초 - 자바스크립트

자료형/변수

연산

제어흐름

함수

객체

■ 객체선언

```
var product = {  
  name: '비타민C',  
  type: '건강식품',  
  price: 10000,  
  stock: true  
};
```

```
var avatar = {  
  model: 100,  
  host: 'bmlee',  
  energy: 100,  
  mylocation: [277, 200, 21],  
  eat: function ( food ) {  
    if (food > 100)  
      (this.energy)++;  
  },  
  run: function ( ) {  
    (this.energy)--;  
  }  
}
```

■ 객체사용

또는

```
console.log( product['name'] );  
console.log( product['type'] );  
console.log( product['price'] );  
console.log( product['stock'] );
```

```
console.log( product.name );  
console.log( product.type );  
console.log( product.price );  
console.log( product.stock );
```

```
console.log(avatar.model);  
console.log(avatar.host);  
avatar.eat(300);  
avatar.run();
```

강의 **Q&A**