

Getting Started with Raspberry Pi

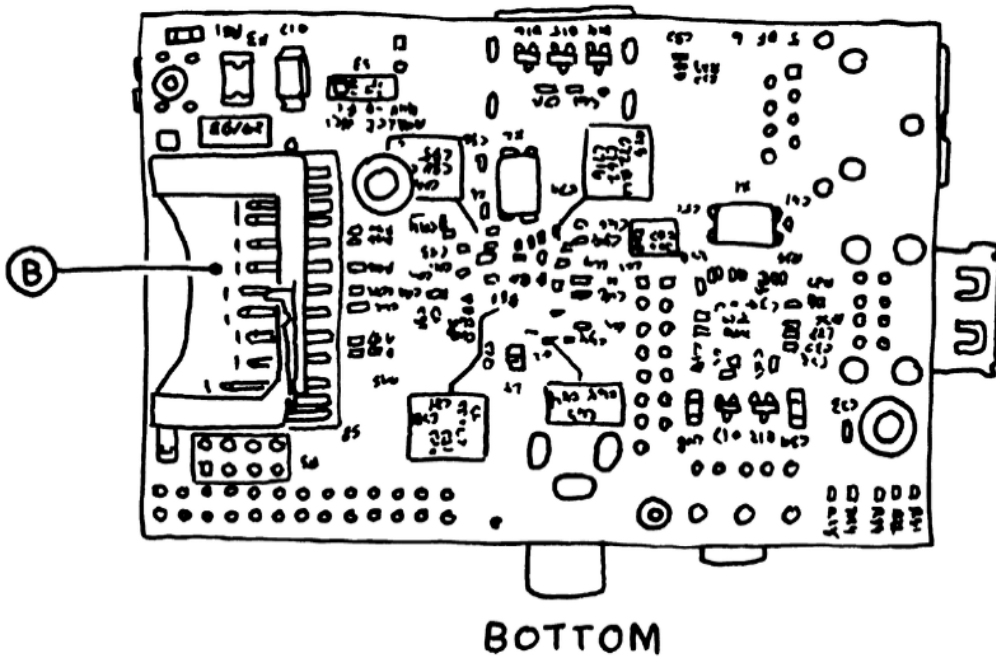
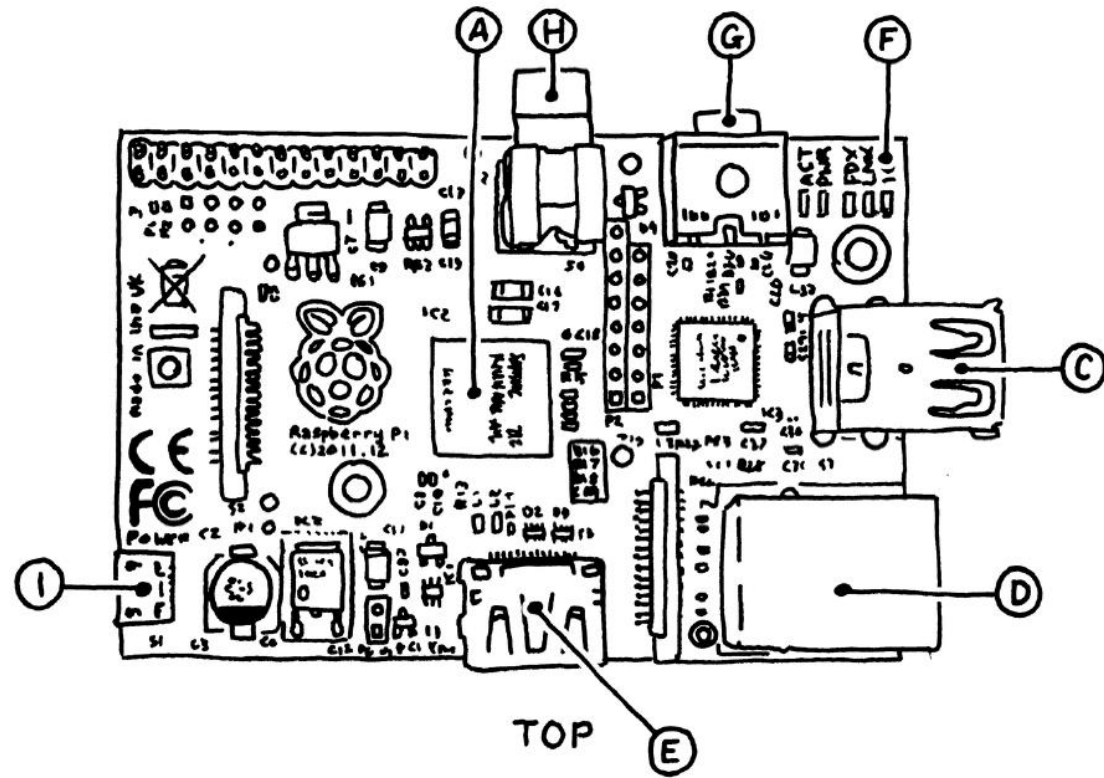
**Matt Richardson and Shawn
Wallace**

를 토대로 정리한 사항입니다.

2015.11
sghong@gmail.com

Getting Up and Running

A tour of the Boards



Getting Up and Running

A tour of the Boards

A. Processor. 라즈베리 파이의 프로세서는 아이폰3G 와 킨들2 에 있는 것과 같다. 그래서 라즈베리 파이를 그런 강력한 작은 장치처럼 생각할 수도 있다. 라즈베리 파이 프로세서는 ARM11, 32bit, 700Mhz SOC 이다. ARM은 다양한 가격에 맞게 구성될 수 있는 칩이다. 라즈베리 파이 Model B는 512MB 의 램을 갖고 있다. Model A는 256MB의 램이 포함되어 있다.(처음 Model B 은 256MB 램이다.)

B. SD card slot. 모든 정보가 SD card에 저장된다. 나중에 라즈베리 파이 이미지를 이 SD card 에 넣어 부팅 할 때 사용한다.

C. USB port. 라즈베리파이 Model B 에는 2개의 USB 2.0 포트가 있다. Model A에는 1개. 몇몇 초기 라즈베리 파이에서 USB 출력 전류가 제한된다. 초기에는 100mA 였으나, USB2.0 을 전부 지원하는 새모델에서는 500mA까지 사용할 수 있다. 이를 확인할 수 있는 두가지 방법이 있는데 Figure1-2 를 보자. 다음 장에 실제 scope를 통해서 확인 한 모습을 참고 하면 좋을 듯 하다.

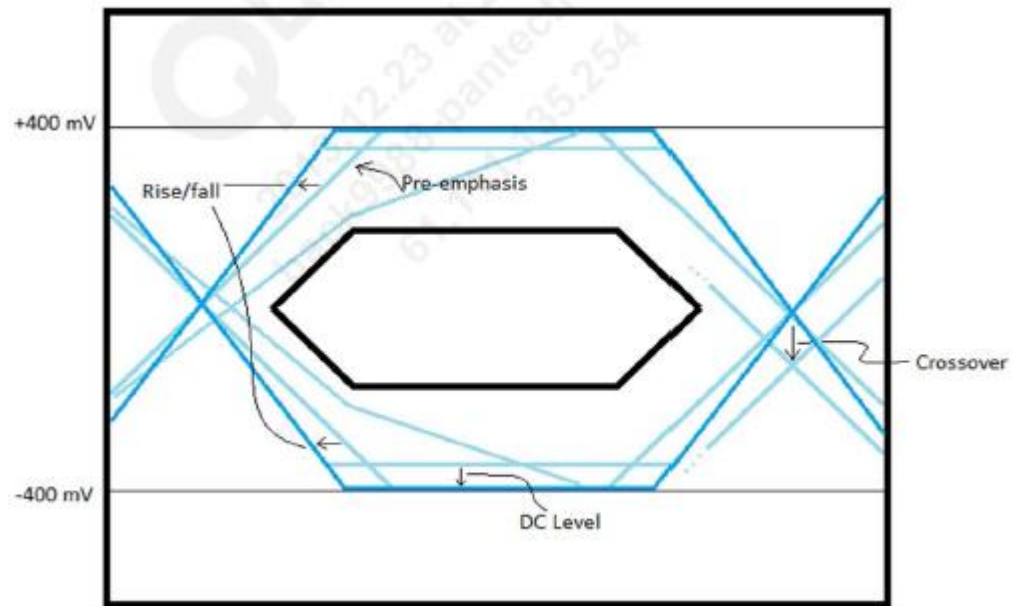
어떤 경우에서든지, 라즈베리 파이에 스마트폰을 충전하는 것은 적절치 못하다. 전원을 더 필요로 하는 주변장치를 사용한다면, 전원이 인가되는 외부 허브를 사용하면 된다.

가. 몇몇 이전 보드에서는 과전류 방지 목적으로 폴리퓨즈가 장착되어 있다.

나. 가에 장착되어 있는 폴리퓨즈가 점퍼선으로 교체된다. 지지대 설치할 수 있는 홀로 변경 됨



Figure 1-2. Some of the older boards came equipped with polyfuses (left) to protect the USB hub. Some boards have the polyfuses replaced with jumpers (center), and the latest revision of the Model B removed them and uses the space for a mounting hole (right).



usb 전원을 scope를 통해 확인한 예시

Getting Up and Running

A tour of the Boards

D. 이더넷 포트. 모델 B 는 기본 RJ45 이더넷 포트가 사용된다. 모델 A에는 이것이 없는데, USB 이더넷 어댑터를 사용해서 유선 네트워크에 연결 될 수 있다. (사실, 모델 B에는 USB 이더넷 어댑터가 장착되어 있는 것이다.) WIFI USB 동글을 통해서도 네트워크에 연결 될 수 있다. 이는 옵션이다. WIFI 동글을 추가로 구입해야 한다.

E. HDMI 커넥터. 비디오 입출력을 제공한다. 14개의 다른 해상도를 지원. DVI 컨버터를 장착해서 변환될 수 있다. RCA 커넥터 노란색으로 오디오 전송.

F. LED. **Table 1-1.** *The five status LEDs.*

ACT	Green	Lights when the SD card is accessed (marked OK on earlier boards)
PWR	Red	Hooked up to 3.3V power
FDX	Green	On if network adapter is full duplex
LNK	Green	Network activity light
100	Yellow	On if the network connection is 100Mbps (some early boards have a 10M misprint)

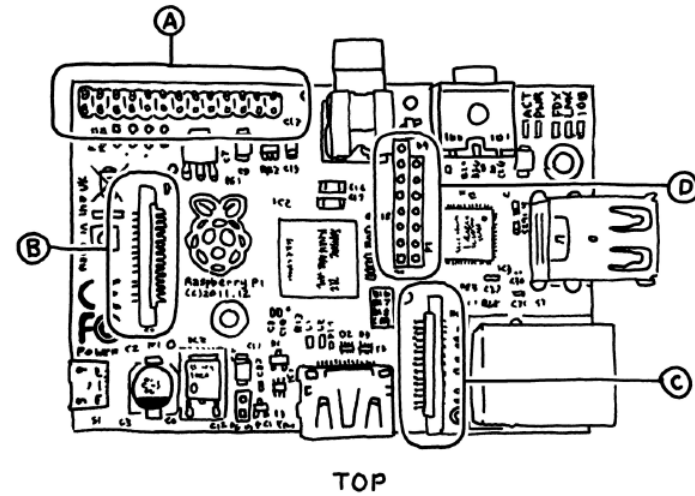
G. 오디오 잭. 3.5mm mini 아날로그 오디오 잭, amp 달린 스피커도 지원.헤드폰이나 전원없는 스피커는 음질이 안좋다. 사실, HDMI로 TV에 전송된 음질보다는 좋지 않다. 오디오 소프트웨어 드라이버를 통해서 개발 진행 중이다.

H. 비디오 출력. NTSC, PAL 비디오 시그널. HDMI 보다는 해상도가 낮다. 이 비디오 포트 보다는 HDMI 텔레비전이나 모니터를 사용하는 것이 좋다.

I. 전원. 라즈베리 파이에는 전원 버튼이 없다. 마이크로 USB를 통해서 전원이 공급된다.

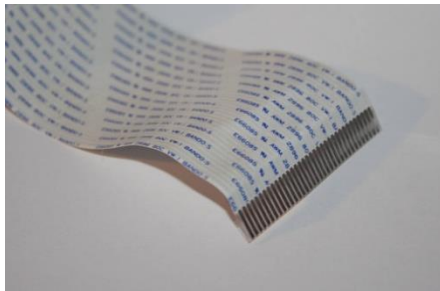
Getting Up and Running

A tour of the Boards

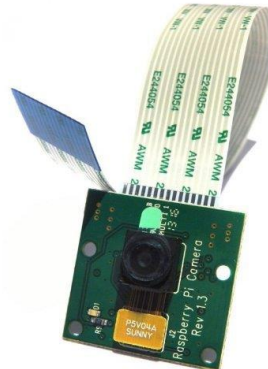


A. General Purpose Input and Output(GPIO) 7,8장에서 어떻게 이 pin 들을 사용하고, 이 pin 을 이용하여 LED를 컨트롤 하는 방법을 알 수 있다.

B. The Display Serial Interface(DSI) connector. 15 pin flat ribbon cable 아래와 같은 형태이다. LCD 및 OLED 의 스크린을 장착 할 수 있다.



C. The Camera Serial Interface(CSI) connector. 아래와 같은 카메라를 장착할 수 있다.



Getting Up and Running

A tour of the Boards

D. Broadcom chip 자체(P2)와 LAN9512(P3)를 디버깅하기 위한 jtag 포트이다. 사용할 일이 많지는 않다.

2015년 4월 라즈베리 파이 2 보드가 출시되었다. 이후 B2 모델까지 출시되었다.

http://ko.wikipedia.org/wiki/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC_%ED%8C%8C%EC%9D%B4 를 참조하면 각 모델별 specification 을 확인할 수 있다.

Getting Up and Running

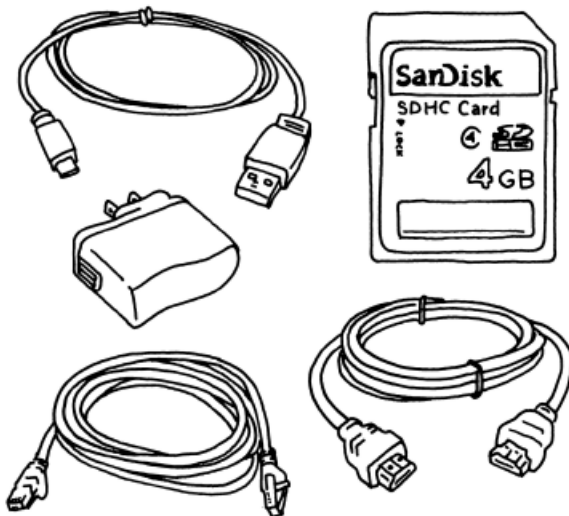
The Proper Peripherals

이제 Pi 보드에 있는 모든 것을 알아 보았다. 앞으로 몇가지 유용한 PI에 필요한 주변장치를 알아보자

http://elinux.org/RPi_VerifiedPeripherals

위 사이트를 참고하면 여러 주변 장치를 연결하는 방법을 알 수 있다.

- A. 전원. 마이크로 USB 를 통해서 전원을 공급받는데, 꼭 필요한 아이템이다. 5v 700mA 정도면 된다. 일부 전화기 충전용 중에는 400mA가 있는데 이걸 사용하면 안된다. 그 이상되는 충전기는 사용해도 된다. 요즘 5V 2mA는 사용해도 된다.
- B. SD card. Pi 이미지가 write 되는 곳이다. 적어도 4G는 되어야 함. 전송속도는 초당 4MB. Pi 의 sdcard 드라이버는 class6 또는 그 보다 높은 사양의 card에서는 높은 속도에 대해서는 문제점이 있지만 그 이하 속도에서는 안정적이다. Micro SD 카드 어댑터에서는 잘 동작 한다.



Getting Up and Running

The Proper Peripherals

C. HDMI 케이블. 모니터 연결을 위한 꼭 필요 함.

D. 이더넷 케이블. 모니터 없이 작업 할 경우. VNC 를 이용해서 화면을 원격으로 브로드캐스팅 할 수 있는데, 이 때 필요하다. 물론 인터넷 사용을 위해서도 필요하다.

전원공급되는 USB

- RPi 의 USB 1개 포트를 통해서 키보드,마우스를 연결 하여 포트를 절약할 수 있다.

발열기 Heatsink

- Pi, 에 장착된 broadcom 칩셋이 대부분 발열기가 필요가 없지만, 더 높은 속도에서 약간 발열이 있을 수 있다. 몇몇 사람들이 network 칩에서 발열이 생길 수 있다고 한다.



Getting Up and Running

The Proper Peripherals

Real Time Clock

Camera module

- 공식적으로 5메가픽셀 카메라 모듈을 사용할 수 있는데 따로 설명하겠다. Chapter 9 에서 좀더 자세히 설명 함.

LCD display

- GPIO 를 통해서 사용할 수 있다.

Wifi USB Dongle

- 많은 종류의 와이파이 동글을 사용할 수 있다. 전원을 많이 소비하지 않는 것을 찾아보자

Case

라즈베리 파이용 케이스는 쉽게 찾아 볼 수 있다. 너덜너덜한 케이블들을 정리 할 수 있다. 정상적인 사용에도 기계적 결함을 일으킬 수 있는 SD 카드 슬롯같은 장치에 적용할 수 있다.

Pi 에는 전기가 흐를 수 있는 6가지 장치가 있다. 반면, 위,아래에 있는 간단한 마이크로 컨트롤러 PCB는 제외된다. 위,아래 사이에 4개의 층이 있는데 부서질 정도로 힘을 줘서 구부려트리면 작동을 하지 않을 수 있는데, 이럴 경우 디버깅이 힘들다. 이런 사항을 막기위해서도 case 가 필요하다.

Getting Up and Running

<https://www.raspberrypi.org/downloads/>

라즈베리 파이를 구동하기 위한 여러가지 버전의 이미지들이 있다.

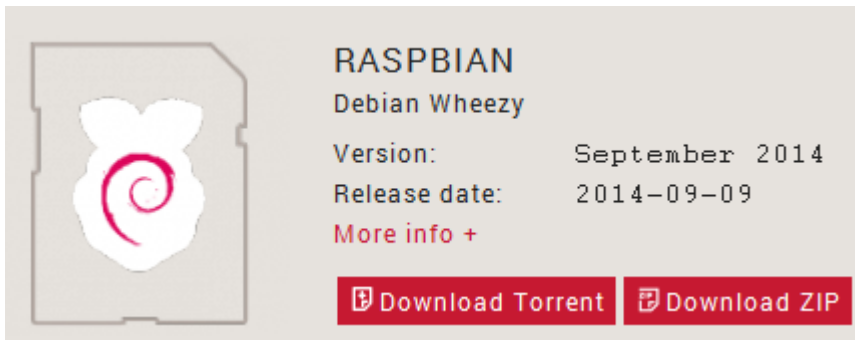
Raspbian

- debian 계열의 리눅스

Xbian / QtoPi / Arch Linux / Adafruit 등등이 있다. 이렇듯 여러 개의 배포판이 있는데 그 중에 다음에 소개하는 raspbian 이미지를 사용하도록 하자.

Flash the SD Card

Raspbian 이미지를 다운로드 한다. NOOBS 를 다운로드 해도 된다. 이후에 upgrade 를 하면 RASPBIAN 이 될 수 있다.



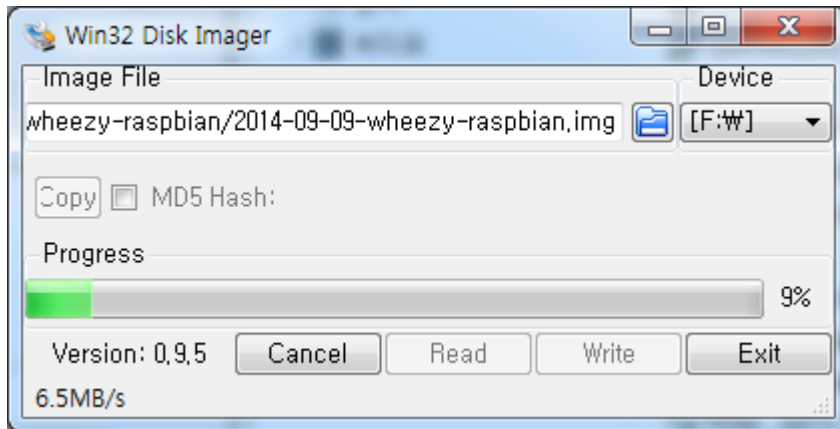
압축을 풀면 2014-09-09-wheezy-raspbian 이미지파일이 생긴다.

Getting Up and Running

Flash the SD Card

라즈베리 파이 이미지가 미리 인스톨된 sdcard가 아니라면, sdcard에 이미지를 작성하는 과정이 필요하다.

<https://sourceforge.net/projects/win32diskimager/files/latest/download>



Write가 성공한 후, sdcard를 라즈베리 파이에 장착 한 후, bootup 한다.

Getting Up and Running

Booting Up

이미지 작성이 끝난 이후 다음 단계를 진행 하자

1. SD 카드를 소켓에 장착한다.
2. USB 키보드 마우스를 연결한다. Model A 의 경우는 USB 소켓이 1개라 전원이 있는 USB 허브를 사용하여 키보드 마우스를 연결한다.
3. HDMI 를 TV 나 모니터에 연결한다.
4. 전원 연결. 전원을 ON 하기 전에 다른 것들은 연결하지 않는다.

모든게 잘 되면, 스크린에 부팅시 쏟아지는 수 많은 글자들을 볼 수 있다. 커널이 부팅할때 콘솔로 뿌려지는 메시지이다. 이는 부팅 후, `cat /proc/kmsg` 를 통해서 확인 할 수 있다.

`raspi-config` 명령을 통해서 언제든지 몇 가지 설정을 할 수 있다.

`#sudo raspi-config`

```
Raspi-config

info      Information about this tool
expand_rootfs  Expand root partition to fill SD card
overscan    Change overscan
configure_keyboard  Set keyboard layout
change_pass   Change password for 'pi' user
change_locale Set locale
change_timezone Set timezone
memory_split  Change memory split
overclock    Configure overclocking
ssh          Enable or disable ssh server
boot_behaviour Start desktop on boot?
update       Try to upgrade raspi-config

<Select>          <Finish>
```

Getting Up and Running

Configuring Your Pi

각 option 을 상/하 키 및 Enter 키 등을 이용하여 둘러볼 수 있다.

Expand rootfs

- ??? 이 부분은 원지 확인해보자

Overscan

- 화면이 찢리거나 할 경우 설정해 주는데 보통 disable 한다.

Keyboard

- 키보드 레이아웃을 설정한다.

Password

- 암호변경 (기본은 pi / raspberry)

Change Locale

- 각 나라에 맞는 locale 설정을 할 수 있다. 기본 UTF-8 로 설정한다.

Change timezone

- 각 나라에 맞는 timezone 을 설정한다.

Memory split

- CPU GPU에 의해 사용되는 메모리를 조정할 수 있는데, 기본설정으로 하자

Overclock

- 700MHz 로 동작되는 Pi 를 변경할 수 있다. Turbo mode 는 1000MHz에서 동작.

SSH

- ssh 서버를 구동하는 것 나중에 ssh client 를 통해서 Pi 에 접속할 경우에는 enable 한다.

Desktop Behavior

- 부팅시 그래픽 모드 또는 텍스트 모드로 할지 선택.

Update

- 마지막으로 인터넷에 연결되어 있으면 수행 할 수 있는데 (처음에는 수행 하지 않는다.)이후에 설명한다.

설정을 변경한 이후에, 다음 명령으로 재부팅한다. **#sudo reboot**

Pi 에는 전원 버튼이 따로 없다. 다음 명령으로 전원을 끌 수 있다. **#sudo shutdown -h now**

Getting Up and Running

Configuring Your Pi



사용하고 있는 Pi 의 OS 버전을 알고 싶으면 command line 에서 다음과 같이 명령을 해준다.

```
#cat /proc/version
```

보드 버전을 알고 싶으면,

```
#cat /proc/cpuinfo
```

Pi 의 하드웨어와 설정에 대한 정보

- http://elinux.org/RPi_Hub

Pi 에서 동작하는 주변 장치들에 대한 정보

- http://elinux.org/RPi_VerifiedPeripherals

2/Getting Around Linux on the Raspberry Pi

이번 chapter 를 통해서 라즈베리에서 동작하고 있는 리눅스에 대해서 알아보는 것을 목표로 한다.

작은 사이즈의 X11 desktop Environment (LXDE) 가 구동되고 있다. 이 LXDE는 80 년대 이후로 리눅스와 유닉스에서 강력한 GUI 인 X window system 을 경량화 시킨 것이다.

이 LXDE 에는 OpenBox 라는 윈도우의 크기를 조절하거나, look&feel 을 담당하는 window manager가 구동되고 있다.

모양이 마음에 들지 않으면 Openbox configuration tools (click the desktop menu in the lower left, then choose Other→Openbox Configuration Manager). 를 통해서 수정할 수 있다.

이 Openbox Window manager 대신 다른 윈도우 매니저를 설치할 수 있고, RPi 의 다른 배포버전에서는 다른 환경을 제공하는데, http://elinux.org/RPi_Distributions 여기에서 확인해 볼 수 있다.

기본적으로 리눅스 사용법을 알기 위한 몇 가지 사항을 언급하겠다.

File Manager

- 윈도우의 탐색기와 같은 역할을 한다. Accessories 메뉴에서 선택한다.

Web Browser

- Midori 브라우저가 구동되고 있다. 라즈베리파이는 작고 가벼운 OS로 디자인되어 있어서 윈도우에서 구동되는 크롬이나 익스플로러같은 화려한 기능을 갖고 있지는 못하다. 예를 들어 Flash 나 Java Plugin 이 동작되지 않는다. HTML5 지원되지 않는다.

Video Audio

Omxplayer 에 의해 멀티미디어가 처리되고 있다. 마찬가지로 많은 동영상(인코딩방식에 따른)을 지원하고 있지는 못하다.

2/Getting Around Linux on the Raspberry Pi

TextEditor

- Leafpad 가 장착되어 있다. Vim이나 Emacs 는 설치되어 있지 않으나 설치할 수 있다. Installing New Software 참조

The Shell

- command line 으로 명령어를 전달하여 실행할 수 있는 방법. LXTerminal 을 통해서 shell 과 commandline 에 접근할 수 있다.

Using the Command Line

아래와 같이 LXTerminal 프로그램을 실행시켜면 command를 입력할 수 있다.

예로, ls 를 타이핑하고 엔터를 누르면 도스창에서 dir 과 같이 파일을 볼 수 있다.

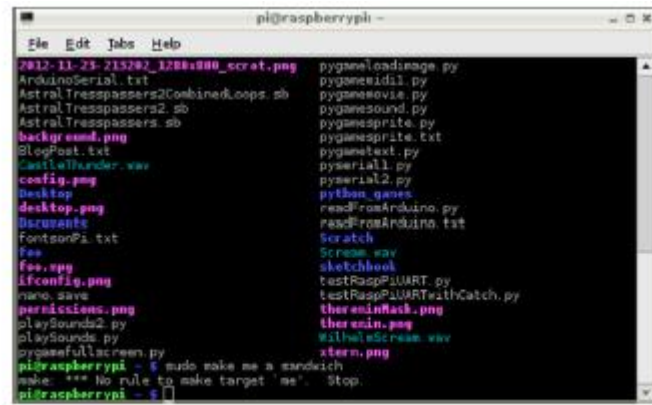


Figure 2-3. LXTerminal gives you access to the command line (or shell).

2/Getting Around Linux on the Raspberry Pi

Files and the Filesystem

아래와 같이 파일시스템에서 어떤 역할을 하는지 알 수 있다.

Directory	Description
/	
/bin	Programs and commands that all users can run
/boot	All the files needed at boot time
/dev	Special files that represent the devices on your system
/etc	Configuration files
/etc/init.d	Scripts to start up services
/etc/X11	X11 configuration files
/home	User home directories
/home/pi	Home directory for pi user
/lib	Kernel modules/drivers
/media	Mount points for removable media
/proc	A virtual directory with information about running processes and the OS
/sbin	Programs for system maintenance
/sys	A special directory on the Raspberry Pi that represents the hardware devices
/tmp	Space for programs to create temporary files
/usr	Programs and data usable by all users
/usr/bin	Most of the programs in the operating system reside here
/usr/games	Yes, games
/usr/lib	Libraries to support common programs
/usr/local	Software that may be specific to this machine goes here
/usr/sbin	More system administration programs
/usr/share	Things that are shared between applications like icons or fonts
/usr/src	Linux is open source; here's the source!
/var	System logs and spool files
/var/backups	Backup copies of all the most vital system files
/var/cache	Any program that caches data (like apt-get or a web browser) stores it here.
/var/log	All of the system logs and individual service logs
/var/mail	All user email is stored here, if you're set up to handle email
/var/spool	Data waiting to be processed (e.g. incoming email, print jobs)

2/Getting Around Linux on the Raspberry Pi

현재 디렉토리를 볼 수있고, 리눅스 홈디렉토리는 tilde(~)로 바로 옮겨 갈 수 있다.

```
$cd ~
```

A terminal prompt on a Raspberry Pi. The text is 'pi@raspberrypi ~ \$'. Each part is annotated with a number in a black circle: 'pi' is 1, '@' is 2, 'raspberrypi' is 3, '~' is 4, and '\$' is 5. The text has a rainbow-colored shadow effect.

pi¹@²raspberrypi³ ~⁴ \$⁵

Prompt 를 설명하면,

1 - 현재 사용자 이름이 Pi

2 - 컴퓨터 이름, raspberrypi

3 - 현재 작업하고 있는 디렉토리를 보여준다.

4 - 타이핑하는 글자가 보여지고 명령어를 입력할 수 있다.

```
$cd /home/pi
```

```
$cd ~
```

위 명령어는 홈디렉토리로 이동하는 같은 명령이다. / 로 시작되는 path 는 절대 경로이다.

```
$cd ..
```

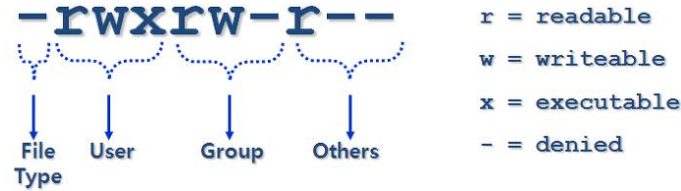
이전 단계의 디렉토리로 옮겨가는 한칸 띄어서 .. 을 입력해야한다. .. 또한 디렉토리이기 때문이다.

```
$ls
```

\$ls -l 파일 속성등 좀 더 많은 정보를 보여준다 파일 크기, 날짜, 파일 권한을 보여준다.

\$ls -la 보여지지 않는 파일도 보여준다.

2/Getting Around Linux on the Raspberry Pi



처음 보여지는 리눅스 파일의 권한은 위와 같이 4가지로 구분해 볼 수 있는데,
 제일 처음은 파일 타입-d(디렉토리), l(링크파일)-을 표시
user는 현재 사용자, **g**roup 은 현재 사용자가 속해 있는 같은 그룹, **o**thers 는 나머지 사용자를 의미한다.

파일 권한을 바꿔주는 `chmod(changed mode)` 명령이 있다.

`$chmod g+w foo` , foo 파일에 그룹에 w 권한 추가
`$chmod u+rw,g-w,o-x foo` user에 rw 권한을 추가,group 에는 쓰기 권한 제거, others 에는 실행권한 제거

rwX 는 각각 read, write, execute 권한을 의미하고, 각각 4, 2, 1 숫자로 설정할 수 있다. 각 그룹에 권한을 숫자를 합한 값으로 한자리로 표현할 수 있다.

$rw- = 4 + 2 + 0 = 6$

$r-x = 4 + 0 + 1 = 5$

$rwX = 4 + 2 + 1 = 7$

`$chmod 755 foo`, foo 파일에 `rwrx-rx-rx` 권한 설정

파일 명령중에 `touch`(빈 파일 생성), `mv`(파일을 옮긴다.move)
`rm`(파일 삭제,remove),`mkdir`(make directory, 디렉토리생성),
`rmdir`(디렉토리 삭제)

명령어에 대한 매뉴얼을 보고 싶을 경우 `man` 명령을 사용한다.
`$man ls`

```
pi@raspberrypi ~ $ ls -la
total 80
drwxr-xr-x 11 pi pi 4096 Oct 12 14:26 .
drwxr-xr-x  3 root root 4096 Sep 18 07:48 ..
-rw-----  1 pi pi  25 Sep 18 09:22 .bash_history
-rw-r--r--  1 pi pi  220 Sep 18 07:48 .bash_logout
-rw-r--r--  1 pi pi 3243 Sep 18 07:48 .bashrc
drwxr-xr-x  6 pi pi 4096 Sep 19 01:19 .cache
drwxr-xr-x  9 pi pi 4096 Oct 12 12:57 .config
drwx-----  3 pi pi 4096 Sep 18 09:24 .dbus
drwxr-xr-x  2 pi pi 4096 Oct 12 14:26 Desktop
-rw-r--r--  1 pi pi  36 Sep 18 09:35 .dmrc
drwx-----  2 pi pi 4096 Sep 18 09:24 .gvfs
drwxr-xr-x  2 pi pi 4096 Oct 12 12:53 .idlerc
-rw-----  1 pi pi  35 Sep 18 12:11 .lessht
drwx-----  3 pi pi 4096 Sep 19 01:19 .local
-rw-r--r--  1 pi pi  675 Sep 18 07:48 .profile
drwxrwxr-x  2 pi pi 4096 Jul 20 14:07 python_games
drwx-----  4 pi pi 4096 Oct 12 12:57 .thumbnails
-rw-----  1 pi pi  56 Sep 18 09:35 .Xauthority
-rw-----  1 pi pi  300 Oct 12 12:57 .xsession-errors
-rw-----  1 pi pi 1391 Sep 18 09:35 .xsession-errors.old
```

2/Getting Around Linux on the Raspberry Pi

More Linux Commands

리눅스와 유닉스 시스템이 성공을 거둔 이유 중 하나는 작고 서로 같이 연결될 수 있는 간단한 모듈을 가지고 매우 복잡한 시스템을 빌드하도록 디자인 된 것이다. 사실 안드로이드 스마트폰에는 복잡한 장치들이 연결되어 있는데 리눅스는 이 장치들을 효과적으로 관리할 수 있도록 디자인 되어 있다. 각각의 장치 드라이버를 insmod rmmod 형태로 컨트롤 할 수 있다. pipe 와 redirection, 이 두 종류를 조금 알 필요가 있다.

Pipes(파이프)는 두개의 프로그램을 같이 연결하는 방법이다. 하나의 output(결과물)이 다른 프로그램의 input(입력)이 될 수 있다. 모든 리눅스 프로그램은 표준입력(stdin, standard input) 으로부터 데이터를 읽고, 표준출력(standard output, stdout)에 데이터를 쓸 수 있다. 에러 메시지를 stderr(standard error) 로 보낼 수 있다. 파이프는 한 프로그램의 stdout 을 다른 프로그램의 stdin 으로 보낼수 있게 해준다.
예를들면,

```
pi@raspberrypi ~ $ ls -la | less
```

(Press q to exit the less program.)

위 명령은 파이프를 통해 ls -al 의 결과를 less 로 전달하는 것. `ls -al` 한 결과가 한 페이지에 표기할 수 없을 정도로 많은 경우 less 를 통해서 페이지 단위로 볼 수 있다.

```
pi@raspberrypi ~ $ sudo cat /boot/kernel.img | aplay
```

위 명령은 audio play 로 kernel.img 의 데이터를 전달하는 것인데, 어떤 소리가 나는지 한번 들어보길 바란다.

redirection 은 이후에 다시 사용되는데, 명령어를 실행하고 그 출력을 파일로 보낼 수 있다. 추후 보여지듯이-리눅스에서 많은 것들이 보통의 파일처럼 취급되어지는데,(GPIO 핀을 컨트롤하는 것도 파일로서 가능하다) - redirection 은 매우 쉽게 처리될 수 있다. `>` 기호를 이용해서 redirection 을 할 수 있다.

```
pi@raspberrypi ~ $ ls > directoryListing.txt
```

ls 의 결과가 redirect 되어 파일 directoryListing.txt 로 저장된다.

2/Getting Around Linux on the Raspberry Pi

More Linux Commands

Special Control Keys

추가적으로 명령창에서 tab 키를 누르면 관련된 명령어를 볼 수 있고 따로 타이핑 하지 않고 자동으로 명령어를 타이핑해 준다. 이전 명령어를 보고 싶을 경우 위 화살표키를 누르면 된다. Shell 에서 사용할 수 있는 몇가지 키 조합이 있다.

Control-C

현재 수행 중인 프로그램을 kill 한다. 간혹 프로그램이 대답이 없거나 할 경우 사용한다.

Control-D

shell 을 빠져나간다. Shell 의 프롬프트에서 어떤 것도 타이핑하지 않은 상태에서만 동작 한다.

Control-A

커서를 라인 첫 부분으로 옮긴다.

Control-E

커서를 라인 마지막 부분으로 옮긴다.

다른 것들도 있는데 이것들이 매일 사용하는 핵심 단축키이다.

위 명령은 audio play 로 kerne.img 의 데이터를 전달하는 것이데, 어떤 소리가 나는지 한번 들어보길 바란다.

redirection 은 이후에 다시 사용되는데, 명령어를 실행하고 그 출력을 파일로 보낼 수 있다. 추후 보여지듯이-리눅스에서 많은 것들이 보통의 파일처럼 취급되어지는데,(GPIO 핀을 컨트롤하는 것도 파일로서 가능하다) – redirection 은 매우 쉽게 처리될 수 있다. `>` 기호를 이용해서 redirection 을 할 수 있다.

예를 들어, 두번째는 copy 와 동일한 기능을 할 수 있다.

```
pi@raspberrypi ~ $ ls > wibble.txt
pi@raspberrypi ~ $ cat wibble.txt > wobble.txt
pi@raspberrypi ~ $ cat wibble.txt wobble.txt > wubble.txt
```

마지막 라인은 두개의 파일을 하나로 합쳐주는 역할을 한다.

특정 문자열을 찾기 위해 grep 을 이용한다. \$grep -nr "찾고자하는 문자열" *

2/Getting Around Linux on the Raspberry Pi

Processes

Pi 에서 동작하는 모든 프로그램은 각각 별개의 프로세스 단위로 구동되고 있다. 적절한 시점 어떤 특정한 순간에 12개의 프로세스가 동작 한다. Pi가 부팅후, 다른 task 또는 service로 처리되는, 75의 프로세스가 동작한다. 실행 중인 모든 프로세스를 보려면 top 명령어를 사용한다. CPU 와 메모리 사용량을 표시해준다.

```
pi@raspberrypi ~ $ ps -aux | less
```

때때로 응답이 없거나 동작이 안되는 프로세스를 kill 하길 원한다. ps 로 프로세스 id를 찾아서 kill 하면된다.



시스템 프로세스의 경우에는 kill 하기 위해서 권한을 갖고 있어야 한다.

Sudo and Permissions

리눅스는 멀티유저 시스템이다. 모든 사용자는 자신의 공간에서 자신의 파일만 생성,수정,삭제를 할 수 있는 것이 일반적인 규칙이다. root 사용자는 파일시스템에 있는 어떤 파일도 바꿀수 있다. 매일 root 로 로그인 하지 않는 연습을 하는 게 좋다.



pi 사용자로 시스템에 해를 끼치지 않는다. Super user 가 우연히 큰 혼란을 줄 수 있다. sudo 상태에서 파일을 지우거나 옮길 경우 조심해라. 이럴 경우 잘못 되면, SD 카드 이미지를 다시 만들면 된다. (Appendix A 를 참조)

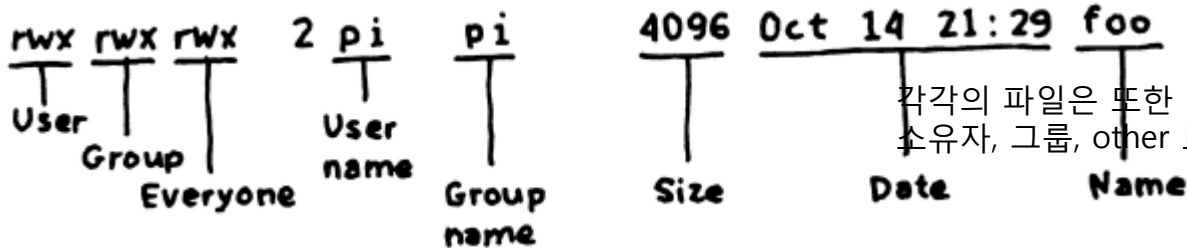
2/Getting Around Linux on the Raspberry Pi

super user 로서 명령어를 실행해 줄 수 있는 sudo 와 같은 도구가 있다. 하드웨어와 직접적으로 다룬다거나 시스템 전반적인 설정을 변경할 때 많이 사용한다.

모든 파일은 특정 사용자와 그룹에 속해있다. chgrp 과 chown 명령어를 이용하여 소유자와 그룹을 변경할 수 있다.이 명령을 사용하기 위해서는 root 권한을 얻어야 한다.

```
pi@raspberrypi ~ $ sudo chown pi garply.txt
pi@raspberrypi ~ $ sudo chgrp staff plugh.txt
```

각각의 파일은 또한 읽기,쓰기,실행 권한을 갖고 있다. 이 권한은 소유자, 그룹, other 로 구분된다.



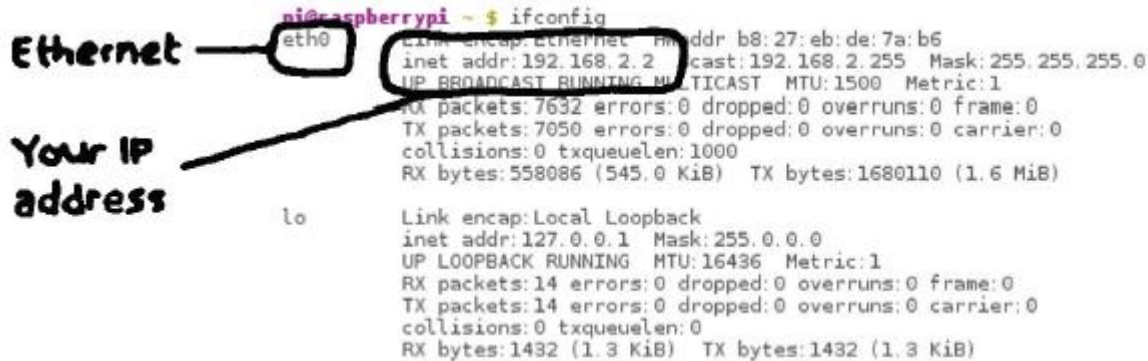
u	user
g	group
o	others not in the group
a	all/everyone
r	read permission
w	write permission
x	execute permission
+	add permission
-	remove permission

4 r
2 w
1 x

2/Getting Around Linux on the Raspberry Pi

The Network

네트워크가 있으면, 보통 기본으로 사용하고 있는 몇가지 리눅스 유틸리티가 있다. 인터넷 연결에 문제가 있으면, ifconfig-IP 주소, MAX 주소를 볼 수 있는 유틸리티



```
pi@raspberrypi ~ $ ifconfig
eth0: Link encap:Ethernet  HWaddr b8:27:eb:de:7a:b6
      inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:7632 errors:0 dropped:0 overruns:0 frame:0
      TX packets:7050 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:558086 (545.0 KiB)  TX bytes:1680110 (1.6 MiB)

lo:    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:14 errors:0 dropped:0 overruns:0 frame:0
      TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:1432 (1.3 KiB)  TX bytes:1432 (1.3 KiB)
```

네트워크가 있으면, 보통 기본으로 사용하고 있는 몇가지 리눅스 유틸리티가 있다. 인터넷 연결에 문제가 있으면, ifconfig-IP 주소, MAX 주소를 볼 수 있는 유틸리티이다.

ping 을 통해서 인터넷 연결을 확인해 볼 수 있다. 이를 통해서 양방향 연결을 확인해볼 수 있다. 그런데 어떤 웹사이트는 ping 의 응답을 하지 않을 수 있기때문에 여러 사이트를 시도해 본다.

원격 컴퓨터에 로그인하기 위해서 SSH(Secure Scoket Shell) 를 사용할 수 있다. 그러기 위해서는 원격으로 접속하는 컴퓨터에 ssh server 가 구동되고 있어야 하는데 라즈베리 파이에는 ssh 가 빌드되어 있다. 사실 Running Headless

SSH와 연관되어 원격으로 파일을 전송할 수 있는 sftp 프로그램이 있다. 인터넷 또는 네트워크를 통해서 다른 컴퓨터로 파일을 전송 할 때 사용하는 scp 가 있다. 이런 툴들은 SSL(Secure Socket Layer) 를 통해서 암호화된 로그인 정보를 갖고 파일을 전송한다. 이런 툴들은 많이 사용되는 기본 리눅스 툴이다.

2/Getting Around Linux on the Raspberry Pi

Running Headless

사실 headless 라는 표현이 낯선데, 여기서는 모니터, 키보드, 마우스 없이 네트워크를 통해서 라즈베리 파이에 연결함을 의미한다. 전통적으로 EXE 같은 executable(x86,arm) 파일이 machine 에서 구동되는 보통의 어플리케이션이 아닌 경우에도 headless 라는 표현을 사용한다. 한편, WebOS 에서 native application 을 실행시키는 방식을 headless 라고도 부른다.

어쨌든 여기서 의미하는 headless 를 하기 위해서는 설정을 해줘야 한다. 커맨드라인에서 사용자는 pi 암호는 raspberry 로 ssh 연결을 한다. ssh server 는 라즈베리 파이에 기본적으로 설치가 되어 있다. 그렇지 않을 경우, raspi-cofnig 를 통해서 ssh server 를 enable 한다.

네트워크를 통해서 Pi 에 접속하는 다른 방법으로는 VNC server(Virtual Network Computing Server)를 이용하는 방법이다. 이를 통해서 그래픽 환경의 윈도우를 볼 수 있다. X 를 이용하여 지정된 원격으로부터 화면을 볼 수 있다. 이 부분은 추후 자세히 설정하고 사용하는 방법을 알아본다.

/etc

/etc 디렉토리에는 시스템 전반적인 설정 파일과 실행 스크립트들이 있다. Pi가 부팅할 때 처음 시작되는 스크립트들의 값을 여기서 조정 할 수 있다. /etc 이하의 파일을 수정할 때 슈퍼 유저 권한일 필요 할 것이다. 파일 수정할 경우 vi 를 사용하면 된다.

Setting the Date and Time

인터넷에 연결되면, NTP(Network Time Protocol)를 이용하여 자동적으로 시간 및 날짜를 설정할 수 있다. 관련 명령어인 date 를 이용하여 아래와 같이 설정한다.

```
$ sudo date --set="Sun Nov 18 1:55:16 EDT 2012"
```

2/Getting Around Linux on the Raspberry Pi Installing New Software

Pi에서는 패키지라는 개념으로 소프트웨어를 설치 관리한다. 패키지 매니저는 프로그램을 다운로드하고 의존성에 따라 설치하는 것을 관리한다. 리눅스에서 이렇게 의존성에 따라 설치하는 패키지 매니저는 굉장히 잘 구축되어 있다.

Pi는 매우 최소화 하여 소프트웨어가 설치되어 있는데, 새로운 프로그램을 다운로드하고 설치할 수 있다. 여기서는 명령창에서 유동적이고 빠르게 소프트웨어 설치를 할 수 있다.

apt-get의 -install 옵션을 가지고 소프트웨어를 다운로드 한다. apt-get은 의존성을 고려하지 않고도 소프트웨어를 다운로드 할 수 있게끔 해준다.

설치를 위해서는 슈퍼유저 권한을 갖고 해야하는데 sudo 명령을 이용한다. emacs라는 텍스트 편집기를 다운로드 하고 설치해 보자.

```
pi@raspberrypi ~ $ sudo apt-get install emacs
```



Takging a Screenshot

이 책을 쓸때 이해해야 할 첫 번째것들 중 하나는 어떻게 Pi에서 스크린샷을 캡처할 수 있는가 이다. scrot (SCReenshOT)이라는 프로그램이 있는데, 스크린 캡처하는 다른 방법은 GIMP 또는 ImageMagick을 설치하는 것이다.

```
sudo apt-get install scrot
```

3/Python On The Pi

파이선은 훌륭한 프로그래밍 언어이다. 명확하고 설치하고 실행하기 쉽다. 더 중요한 점은 많은 사람들이 서로 코드를 공유하고 질문을 할 수 있는 점이다.

Guido van Rossum 이 python 프로그래밍 언어를 만들었다. 1992년에 폭넓게 읽히는 Computer Programming for Everybody 를 같이 만들었다. 이것은 파이선으로 학생들에게 프로그래밍을 가르치는데 의욕적인 프로그램 비전을 세울 수 있게 한다. 10년 후에, 실제 이런 일이 라즈베리파이가 나오면서 실현되었다.

파이선은 인터프리터 언어이다. 즉, 코드를 컴파일 하여 machine code 로 변환하는 것보다는(컴파일 할 필요가 없다.) 프로그램이나 스크립트를 작성하여 직접 실행할 수 있다. 인터프리터 언어는 프로그램하는 것보다 약간 빠르다. 파이선에서는 변수가 숫자 타입, 리스트 타입, 문자열 타입을 별도로 정의하지 않아도 된다. 인터프리터는 스크립트가 실행 될 때, 데이터 타입을 인지한다.

파이선 인터프리터는 두 가지 방법으로 실행 될 수 있다. 셸에서 개별 명령을 입력하면서 실행 될 수 있고, standalone 스크립트를 실행하기 위해 라인단위의 명령을 통해서 실행될 수도 있다.

라즈베리파이에는 파이선을 위한 IDE(integrated development environment) 툴인 IDLE 가 포함되어 있다.

The Python Version Conundrum

라즈베리파이에 설치된 Python 버전이 두가지가 있어서 IDLE 역시 두가지 버전이 설치되어 있다.

이 책이 쓰여질때, Python 3 가 최신버전이고, 버전 2와 3이 하위 호환성이 없어졌다. Python3 이 폭넓게 사용되려면 몇 년 정도 걸릴 것이다. 배포된 패키지를 사용하는 많은 사용자들은 Python3 로 업그레이드 하지 않은 상태인데, 이 부분으로 Python 문서를 찾을 때 많은 혼란이 있다. 찾고자하는 올바른 Python 버전을 찾아야 한다.

현재 Python 최신 버전은 3.4.2 이다.

이 책에 있는 예제들은 Python 2.7 또는 3.X 에서 동작 된다.

3/Python On The Pi



만약, IDLE 를 실행할 때, TCP/IP port 에 대한 에러 메시지를 보면,
\$sudo ifconfig lo up

Hello Python

IDLE 3 을 실행하면 아래와 같은 `>>>` 프롬프트를 볼 수 있다.

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Mar 1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print("hello")
hello
>>> help
Type help() for interactive help, or help(object) for help about object.
>>> 3+4+5
12
>>> pinto
SyntaxError: invalid syntax
>>> print("dfe");
dfe
>>> import math
>>> (1+math.sqrt(5))/2
1.618033988749895
>>>
>>> radius = 200
>>> radius * 2 * math.pi
1256.6370614359173
>>>
```



IDLE can't bind to a TCP/IP port, which is necessary to communicate with its Python execution server. This might be because no networking is installed on this computer. Run IDLE with the -n command line switch to start without a subprocess and refer to Help/IDLE Help 'Running without a subprocess' for further details.

OK

IDLE 3 을 실행하면 아래와 같은 `>>>` 프롬프트를 볼 수 있다.

```
>>> print("Saluton Mondo!");
```

이 셸에서 계산도 할 수 있다.

```
>>> 3+4+5
```

```
12
```

셸에서 프로그래밍을 라인 단위로 수행 할 수 있다. 변수를 선언하거나 모듈을 import 할 수 있다.

```
>>> import math
```

```
>>> (1+math.sqrt(5)) / 2
```

```
1.618033988749895
```

math 모듈에 포함된 모든 함수를 사용할 수 있다.

```
>>> radius = 200
```

```
>>> radius * 2 * math.pi
```

```
125.66370614359173
```

3/Python On The Pi

모든 변수를 초기화 하거나 셸을 다시 시작 하고 싶으면, 메뉴에서, Shell → Restart Shell 을 선택한다. help() 커맨드를 이용해서 사용방법을 알아볼 수 있다.

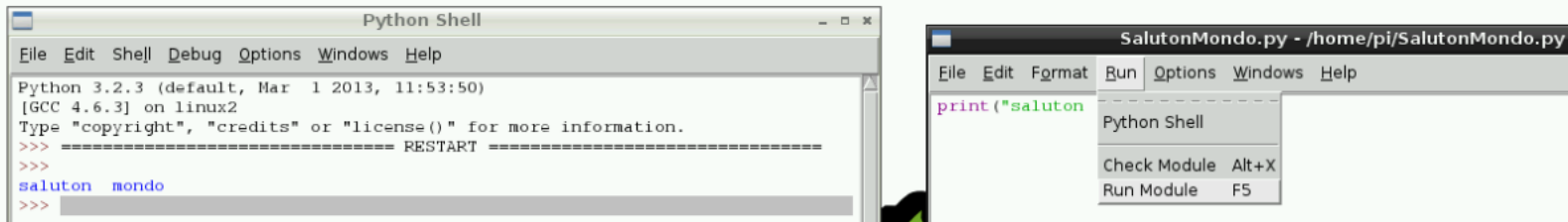
Python shell 프로그램에서 아래와 같이 입력하면 프롬프트가 **help>** 로 변경된다. 빠져나오기 위해서는 **quit** 라고 입력한다.

```
>>>help()
help>
```

이 상태에서 print, topics, keywords, modules 를 입력하는 각각에 대한 도움말을 확인할 수 있다.

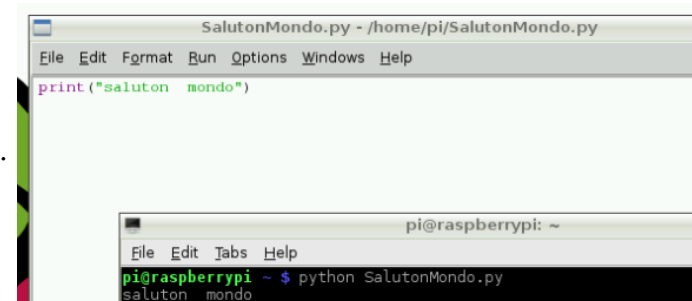
```
help("print")
help("topics")
help("keywords")
help("modules")
```

간단한 명령어나 동작을 테스트하는데 Python은 좋은 방법이다. Standalone Application 처럼 Python 스크립트를 실행할 수 있는데, File → New Window 를 수행하면 또하나의 창(Standalone application)이 수행되는 것을 확인할 수 있다.



Run → Run Module 를 수행하면 Python Shell 으로 결과가 출력된다. 파일을 저장한 후, 터미널에서 다음과 같이 수행해도 같은 결과를 얻을 수 있다.

```
$python SalutonMondo.py
```



3/Python On The Pi

Command Line vs. IDLE

Shell 에 출력을 하는 예제를 IDE에서 실행할 경우 IDLE 에 그 결과가 나오기까지 늦어지는 현상을 볼 수 있다.

IDLE를 통해서(Run Module) 수행하는 것이 Command Line 에서 수행하는 것보다는 느린 현상을 볼 수 있다.

Pi 의 한정된 리소스 때문인데, 이후의 예제는 command line 에서 수행 될 것이다. 하지만, IDLE 가 좋다면, 편집기로 여전 히 사용될 것이다.

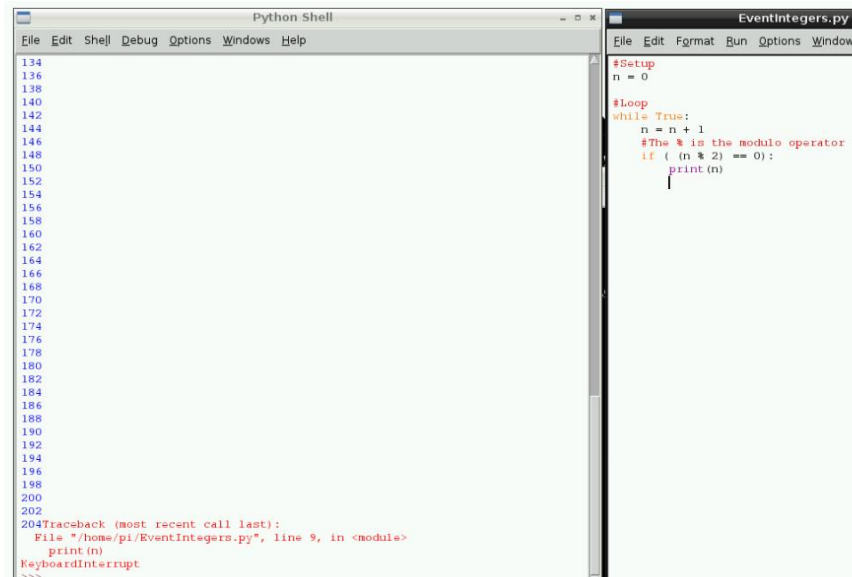
사실 VI 를 이용해서 작업하는 것이 IDLE 를 이용하는 것보다는 좋은 방법이다.

A Bit More Python

setup() 함수는 한번만 실행이 되고, loop() 함수는 계속해서 실행된다. Run Module 을 통해서 작성한 스크립트를 실행한다. 숫자가 계속해서 출력이되므로, 중지하기 위해서 Control-C 를 입력해준다.

```
# Setup
n = 0

# Loop
while True:
    n = n + 1
    # The % is the modulo operator
    if ((n % 2) == 0):
        print(n)
```



```
Python Shell
File Edit Shell Debug Options Windows Help
134
136
138
140
142
144
146
148
150
152
154
156
158
160
162
164
166
168
170
172
174
176
178
180
182
184
186
188
190
192
194
196
198
200
202
204Traceback (most recent call last):
  File "/home/pi/EventIntegers.py", line 9, in <module>
    print(n)
KeyboardInterrupt
>>>
```

```
EventIntegers.py
File Edit Format Run Options Window
#Setup
n = 0
#Loop
while True:
    n = n + 1
    #The % is the modulo operator
    if ( (n % 2) == 0):
        print(n)
```

3/Python On The Pi



Python에서 각각의 라인에서 들여쓰기가 공백(whitespace) 4 칸으로 작성되었다. Tab 키가 아니다. IDLE 에서는 tab을 사용할 수 있는데, Space 4개로 표기된다. Tab 키와 공백은 구분된다. Tab 키를 누르면 커서가 4칸 움직이는데 tab은 공백이 아니다. Space 키 4 번 누르는 것과 Tab 키는 다르다.

Python 에서 공백은 특별한 의미를 지닌다. Copy 나 Past 할때, 이 부분이 초보자에게는 큰 장애물이 될 수 있는데, 공백(whitespace)으로 들여쓰기를 하는 것이 Python 소스를 읽기 쉽게 해준다.

Python 이 문맥을 구분하는데 공백을 사용하는 언어이다. 다음예제에서, loop() 함수가 어떻게 들여쓰기 되는지 보자. Loop 함수의 끝이 들여쓰기가 어느 곳에서 되어야 함수의 끝이 결정되는지 유심히 보자. C 언어의 경우는 {} 기호를 사용해서 함수의 시작과 끝을 표기한다.

하나의 스크립트 혹은 파일에서 코드의 양이 클 경우 다른 곳에서 호출하려고 할 때 함수를 사용한다. 전에 예제를 함수를 가지고 다시 작성해 보자.

숫자102 이후 2로 나누어 떨어지는 숫자를 출력한다.

```
# Declare global variables
n = 0 ❶

# Setup function
def setup(): ❷
    global n
    n = 100

def loop(): ❸
    global n
    n = n + 1
    if ((n % 2) == 0):
        print(n)

# Main ❹
setup()
while True:
    loop()
```

- 1 – 어떤 함수에서도 사용할 수 있는 전역 변수를 선언한다.
- 2 – setup 함수를 정의 한다.
- 3 – loop 함수를 정의 한다.
- 4 – main 코드가 실행되는 블록이다. 한번만 호출된 후에, loop 함수가 실행된다.

각 함수 처음에 사용된 global 키워드는 중요하다. Interpreter 에게 함수 내에서만 사용되기 보다는 전역으로 선언된 n 변수를 사용하겠다고 알려준다.

이 설명이 다소 부족한 면이 있는데, Think Python, Python Pocket Reference 책으로 좀 더 자세하게 배울 수 있다. 이책 chapter 4 장에서도 기본적인 것과 이용할 수 있는 모듈을 사용한 예제를 볼 수 있다. PyGame 은 라즈베리파이에서 멀티미디어를 프로그램 해 볼 수 있는 훌륭한 방법이다.

3/Python On The Pi

Objects and Modules

이 책을 통해서 객체(objects) 와 모듈(module) 을 다루는 기본 문법을 이해 할 필요가 있다. Python은 34개의 예약된 키워드가 있는 언어이다. 이 키워드는 스크립트에서 구조와 제어의 흐름을 조절하게해주는 언어의 핵심이다.

Table 3-1. *Python has just 34 reserved keywords*

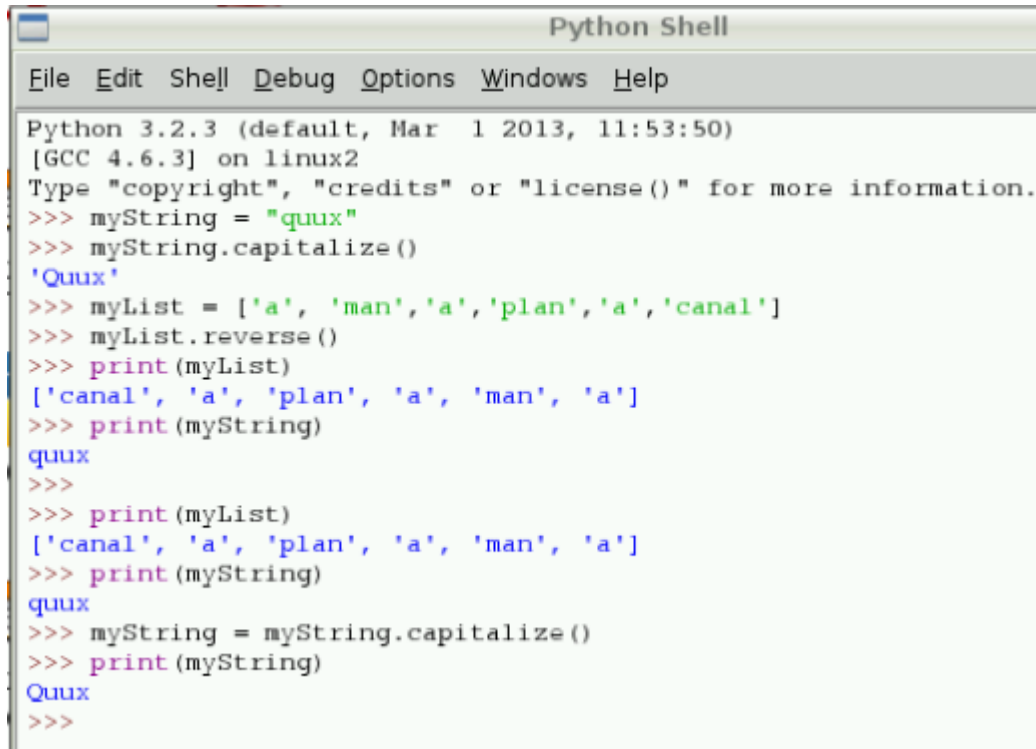
Conditionals	Loops	Built-in Functions	Classes, Modules, Functions	Error Handling
if	for	print	class	try
else	in	pass	def	except
elif	while	del	global	finally
not	break		lambda	raise
or	as		nonlocal	assert
and	continue		yield	with
is			import	
True			return	
False			from	
None				

3/Python On The Pi

Objects and Modules

파이선에서 문자열, 리스트, 함수, 모듈 그리고 숫자들은 모두 객체이다. 파이선 객체는 method, attribute 의 캡슐화된 집합체로 생각하면된다. 이런 attribute 와 method는 . **(dot syntax)**을 가지고 간단히 사용할 수 있다.

예를 들어, 파이선 셸 프롬프트에서 스트링의 함수를 호출하는 것을 보자

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code and its output:

```
Python 3.2.3 (default, Mar 1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> myString = "quux"
>>> myString.capitalize()
'Quux'
>>> myList = ['a', 'man', 'a', 'plan', 'a', 'canal']
>>> myList.reverse()
>>> print(myList)
['canal', 'a', 'plan', 'a', 'man', 'a']
>>> print(myString)
quux
>>>
>>> print(myList)
['canal', 'a', 'plan', 'a', 'man', 'a']
>>> print(myString)
quux
>>> myString = myString.capitalize()
>>> print(myString)
Quux
>>>
```



스트링과 리스트는 Standard library – 파이선에서 사용할 수 있는 built-in 된 라이브러리이다. C 의 stdio.h 에 해당한다. capitalize() 와 reverse() 같은 API 는 <https://docs.python.org/3/library/stdtypes.html#str.capitalize> 를 참조

어떤 함수가 있는지 확인하려면 파이선 shell 에서 help 를 이용하면 된다. help 프롬프트에서 string 을 입력하면 사용할 수 있는 함수 프로토타입을 확인할 수 있다.

A screenshot of the help prompt in a Python shell, showing the text "help> string" in a blue monospace font.

3/Python On The Pi

Objects and Modules

몇몇 standard library 모듈은 built-in 되어 있지 않다. 이런 경우는 명시적으로 **import** 명령어를 사용하여 사용하고자 하는 모듈을 선언해 줘야 한다.

```
>>>import time
```

다른 이름으로 사용할 수도 있다.

```
>>>import time as myTime
```

사용하려는 라이브러리에서 특정 함수만 사용할 수도 있다.

```
>>>from time import clock
```

Time 과 datetime 을 이용한 예제가 있다. 매 초마다 시간을 출력한다.

```
>>> from datetime import datetime
>>> from time import sleep
>>> while True:
    now = str(datetime.now())
    print(now)
    sleep(1)

2015-11-30 06:47:03.921013
2015-11-30 06:47:05.055717
2015-11-30 06:47:06.189851
2015-11-30 06:47:07.661627
2015-11-30 06:47:08.818654
2015-11-30 06:47:09.960961
2015-11-30 06:47:11.095137
2015-11-30 06:47:12.230883
2015-11-30 06:47:13.402394
2015-11-30 06:47:14.551520
2015-11-30 06:47:15.681833
```

Sleep 함수가 프로그램 수행을 1초 멈추게 한다. 약간의 시간차가 발생하게 되는데 두가지 이유가 있다.

1. 현재 시간을 계산하는데 걸리는 시간, 즉 datetime.now 함수의 수행 시간을 고려하지 않아서이다.
2. 다른 프로세서와 라즈베리파이의 CPU를 공유하고 있는 상태이고, 이 스크립트가 동작하는 중에도 CPU를 공유하고 있는 것이다. 라즈베리 파이는 실시간 real time OS가 아니다.

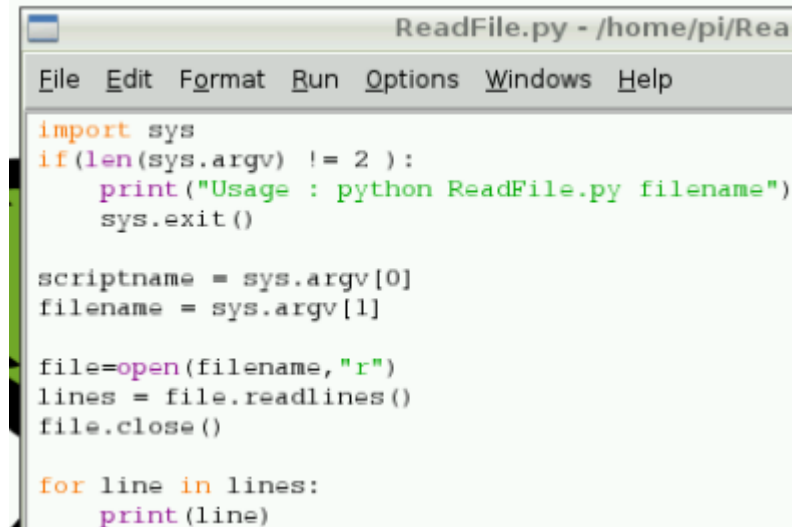


이때 사용되는 sleep(.9) 예제가 뭘 의미하는지 모르겠다. ReadFile.py 역시 마찬가지다.

3/Python On The Pi

Objects and Modules

파일을 라인단위로 읽어서 콘솔로 프린트해주는 예제

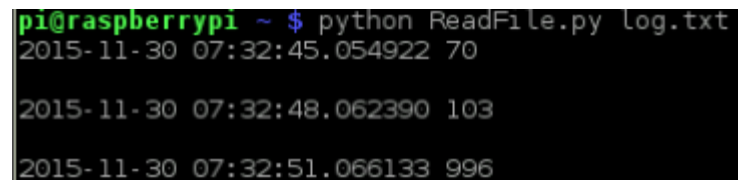
A screenshot of a text editor window titled "ReadFile.py - /home/pi/Rea". The editor has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code is as follows:

```
import sys
if(len(sys.argv) != 2 ):
    print("Usage : python ReadFile.py filename")
    sys.exit()

scriptname = sys.argv[0]
filename = sys.argv[1]

file=open(filename, "r")
lines = file.readlines()
file.close()

for line in lines:
    print(line)
```

A screenshot of a terminal window on a Raspberry Pi. The prompt is "pi@raspberrypi ~ \$". The command "python ReadFile.py log.txt" has been executed, and the output shows three lines from the log file, each with a timestamp and a value:

```
pi@raspberrypi ~ $ python ReadFile.py log.txt
2015-11-30 07:32:45.054922 70

2015-11-30 07:32:48.062390 103

2015-11-30 07:32:51.066133 996
```

3/Python On The Pi

Even More Modules

Table 3-2. *Some packages of particular interest to Pi users*

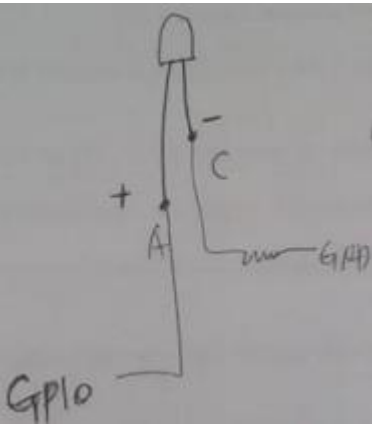
Module	Description	URL	Package Name
RPi.GPIO	Access to GPIO pins	http://code.google.com/p/raspberry-gpio-python/	python-rpi.gpio
Pygame	Gaming framework	http://pygame.org	python-pygame
SimpleCV	Easy API for Computer Vision	http://simplecv.org/	No package
Scipy	Scientific computing	http://www.scipy.org/	python-scipy
Numpy	The numerical underpinings of Scipy	http://numpy.scipy.org/	python-numpy
Flask	Microframework for web development	http://flask.pocoo.org/	python-flask
Requests	"HTTP for Humans"	http://ocs.python-requests.org	python-requests
PIL	Image processing	http://www.pythonware.com/products/pil/	python-imaging
wxPython	GUI framework	http://wxpython.org	python-wxgtk2.8
PySerial	Access to serial port	http://pyserial.sourceforge.net/	python-serial
pyUSB	FTDI-USB interface	http://bleyer.org/pyusb	No package

RPi.GPIO 를 이용해서 LED를 on/off 하는 예제를 작성해 보자

3/Python On The Pi

Even More Modules

RPi.GPIO 를 이용해서 LED를 on/off 하는 예제
아래에 GPIO 23 번에 LED를 연결한다.



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

print ("setup LED pin")

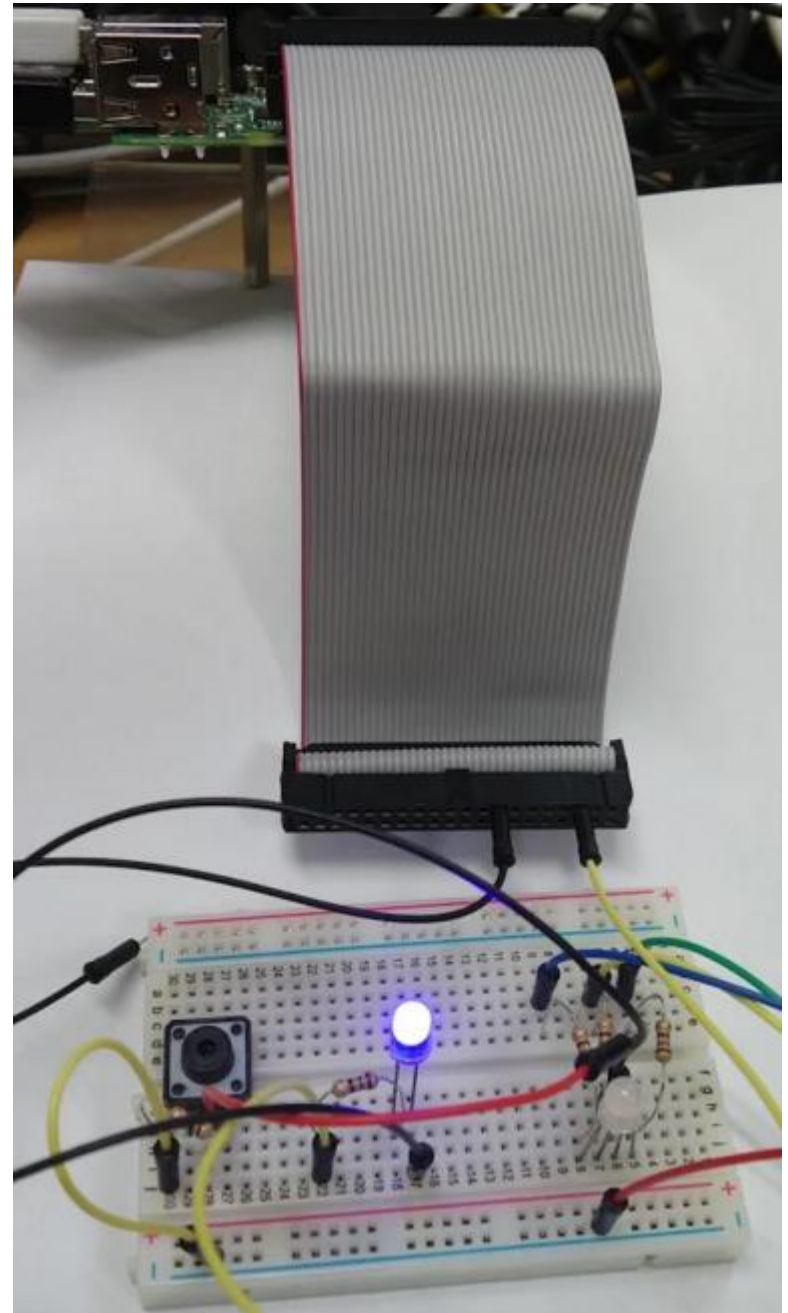
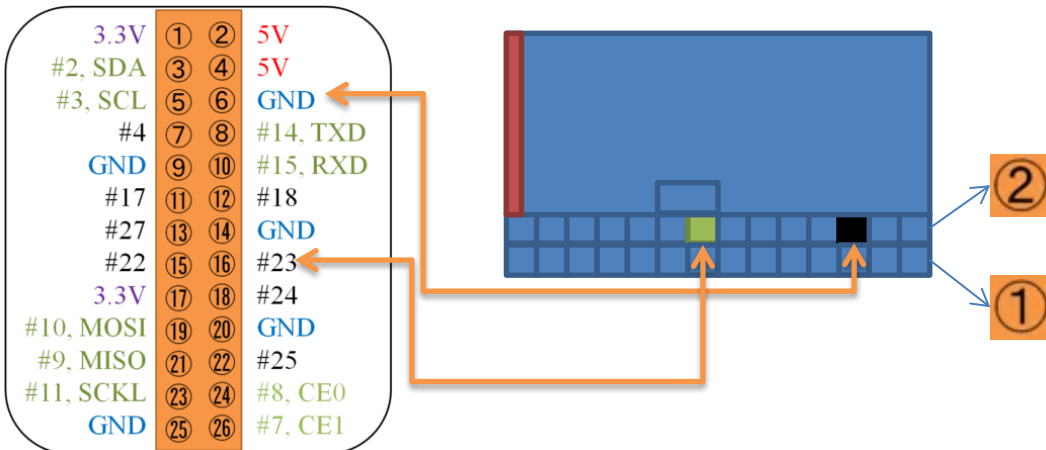
GPIO.setup(23,GPIO.OUT)
GPIO.output(23, False)

GPIO.output(23,True)

time.sleep(25)

GPIO.output(23,False)

GPIO.cleanup()
```



B+, GPIO



3.3V	1		2	5V
GPIO 2 (I2C1_SDA)	3		4	5V
GPIO 3 (I2C1_SCL)	5		6	GND
GPIO 4 (GPCLK0)	7		8	GPIO 14 (UART_TXD)
GND	9		10	GPIO 15 (UART_RXD)
GPIO 17	11		12	GPIO 18
GPIO 27	13		14	GND
GPIO 22	15		16	GPIO 23
3.3V	17		18	GPIO 24
GPIO 10 (SPI_MOSI)	19		20	GND
GPIO 9 (SPI_MISO)	21		22	GPIO 25
GPIO 11 (SPI_SCLK)	23		24	GPIO 8 (SPI_CE0)
GND	25		26	GPIO 7 (SPI_CE1)
ID_SD	27		28	ID_SC
GPIO 5	29		30	GND
GPIO 6	31		32	GPIO 12
GPIO 13	33		34	GND
GPIO 19	35		36	GPIO 16
GPIO 26	37		38	GPIO 20
GND	39		40	GPIO 21

Key

- Power (5 Volts)
- Power (3.3 Volts)
- Ground
- General Inputs/Outputs
- I2C Interface
- SPI Interface
- UART Interface
- ID EEPROM Interface

3/Python On The Pi

Troubleshooting Errors

스크립트를 작성하다보면 당연히 에러를 보게될 것이다. IDLE 의 Debug 메뉴를 사용하면 라인단위의 디버깅을 할 수 있다.

문맥에러-syntax error 는 다루기 쉬운 에러이다. 타이핑 실수나 공백 실수가 대부분이다. 반면에, semantic error 는 구분하기가 쉽지가 않다. 성능이 안나온다던지, 이런 부분을 디버깅하는데는 좀 노력을 하면된다.

디버깅하는 몇가지를 소개하면

- print 함수를 이용해서 실행되고 있는 위치를 알 수 있다.
- _print()+ 를 사용하여 변수를 출력해보고
- 괄호를 확인하고
- Syntax error는 interpreter 가 친절히 알려준다.
- 괄호를 사용해서 좀 더 명확히 우선순위를 정한다.

무엇보다도 많이 작성해 보아야 한다.

Going Further

<http://learnpythonthehardway.org/book/> ← 무료 ebook

<http://shop.oreilly.com/product/9780596158095.do>

<http://stackoverflow.com/>

<http://learnpythonthehardway.org/>

<http://shop.oreilly.com/product/9781593274078.do>

4/Animation and Multimedia in Python

Pygame 이라는 파이선에서 간단하게 게임을 개발할 수 있는 가벼운 프레임웍이 있다. 일반적인 멀티미디어를 구동할 때 사용할 수도 있다.

화면에 그리거나, 소리를 내거나, 키보드, 마우스를 처리하는 편리한 방법을 제공한다.

Pygame 은 Simple DirectMedia Layer(SDL) 을 기초로 만들어진 모듈이다. SDL은 라즈베리파이의 키보드, 마우스, 오디오 드라이버, 비디오 드라이버를 저수준에서 처리할 수 있다. Pygame 은 이런 SDL을 좀 더 다루기 쉽도록 만들어졌다.

이 장에서는 Pygame 멀티미디어의 기본에 중점을 둔다. 게임에 대한 다른 참고 할 수 있는 자료는 이 장 맨 끝에 알려주겠다.

Hello Pygame

Pygame 은 라즈베리파이에 이미 설치되어 있다. Pygame 은 Python3.0에서 동작이 되나 라즈베리에 있는 Pygame 은 Python 2.7 에서 동작된다. IDLE를 사용한다면 3.0 이 아닌 버전으로 변경해야 하고, 쉘에서 실행하는 경우 2.7 을 사용하도록 해야 한다. 파이에에는 2.7.3이 설치되어 있다.

윈도우에 원을 그리는 다음 예제를 보자.

```
import pygame ❶

width = 640 ❷
height = 480
radius = 100
fill = 1

pygame.init() ❸

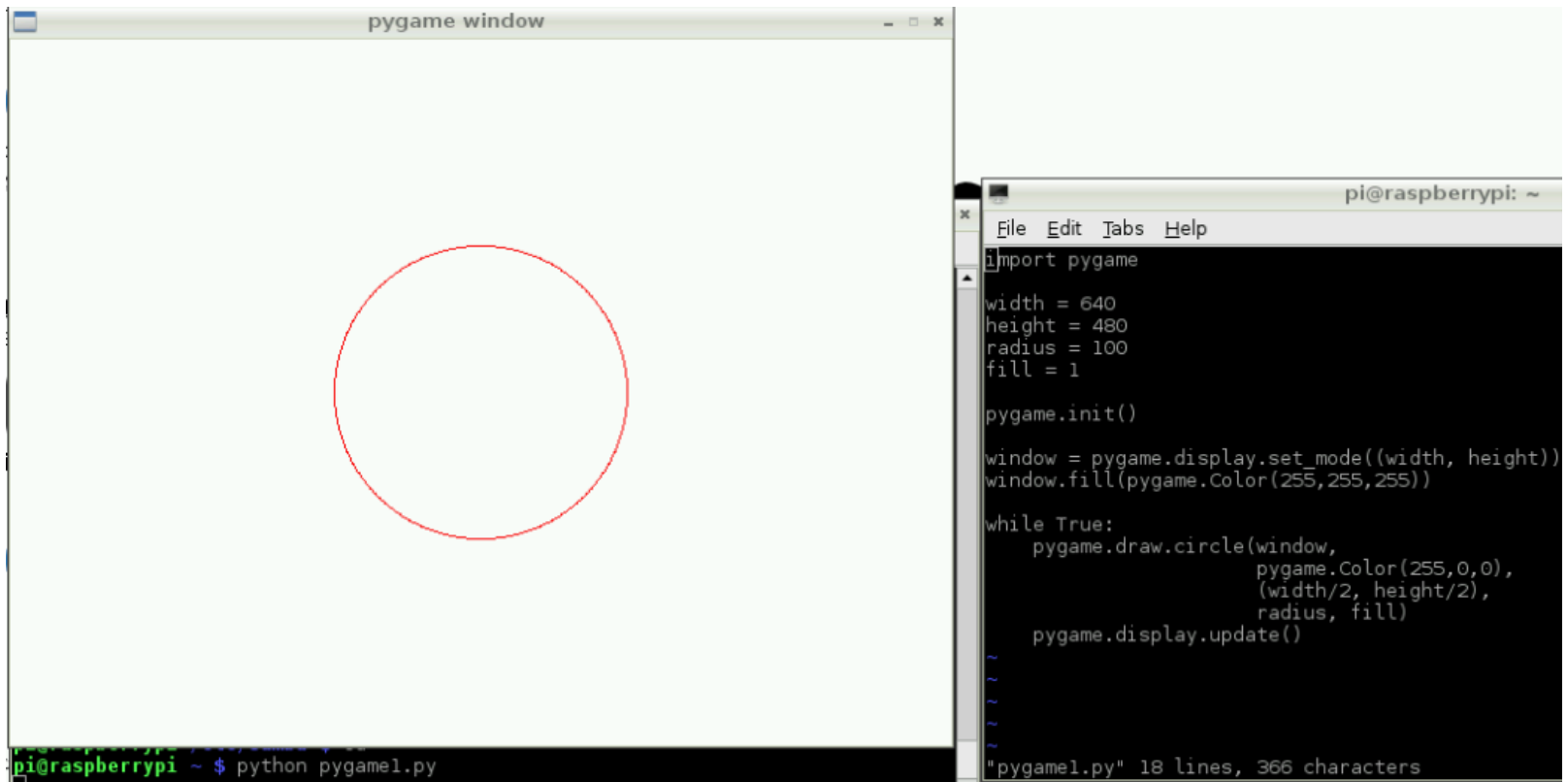
window = pygame.display.set_mode((width, height)) ❹

window.fill(pygame.Color(255, 255, 255)) ❺

while True: ❻
    pygame.draw.circle(window, ❼
                        pygame.Color(255, 0, 0),
                        (width/2, height/2),
                        radius, fill)
    pygame.display.update() ❽
```

4/Animation and Multimedia in Python

- 1 - Pygame module 을 import 한다.
- 2 - 전역변수를 정의
- 3 - pygame 모듈을 사용하기 위한 초기화 작업
- 4 - 정의된 윈도우를 만든다.
- 5 - white 색상으로 채우고
- 6 - 무한루프를 수행한다.
- 7 - red 로 원을 하나 그린다.
- 8 - 그림을 윈도우에 그리기 위해서는 screen buffer 라는 곳에 그리는데 이 screen buffer 는 그린다고 바로 보이는 것이 아니라. Screenshot 을 update 해줘야 보여진다.



4/Animation and Multimedia in Python



스크립트를 끝내려면, Control-C 를 사용해야 한다. Pygame 윈도우에 close 버튼을 추가하려면 다음 코드를 while 루프에 추가한다.

```
pygame.display.update()
if pygame.QUIT in [e.type for e in pygame.event.get()]:
    break
```

버튼을 눌렀을 경우, 트리거된 이벤트를 처리할 수 있다.

Pygame Surfaces

Pygame 에서 surface 는 하나의 사각형의 이미지를 의미한다. 게임이나 애니메이션에서 한 개의 장면을 구성하는 각각의 프레임에 surface 가 위치할 수 있다. 이 surface 의 픽셀은 RGB 의 숫자로 표현될 수 있다. Green 은 (0,255,0) 으로 표현된다. 추가적으로 투명도를 지정할 수 있다. (0,255,0,127) 마지막 4번째 숫자가 투명도를 설정하는데, 50%로 지정되었다.

pygame.display window 는 surface의 기본구조이다. Pygame.display 모듈은 display window 에 대한 정보와 조정(control)할 수 있다. set_mode() 함수는 새로운 윈도우를 만들고, update() 함수는 매 프레임마다 그 윈도우의 display 를 다시 그려준다.

Surface 를 필름 한 장 이라고 생각하면 된다. 우리는 이 필름에 그림을 그릴 것이고, 이 필름(surface) 몇 개를 모아서 frame 을 구성할 수도 있고, 매 frame 마다 surface 에 있는 것을 update() 함수를 이용해서 그릴 수 있는 것이다.

보여지는 surface 에 그림 파일을 그리기 위해서는 pygame.image 모듈의 load() 함수를 이용한다.

4/Animation and Multimedia in Python

Pygame Surfaces

```
import pygame

pygame.init()
screen= pygame.display.set_mode((350,200))
background = pygame.image.load("toy1.jpg") # 1
background.convert_alpha() # 2
screen.blit(background, (0,0)) # 3
while True:
    pygame.display.update()
```



- 1 – 배경화면에 보여질 그림을 로드한다.
- 2 – `convert_alpha` 함수는 현재 display 에 해당하는 surface 형식으로 바꿔준다. 해줄 필요는 없지만, 이미지 로드하는 시간을 단축하기 위해 필요하다. (실제 해보니 별 차이를 못 느낌)
- 3 – 기본 배경화면은 검정색이다. `blit()` 함수는 검은색 배경화면으로 배경 surface 를 조합한다.

4/Animation and Multimedia in Python

Pygame Surfaces

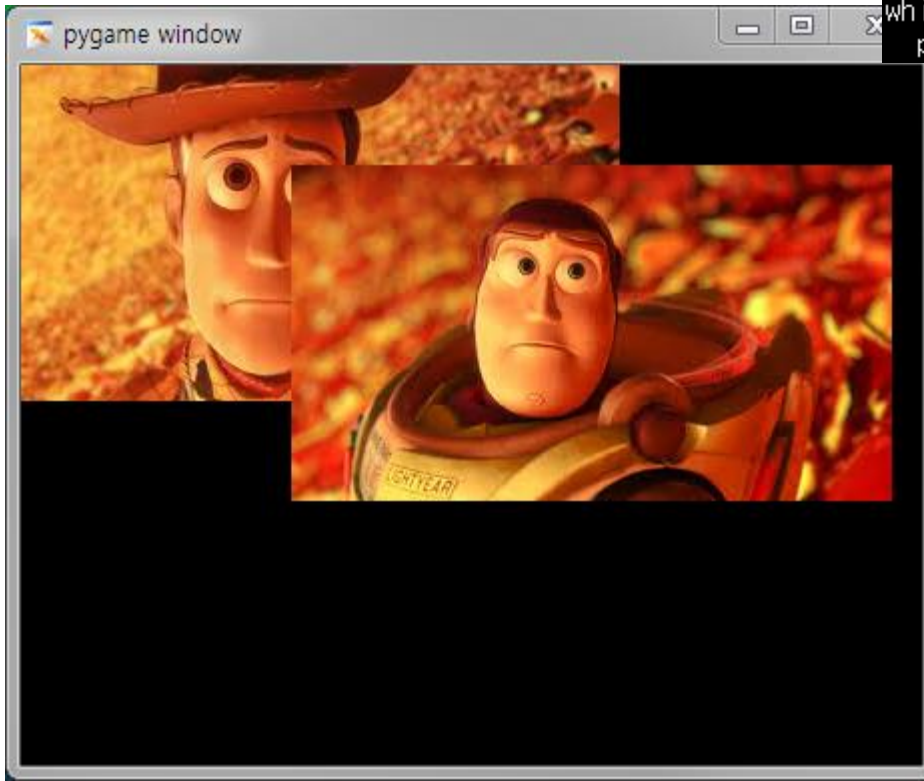
2 개의 이미지를 같이 표시하는 예제

```
import pygame

pygame.init()
screen= pygame.display.set_mode((450,350))
background = pygame.image.load("toy1.jpg").convert_alpha()
theremin = pygame.image.load("toy2.jpg").convert_alpha()

screen.blit(background, (0,0))
screen.blit(theremin, (135,50))

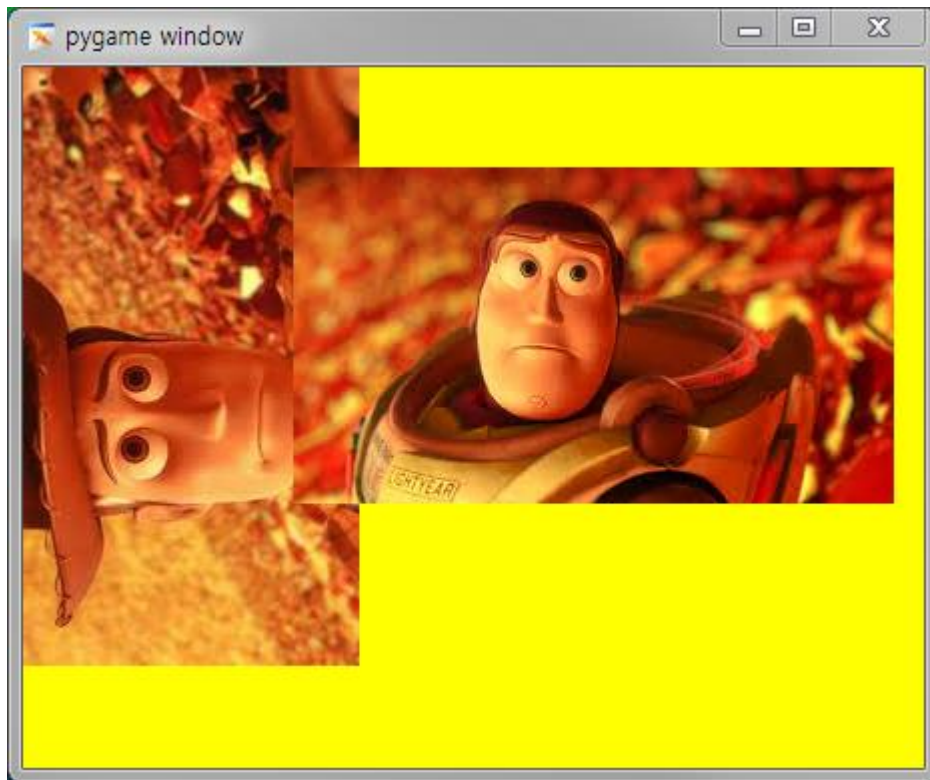
while True:
    pygame.display.update()
```



4/Animation and Multimedia in Python

Pygame Surfaces

Pygame.transform 모듈은 surface 를 회전하거나 크기를 조절할 수 있는 함수를 제공한다.



```
import pygame

yellow=(255,255,0)

pygame.init()
screen= pygame.display.set_mode((450,350))
background = pygame.image.load("toy1.jpg").convert_alpha()
theremin = pygame.image.load("toy2.jpg").convert_alpha()

screen.fill(yellow)
background = pygame.transform.rotate(background,90)
screen.blit(background, (0,0))
screen.blit(theremin, (135,50))

while True:
    pygame.display.update()
```

pygame 의 모듈에 대한 사용방법은 아래 사이트를 참고하라.

<https://pygame.org/wiki/index>

4/Animation and Multimedia in Python

Drawing on Surfaces

Text, 글씨를 그리기 위해서는 Font 객체를 만들어야 한다.(pygame.font 모듈을 사용하면 된다.)
먼저 라즈베리 파이에서 사용가능한 font 목록을 get_fonts() 함수를 이용하여 알아 볼 수 있다.

freeserif 라는 폰트를 사용하기 위해 SysFont 객체를 사용한다.

```
import pygame
pygame.init()

screen = pygame.display.set_mode((725, 92))
font = pygame.font.SysFont("freeserif", 72, bold = 1)
textSurface = font.render("1 Theremin Per Child!", 1,

pygame.Color(255, 255, 255))
screen.blit(textSurface, (10, 10))

while True:
    pygame.display.update()
```



더 다양한 폰트를 사용하기 위해서는 아래를 실행한다.

```
sudo apt-get install ttf-mscorefonts-installer
```

```
sudo apt-get install ttf-liberation
```

4/Animation and Multimedia in Python

Handling Events and Inputs

Pygame 에서, 마우스 click, move 그리고 key 입력같은 이벤트는 Event 객체에서 발생여부를 알 수 있고, 이벤트를 event queue 에 넣을 수 있다. 이벤트들은 프로그램에서 처리가 될때까지 이벤트 큐에 쌓여 있다. 이런 일을 할 수 있게 pygame.event 를 이용할 수 있다.

아래 예제는, 이벤트 큐를 사용하는 방법을 말해주고 있다. 빨간색 원을 그리는 프로그램을 확장하여, 윈도우의 중심으로 갈 수록 원이 작아지고, 윈도우의 끝쪽으로 갈 수록 원이 커지는 프로그램이다.

```
import pygame
from pygame.locals import * #

width, height = 640, 640
radius = 0
mouseX, mouseY = 0, 0 #2

pygame.init()
window = pygame.display.set_mode((width, height))
window.fill(pygame.Color(255, 255, 255))

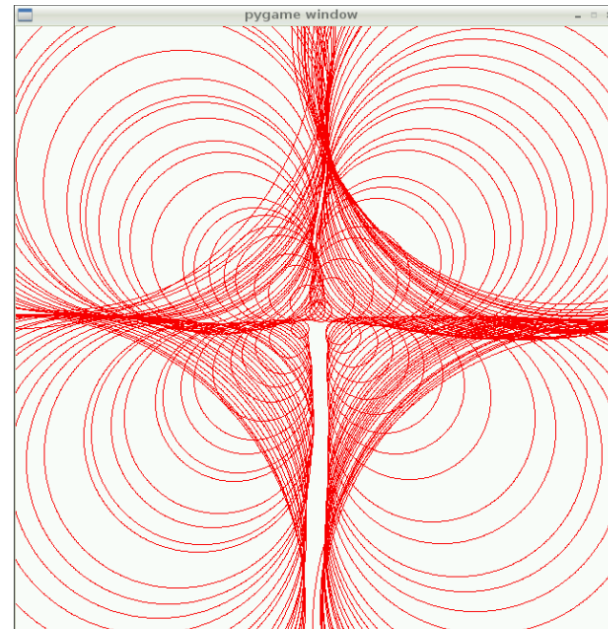
fps = pygame.time.Clock() #3

while True: #4
    for event in pygame.event.get(): #5
        if event.type == MOUSEMOTION: #6
            mouseX, mouseY = event.pos
            if event.type == MOUSEBUTTONDOWN: #7
                window.fill(pygame.Color(255, 255, 255))
            radius = (abs(width/2 - mouseX)+abs(height/2 - mouseY))/2 + 1 #8
            pygame.draw.circle(window, pygame.Color(255, 0, 0), (mouseX, mouseY), radius, 1) #9
        pygame.display.update()
    fps.tick(30) #10
```


4/Animation and Multimedia in Python

Handling Events and Inputs

- 1 - pygame.locals 모듈은 MOUSEMOTION 같은 상수를 정의한다. pygame. 같은 전처리 없이 상수 값 그대로 사용할 수 있다.
- 2 - 마우스 좌표를 저장 하는 변수
- 3 - 프레임 카운터로 사용할 객체를 초기화 한다.
- 4 - 무한 루프, 루프 한번이 하나의 프레임을 그린다.
- 5 - 이벤트 큐를 얻는다. 이벤트 변수는 다음번 이벤트가 처리 될때까지 그 값을 유지하게 된다.
- 6 - mouse move 이벤트일 경우, 마우스 위치를 다시 수정해 준다.
- 7 - mouse click 일 경우, 화면을 지워준다.
- 8 - 다양한 지름이 윈도우 중심으로부터 떨어져 있을 수록 크게, 중심에서 가까울 수록 작게 설정된다.
- 9 - 해당하는 지름크기의 원을 그린다.
- 10 - 30 fps, 즉 30 tick 만큼 기다린다. 이벤트를 처리하고 원을 그릴때까지 시간을 확보해야 한다. 3 으로 변경해서 그려지는 형태를 관찰해 본다.



4/Animation and Multimedia in Python

Handling Events and Inputs

이벤트와 사용자 입력을 처리할 수 있는 다른 모듈에는 아래와 같은 것이 있다.

pygame.time

- 시간에 관한 처리를 할 수 있는 모듈

pygame.mouse

- 마우스에 관한 처리를 할 수 있는 모듈

pygame.key

- 입력한 key 및 처리를 할 수 있는 모듈

pygame.joystick

- joystick 관련된 모듈

```
import pygame
import random
from time import sleep
running = True
pygame.init()
screen=pygame.display.set_mode((0,0), pygame.FULLSCREEN)
while running:
    pygame.draw.circle(
        screen,
        pygame.Color(int(random.random()*255),
                     int(random.random()*255),
                     int(random.random()*255)),
        (int(random.random()*1500),
         int(random.random()*1500)),
        int(random.random()*500), 0)
    pygame.display.update()
    sleep(.1)
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            running = False
pygame.quit()
```

Going Full Screen

Pygame.FULLSCREEN 플래그를 설정하여 전체화면으로 스크린을 만든다. Ctrl-C 를 누르기 전까지는 key-down, 즉 어떤 키를 누를 때까지 반복된다.



4/Animation and Multimedia in Python

Sprites

게임에서 움직이거나 조정할 수 있는 그래픽 요소들은 *sprites*에 의해 처리 된다. `pygame.sprite` 모듈은 스크린에 sprite 을 그릴 수 있는 기본적인 함수와 그들 사이의 문제점을 처리할 수 있는 함수를 제공한다.

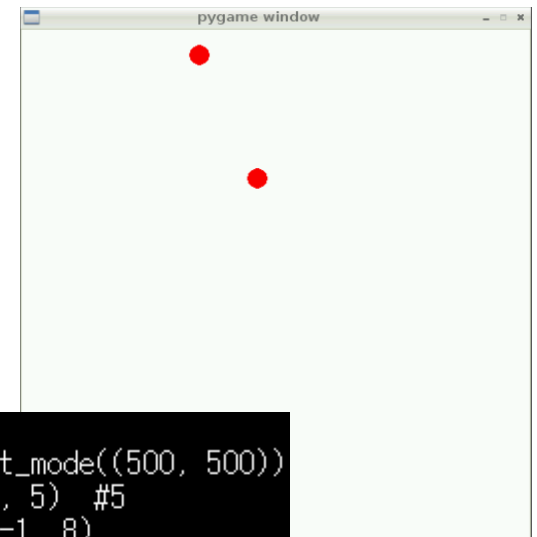
Sprites 는 같은 코드를 공유하는 스크린에서 사용되는 객체를 만들때 유용하다. 몇 개의 sprites 를 만들고 처리하는 방법을 다음예제에서 볼 수 있다. 시작점, 방향, 스피드를 갖는 ball 을 추가할 수 있다.

```
import pygame
class Ball(pygame.sprite.Sprite): #1
    def __init__(self, x, y, xdir, ydir, speed): #2
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface([20, 20])
        self.image.fill(pygame.Color(255, 255, 255))
        pygame.draw.circle(self.image,
                           pygame.Color(255,0,0),
                           (10,10), 10, 0)

        self.rect = self.image.get_rect()
        self.x, self.y = x, y #3
        self.xdir, self.ydir = xdir, ydir
        self.speed = speed

    def update(self): #4
        self.x = self.x + (self.xdir * self.speed)
        self.y = self.y + (self.ydir * self.speed)
        if (self.x < 10) | (self.x > 490):
            self.xdir = self.xdir * -1
        if (self.y < 10) | (self.y > 490):
            self.ydir = self.ydir * -1
        self.rect.center = (self.x, self.y)

pygame.init()
```



```
fps = pygame.time.Clock()
window = pygame.display.set_mode((500, 500))
ball = Ball(100, 250, 1, 1, 5) #5
ball2 = Ball(400, 10, -1, -1, 8)

while True:
    ball.update() #6
    ball2.update()
    window.fill(pygame.Color(255,255, 255))
    window.blit(ball.image, ball.rect) #7
    window.blit(ball2.image, ball2.rect)
    pygame.display.update()
    fps.tick(30)
```

4/Animation and Multimedia in Python

Sprites

- 1 – class 를 선언한다. Sprites 객체에 기반한 Ball 클래스이다. Sprite 을 그리는 자신만의 함수를 정의할 수 있고 sprite 이 매 시간 갱신되는 것을 정의 할 수 있다.
- 2 – Ball 함수가 호출 될 때, 새로운 ball 객체를 생성한다.
- 3 – x, y 변수는 Ball instance 의 변수이다. 움직이는 ball 의 x,y 좌표를 instance 의 x, y 변수로 복사 된다.
- 4 – 각 프레임마다 update() 함수가 호출 된다.
- 5- 시작 위치,방향,속도가 다른 2개의 ball 을 생성한다.
- 6 – 매 프레임마다 ball 의 새로운 위치를 갱신한다.
- 7 – 현재 위치에 해당하는 ball 을 그린다.

Playing Sound

먼저 사운드 모듈이 제대로 로드 되었는지 확인한다.



```
$sudo apt-get install alsa-utils  
$sudo modprobe snd_bcm2835  
$sudo amixer cset numid=3 n ← 출력단자를 지정한다. 1 – analog, 2 – hdmi  
$sudo aplay loop_compus.wav ← 이어폰을 통해서 소리를 들을 수 있다.
```

Pygame 은 사운드 파일을 로드한후, pygame.mixer 모듈을 통해서 플레이 할 수 있다.또한, pygame.midi 모듈을 통해서 MIDI 이벤트를 실행 중인 다른 Pi 로 전달 할 수 있다.

```
import pygame.mixer  
from time import sleep  
pygame.mixer.init(48000, -16, 1, 1024)  
sound = pygame.mixer.Sound("loop_compus.wav")  
channelA = pygame.mixer.Channel(1)  
channelA.play(sound)  
sleep(2.0)
```



4/Animation and Multimedia in Python

Playing Sound

사운드 드라이버가 로드 되었는지 확인해본다.

```
$lsmod | grep bcm
```

```
snd_bcm2835          21342  0
```

```
snd_pcm              93100  1 snd_bcm2835
```

```
snd                  67211  5 snd_bcm2835,snd_timer,snd_pcm,snd_seq,snd_seq_device
```

볼륨을 설정해 준다.

```
$amixer cset numid=3 1
```

```
import pygame
```

```
import pygame.midi #1
```

```
pygame.init()
```

```
pygame.midi.init() #2
```

```
for id in range(pygame.midi.get_count()): #3
```

```
print pygame.midi.get_device_info(id) #4
```

1 – midi 모듈은 기본 pygame 모듈에는 포함이 되지 않는데 별도로 import 해준다.

2 – midi 는 별도로 초기화 해 준다.

3- get_count() 함수는 Pi 에서 동작할 수 있는 midi 장치 숫자를 리턴한다.

4- 그 정보를 출력한다.

외부 midi keyboard를 장착할 경우는 이런 정보를 확인할 수 있다.

```
('ALSA', 'USB Uno MIDI Interface MIDI 1', 0, 1, 0)
```

```
('ALSA', 'USB Uno MIDI Interface MIDI 1', 1, 0, 0)
```

```
pi@raspberrypi /home/sghong $ python midilist.py
```

```
('ALSA', 'Midi Through Port-0', 0, 1, 0)
```

```
('ALSA', 'Midi Through Port-0', 1, 0, 0)
```

```
pi@raspberrypi /home/sghong $
```

첫번째는 ALSA를 사용하고 있고, 두번째는 midi port 번호를 보여준다. 그리고 마지막 3개의 숫자는 각각 입력,출력,사용여부를 표시한다.

Midi Through Port-0 (0, 1, 0)

0번 port 를 사용하고, 출력용도, 사용중이 아님을 표기

Midi Through Port-0 (1, 0, 0)

0번 port 를 사용하고, 입력 용도, 사용중이 아님을 표기

4/Animation and Multimedia in Python

Playing Video

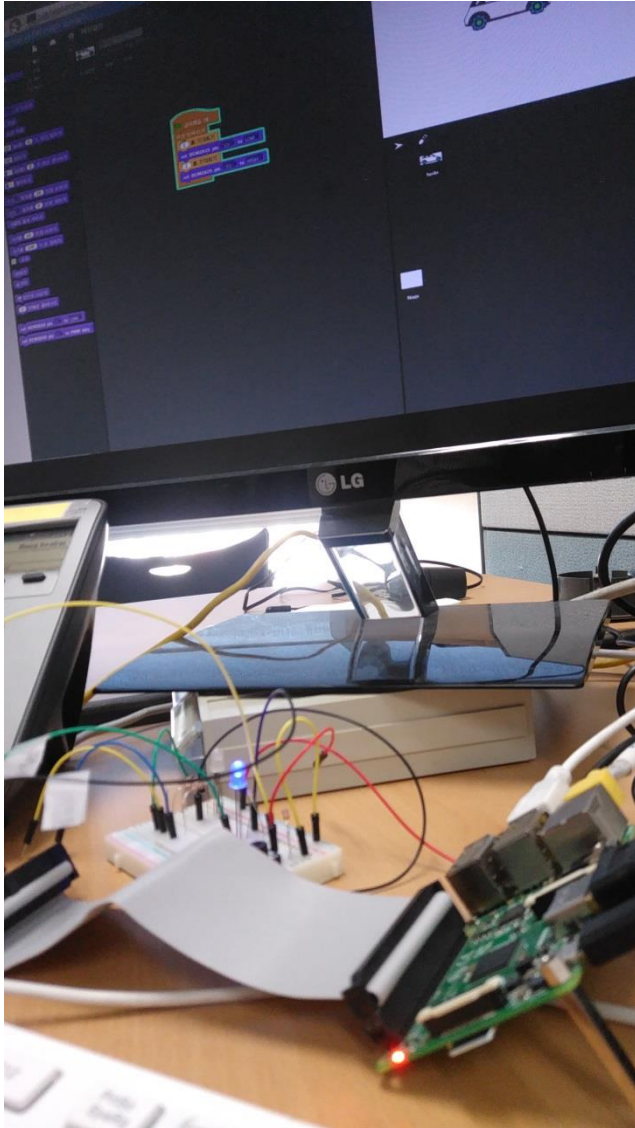
Pygame 은 또한 pygame.movie 모듈을 통해서 video를 play 할 수 있다. Video 는 MPEG1 format 만 지원된다.



omxplayer 를 통해서 video를 플레이한다.
\$ omxplayer Ex_gyro.avi

5/Scratch on the Pi

본 파트에서 설명하는 scratch 에 관련된 사항은 산딸기 마을의 PiSnap 이라는 아주 훌륭한 글로 대신 하려 한다.



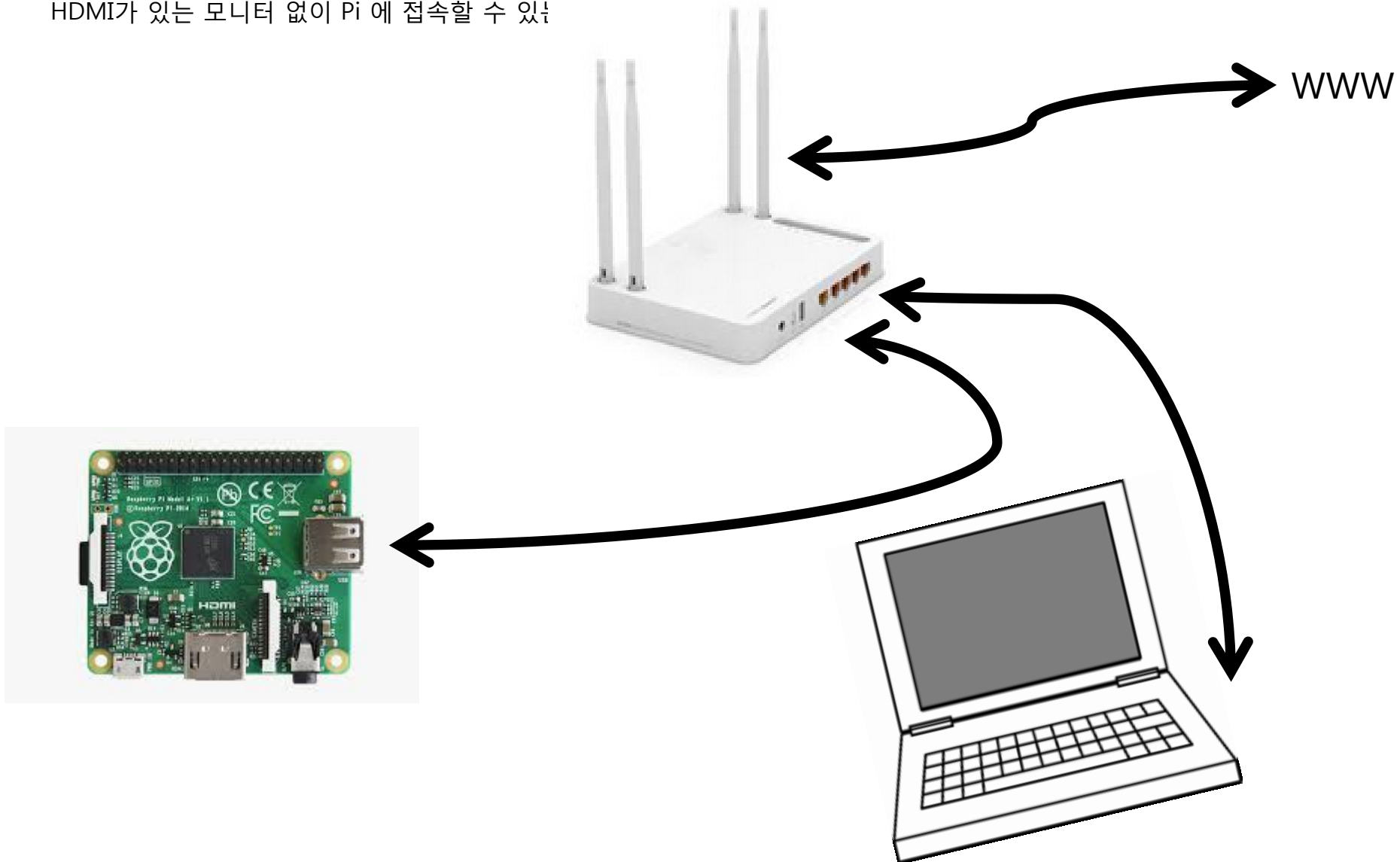
옆의 그림도 PiSnap 을 이용하여 LED on/off 를 수행해 본 것이다.

<http://www.rasplay.org/?p=23946>

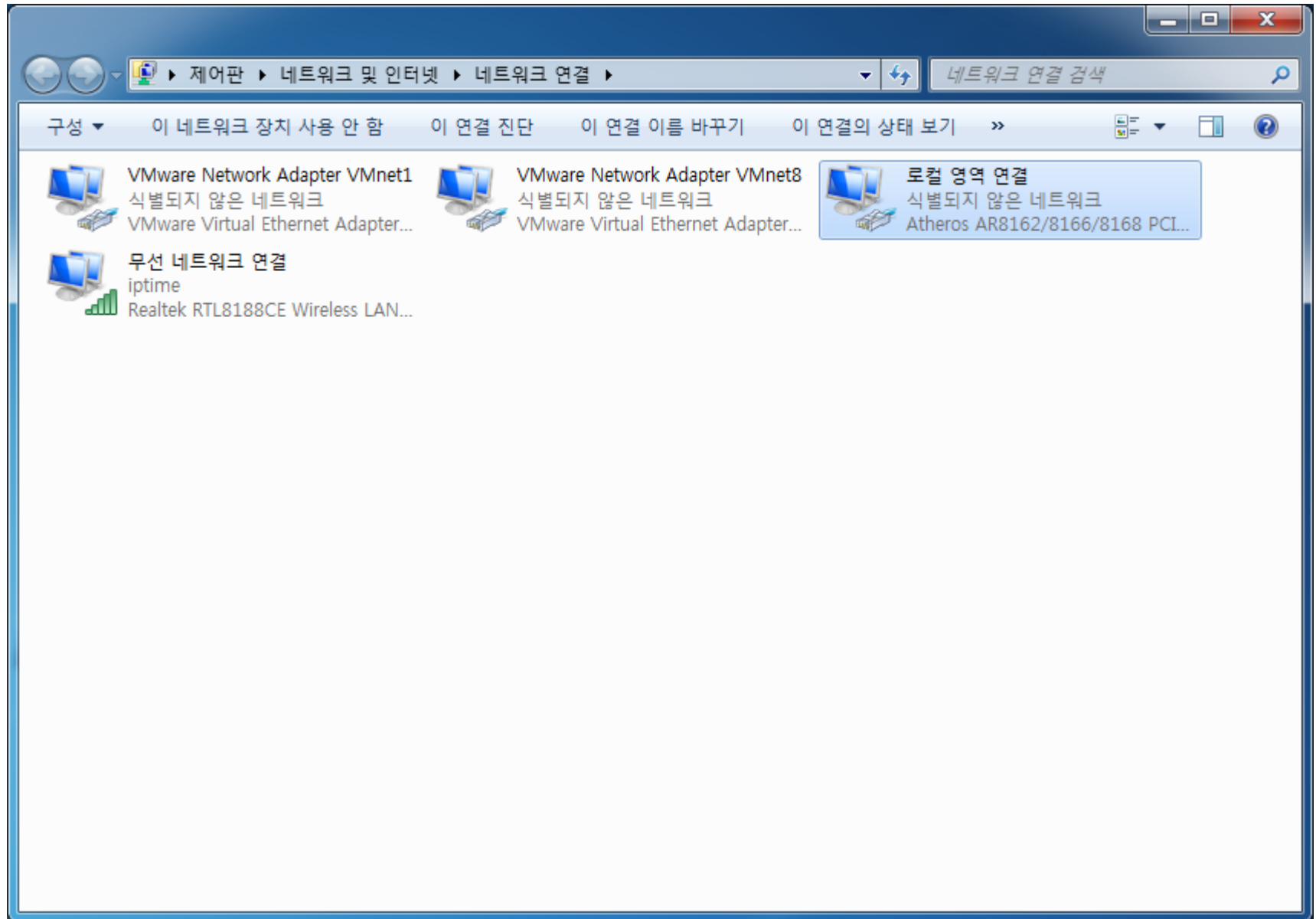
Connect Raspberry with putty

다음은 Putty 를 이용해서 Pi 에 SSH 로 접속하는 방법과 VNC를 통해서 접속하는 방법을 알아본다.

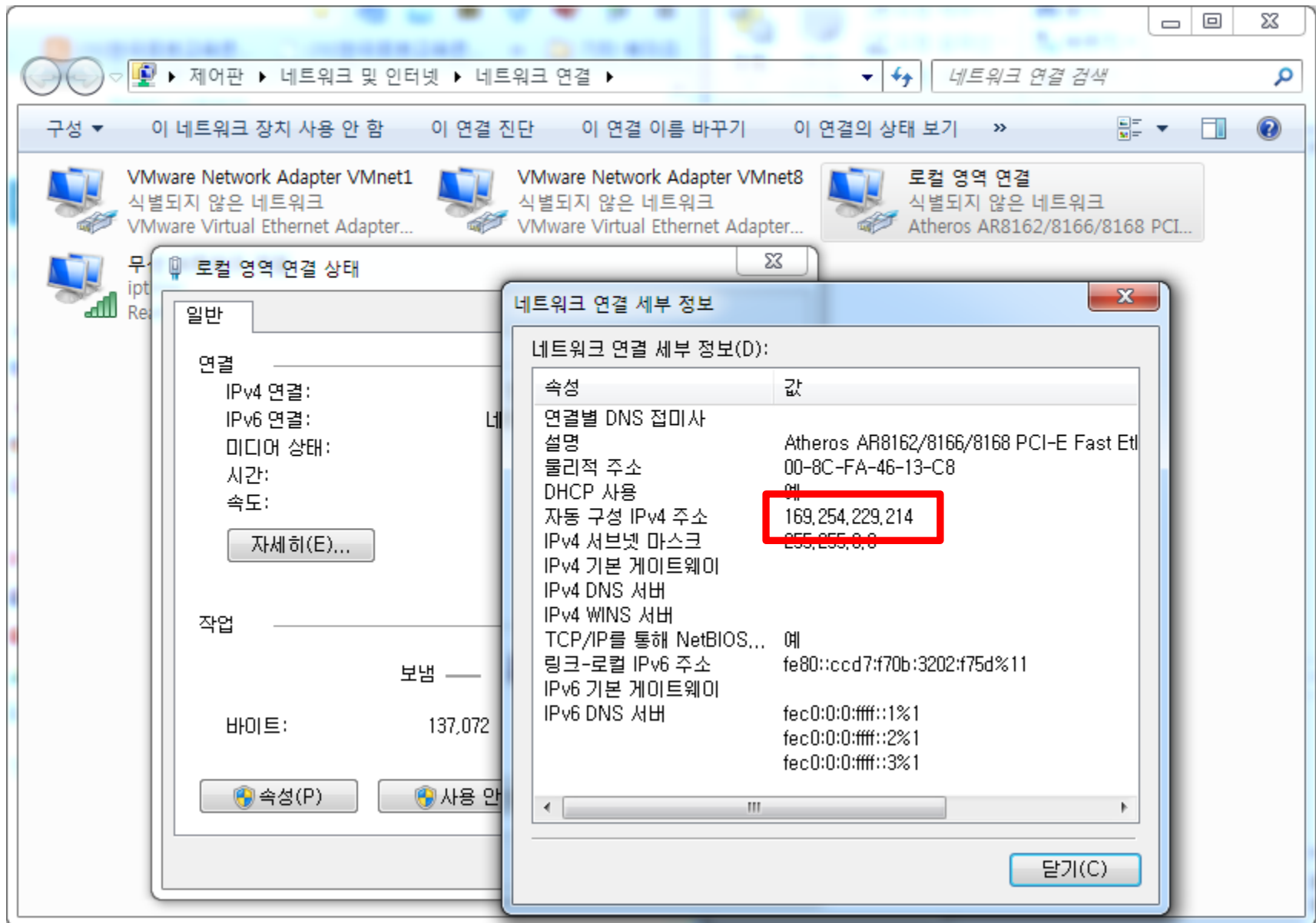
HDMI가 있는 모니터 없이 Pi 에 접속할 수 있다



Connect Raspberry with putty



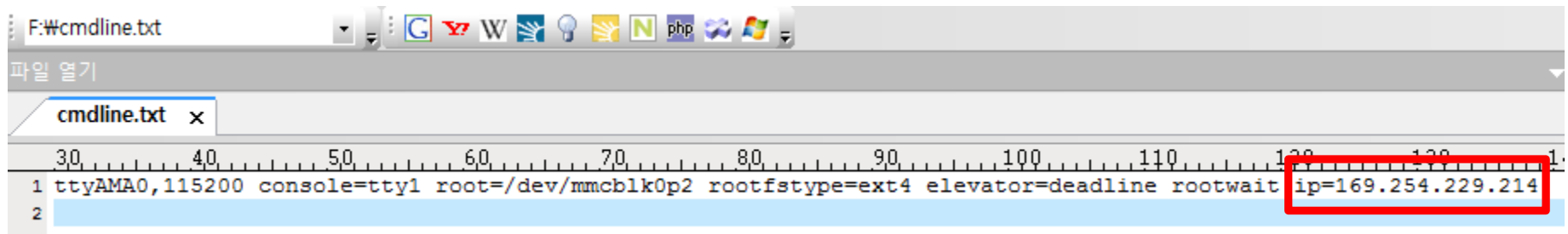
Connect Raspberry with putty



Connect Raspberry with putty

라즈베리 파이의 SDCARD 를 카드리더기로 읽어, cmdline.txt 파일을 편집기로 열면 아래와 같이 볼 수 있다.

제일 끝에 `ip=169.254.229.214` 을 아래와 같이 입력 한다.



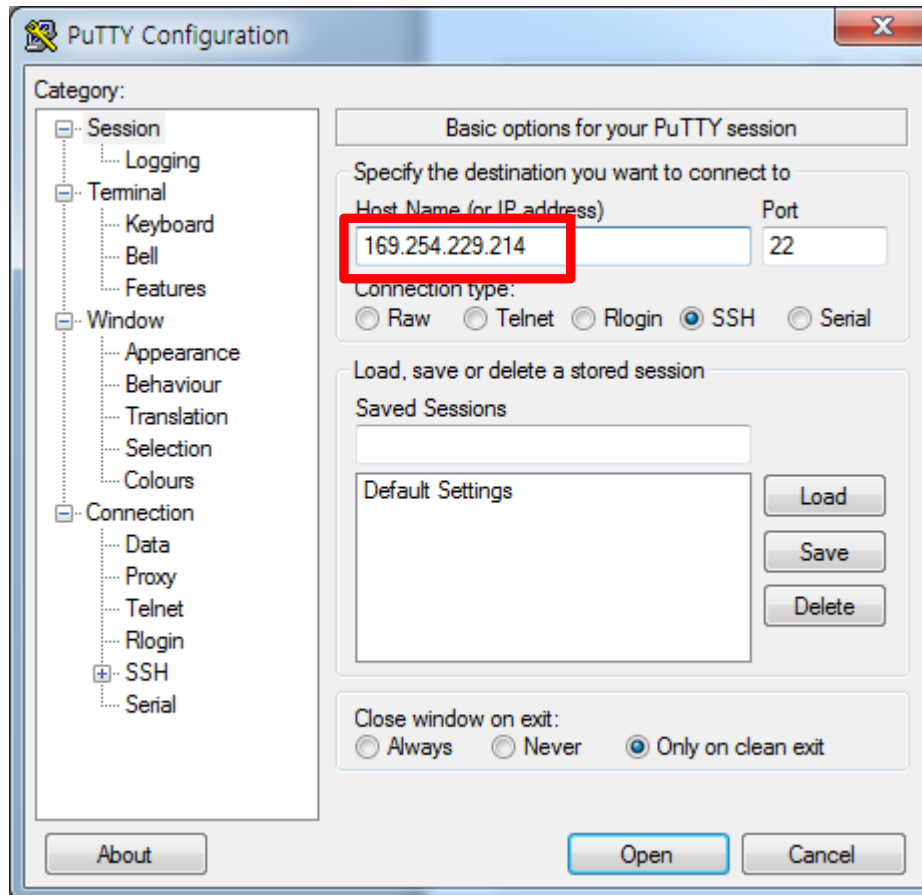
```
F:\cmdline.txt
파일 열기
cmdline.txt x
3,0 4,0 5,0 6,0 7,0 8,0 9,0 10,0 11,0 12,0 13,0 14,0
1 ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait ip=169.254.229.214
2
```

이후, SDCARD 를 라즈베리에 다시 장착 한 후, 부팅한다.

이렇게 cmdline.txt 에 ip 를 적어 놓아도 되지만, /etc/network/interfaces 파일을 수정해서 ip를 변경할 수 있다.

Connect Raspberry with putty

Putty SSH설정은 아래와 같이 한다.

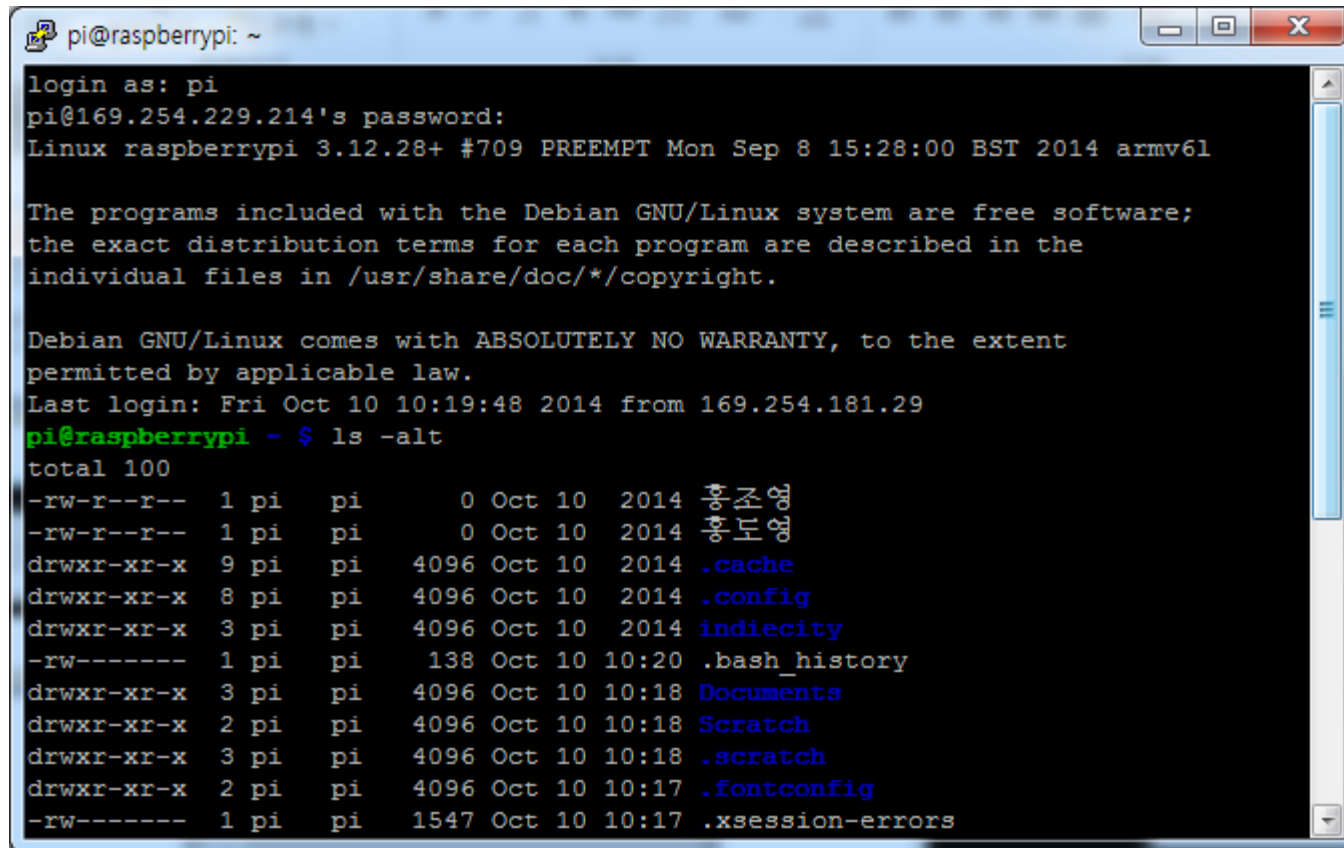


Connect Raspberry with putty



Connect Raspberry with putty

pi/raspberry



```
pi@raspberrypi: ~
login as: pi
pi@169.254.229.214's password:
Linux raspberrypi 3.12.28+ #709 PREEMPT Mon Sep 8 15:28:00 BST 2014 armv6l

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: Fri Oct 10 10:19:48 2014 from 169.254.181.29
pi@raspberrypi ~$ ls -alt
total 100
-rw-r--r--  1 pi   pi       0 Oct 10  2014 홍조영
-rw-r--r--  1 pi   pi       0 Oct 10  2014 홍도영
drwxr-xr-x  9 pi   pi    4096 Oct 10  2014 .cache
drwxr-xr-x  8 pi   pi    4096 Oct 10  2014 .config
drwxr-xr-x  3 pi   pi    4096 Oct 10  2014 indiecity
-rw-----  1 pi   pi     138 Oct 10 10:20 .bash_history
drwxr-xr-x  3 pi   pi    4096 Oct 10 10:18 Documents
drwxr-xr-x  2 pi   pi    4096 Oct 10 10:18 Scratch
drwxr-xr-x  3 pi   pi    4096 Oct 10 10:18 .scratch
drwxr-xr-x  2 pi   pi    4096 Oct 10 10:17 .fontconfig
-rw-----  1 pi   pi    1547 Oct 10 10:17 .xsession-errors
```

VNC on Pi

라즈베리파이에 VNC server 를 설치한다.

\$sudo apt-get update

```
pi@raspberrypi /etc/init.d $ sudo apt-get update
Get:1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:2 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]
Get:3 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Get:4 http://raspberrypi.collabora.com wheezy Release [7,514 B]
Get:5 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:6 http://archive.raspberrypi.org wheezy Release [10.2 kB]
Get:7 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,214 B]
Get:8 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,897 kB]
Get:9 http://archive.raspberrypi.org wheezy/main armhf Packages [108 kB]
Hit http://repository.wolfram.com stable Release.gpg
Hit http://repository.wolfram.com stable Release
Hit http://repository.wolfram.com stable/non-free armhf Packages
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Ign http://repository.wolfram.com stable/non-free Translation-en_GB
Ign http://repository.wolfram.com stable/non-free Translation-en
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Hit http://mirrordirector.raspbian.org wheezy/contrib armhf Packages
Hit http://mirrordirector.raspbian.org wheezy/non-free armhf Packages
Hit http://mirrordirector.raspbian.org wheezy/rpi armhf Packages
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 7,041 kB in 2min 20s (50.2 kB/s)
Reading package lists... Done
```

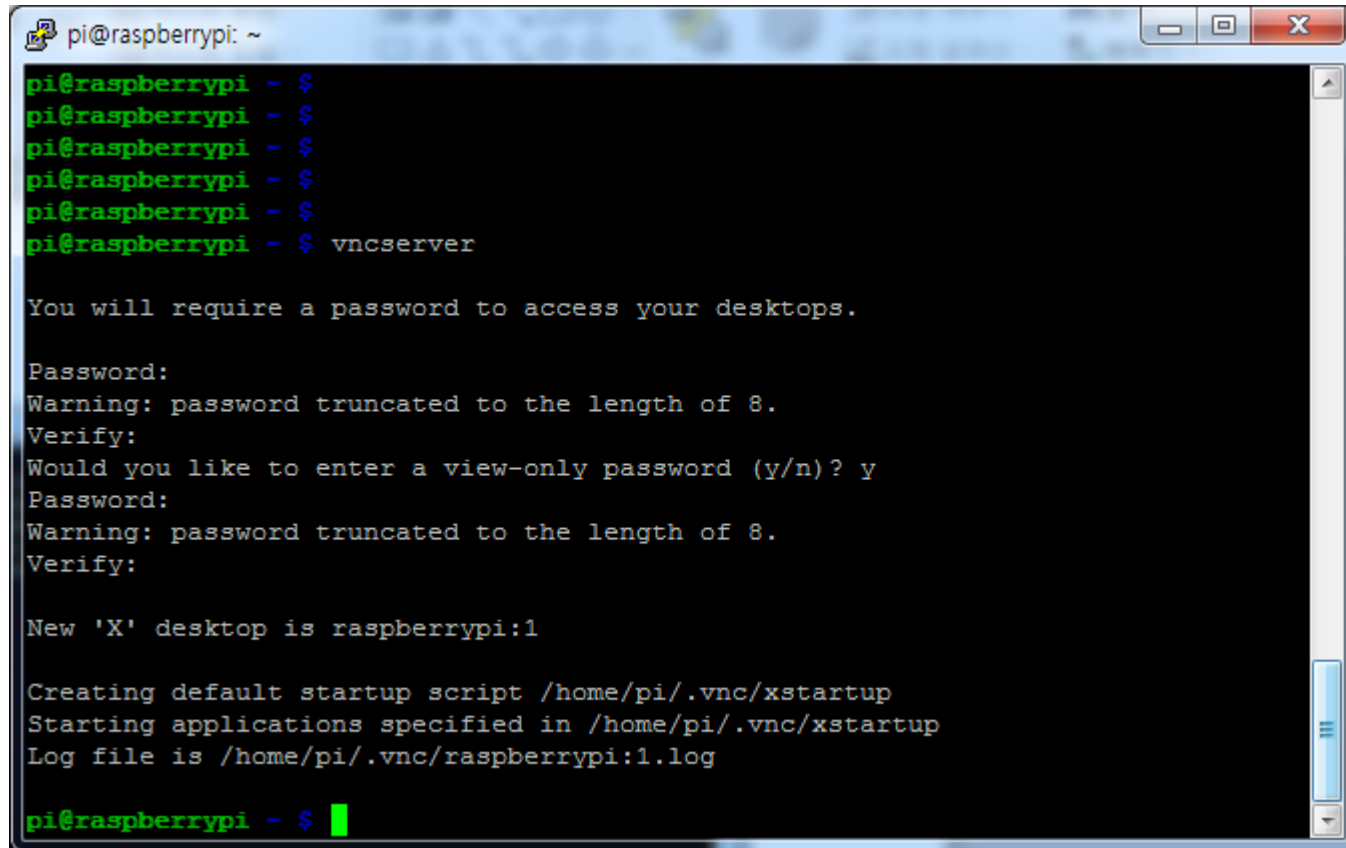
VNC on Pi

```
$sudo apt-get  
install  
tightvncserver
```

```
pi@raspberrypi /etc/network $ sudo apt-get install tightvncserver
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  x11-xserver-utils xfonts-base
Suggested packages:
  tightvnc-java nickle cairo-5c xorg-docs-core
The following NEW packages will be installed:
  tightvncserver x11-xserver-utils xfonts-base
0 upgraded, 3 newly installed, 0 to remove and 103 not upgraded.
Need to get 7,148 kB of archives.
After this operation, 10.4 MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main tightvncserver ar
mhf 1.3.9-6.4 [786 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main x11-xserver-utils
armhf 7.7~3 [181 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main xfonts-base all 1
:1.0.3 [6,181 kB]
Fetched 7,148 kB in 11s (628 kB/s)
Selecting previously unselected package tightvncserver.
(Reading database ... 73301 files and directories currently installed.)
Unpacking tightvncserver (from .../tightvncserver_1.3.9-6.4_armhf.deb) ...
Selecting previously unselected package x11-xserver-utils.
Unpacking x11-xserver-utils (from .../x11-xserver-utils_7.7~3_armhf.deb) ...
Selecting previously unselected package xfonts-base.
Unpacking xfonts-base (from .../xfonts-base_1%3a1.0.3_all.deb) ...
Processing triggers for man-db ...
Processing triggers for menu ...
Processing triggers for fontconfig ...
Setting up tightvncserver (1.3.9-6.4) ...
update-alternatives: using /usr/bin/tightvncserver to provide /usr/bin/vncserver (vncserver) in auto mode
update-alternatives: using /usr/bin/Xtightvnc to provide /usr/bin/Xvnc (Xvnc) in auto mode
update-alternatives: using /usr/bin/tightvncpasswd to provide /usr/bin/vncpasswd (vncpasswd) in auto mode
Setting up x11-xserver-utils (7.7~3) ...
Setting up xfonts-base (1:1.0.3) ...
Processing triggers for menu ...
pi@raspberrypi /etc/network $
```


VNC on Pi

vncserver 를 실행 한 후
아래 암호를 `vncserver` 라고 3번 입력한다.

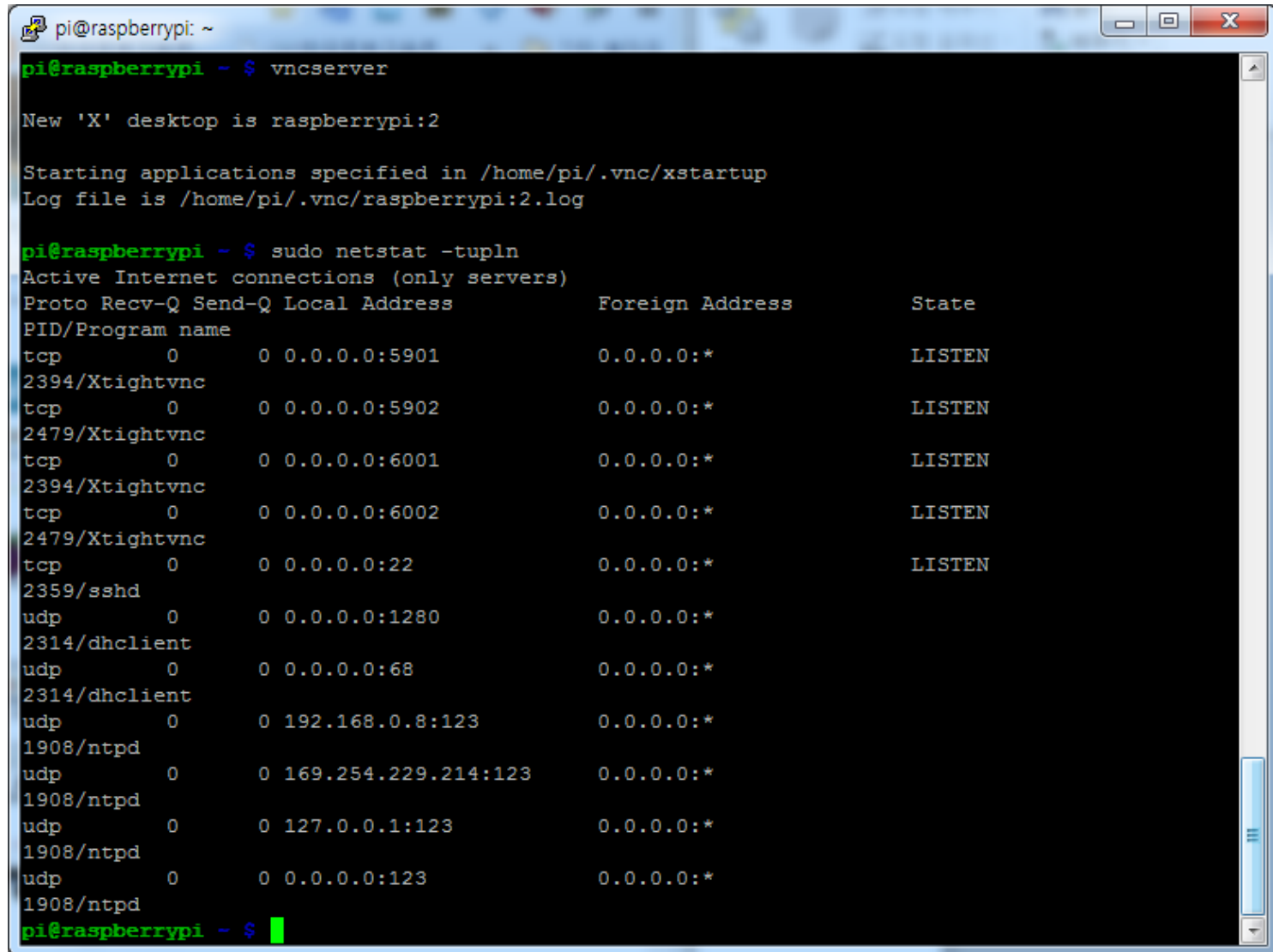
A terminal window titled 'pi@raspberrypi: ~' with standard window controls. The terminal shows the execution of 'vncserver' and its output. The output includes prompts for a password and verification, a warning about password truncation, a confirmation to use a view-only password, and the successful creation and startup of a new 'X' desktop environment named 'raspberrypi:1'. The terminal ends with a new command prompt.

```
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $ vncserver  
  
You will require a password to access your desktops.  
  
Password:  
Warning: password truncated to the length of 8.  
Verify:  
Would you like to enter a view-only password (y/n)? y  
Password:  
Warning: password truncated to the length of 8.  
Verify:  
  
New 'X' desktop is raspberrypi:1  
  
Creating default startup script /home/pi/.vnc/xstartup  
Starting applications specified in /home/pi/.vnc/xstartup  
Log file is /home/pi/.vnc/raspberrypi:1.log  
  
pi@raspberrypi ~ $
```

VNC on Pi

vncserver 가 실행 중인 것을 아래 명령으로 확인 한다.

\$sudo netstat -tupln

A terminal window titled 'pi@raspberrypi: ~' with standard window controls. It shows the execution of 'vncserver' and 'sudo netstat -tupln'. The vncserver output indicates a new desktop is starting. The netstat output lists active internet connections for servers, showing several Xtightvnc instances listening on ports 5901, 5902, 6001, and 6002, along with other system services like sshd, dhclient, and ntpd.

```
pi@raspberrypi ~ $ vncserver

New 'X' desktop is raspberrypi:2

Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:2.log

pi@raspberrypi ~ $ sudo netstat -tupln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:5901             0.0.0.0:*               LISTEN
2394/Xtightvnc
tcp        0      0 0.0.0.0:5902             0.0.0.0:*               LISTEN
2479/Xtightvnc
tcp        0      0 0.0.0.0:6001             0.0.0.0:*               LISTEN
2394/Xtightvnc
tcp        0      0 0.0.0.0:6002             0.0.0.0:*               LISTEN
2479/Xtightvnc
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN
2359/sshd
udp        0      0 0.0.0.0:1280            0.0.0.0:*
2314/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*
2314/dhclient
udp        0      0 192.168.0.8:123         0.0.0.0:*
1908/ntpd
udp        0      0 169.254.229.214:123    0.0.0.0:*
1908/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*
1908/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*
1908/ntpd
pi@raspberrypi ~ $
```

VNC on Pi

다음 부팅때에 vncserver 를 자동으로 실행 시키기 위해서는

tightvncserver 라는 파일을 아래와 같이 /etc/init.d/ 에 작성 한다.

```
$pwd  
/etc/init.d  
$sudo vi tightvncserver  
$sudo chmod 755 tightvncserver  
$sudo update-rc.d tightvncserver defaults  
update-rc.d: using dependency based boot sequencing
```

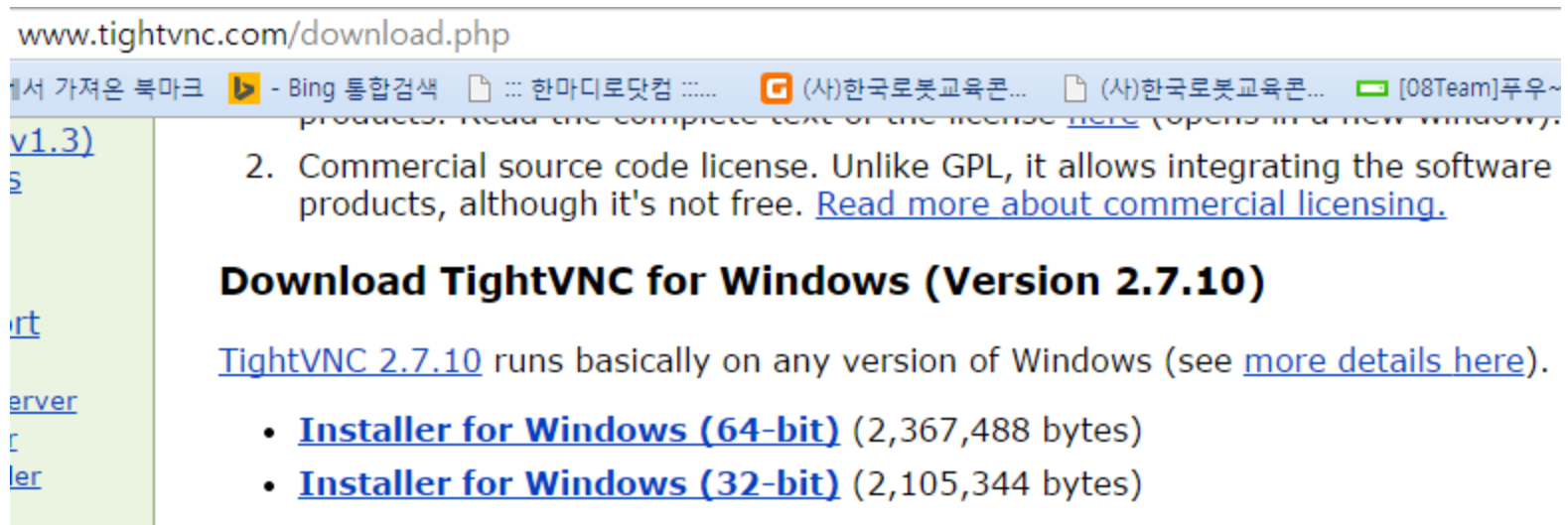
이렇게 해도 다음 부팅때 자동 실행이 안되서

/home/pi/.bashrc 파일 제일 마지막에
vncserver 를 적어준다.

```
### BEGIN INIT INFO  
# Provides: vncboot  
# Required-Start: $remote_fs $syslog  
# Required-Stop: $remote_fs $syslog  
# Default-Start: 2 3 4 5  
# Default-Stop: 0 1 6  
# Short-Description: Start VNC Server at boot time  
# Description: Start VNC Server at boot time.  
### END INIT INFO  
#!/bin/sh  
# /etc/init.d/vncboot  
USER=root  
HOME=/root  
export USER HOME  
case "$1" in  
start)  
echo "Starting VNC Server"  
#Insert your favoured settings for a VNC session  
/usr/bin/vncserver :0 -geometry 1280x800 -depth 16 -pixelformat rgb565  
"  
stop)  
echo "Stopping VNC Server"  
/usr/bin/vncserver -kill :0  
"  
*)  
echo "Usage: /etc/init.d/vncboot {start|stop}"  
exit 1  
"  
esac  
exit 0
```

VNC on Pi

vncviewer 는 여러가지 어플리케이션이 있는데
그 중 하나인 tight VNC를 통해 접속한다.
아래에서 다운로드 한 후, 설치를 진행 한다.



The screenshot shows a web browser window with the address bar displaying www.tightvnc.com/download.php. The browser's address bar also shows several tabs, including "에서 가져온 북마크", "Bing 통합검색", and "(사)한국로봇교육콘...". The main content area of the page is titled "Download TightVNC for Windows (Version 2.7.10)". Below the title, there is a paragraph stating "TightVNC 2.7.10 runs basically on any version of Windows (see [more details here](#)).". To the right of this paragraph, there is a list of two download links: "Installer for Windows (64-bit) (2,367,488 bytes)" and "Installer for Windows (32-bit) (2,105,344 bytes)".

www.tightvnc.com/download.php

에서 가져온 북마크 - Bing 통합검색 ... 한마디로닷컴 ... (사)한국로봇교육콘... (사)한국로봇교육콘... [08Team]푸우~

v1.3)

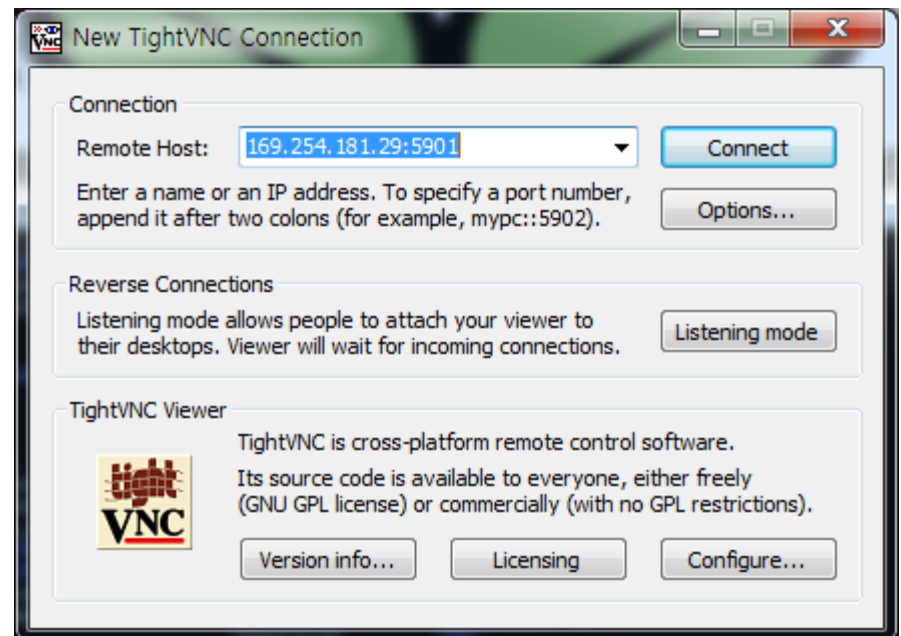
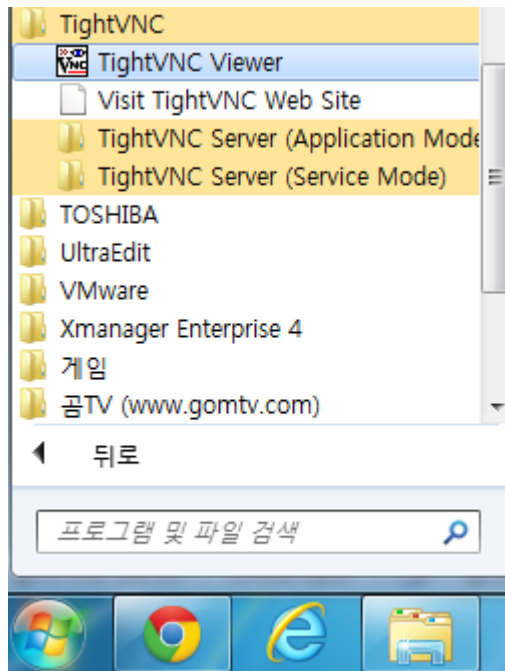
2. Commercial source code license. Unlike GPL, it allows integrating the software products, although it's not free. [Read more about commercial licensing.](#)

Download TightVNC for Windows (Version 2.7.10)

[TightVNC 2.7.10](#) runs basically on any version of Windows (see [more details here](#)).

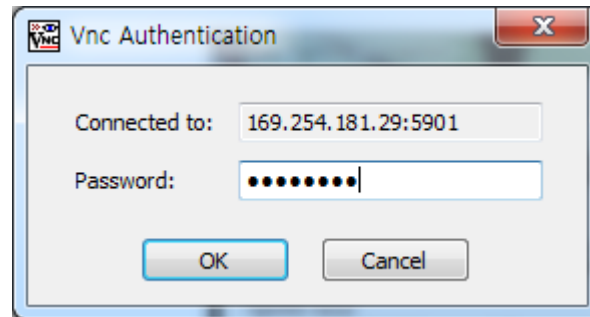
- [Installer for Windows \(64-bit\)](#) (2,367,488 bytes)
- [Installer for Windows \(32-bit\)](#) (2,105,344 bytes)

VNC on Pi

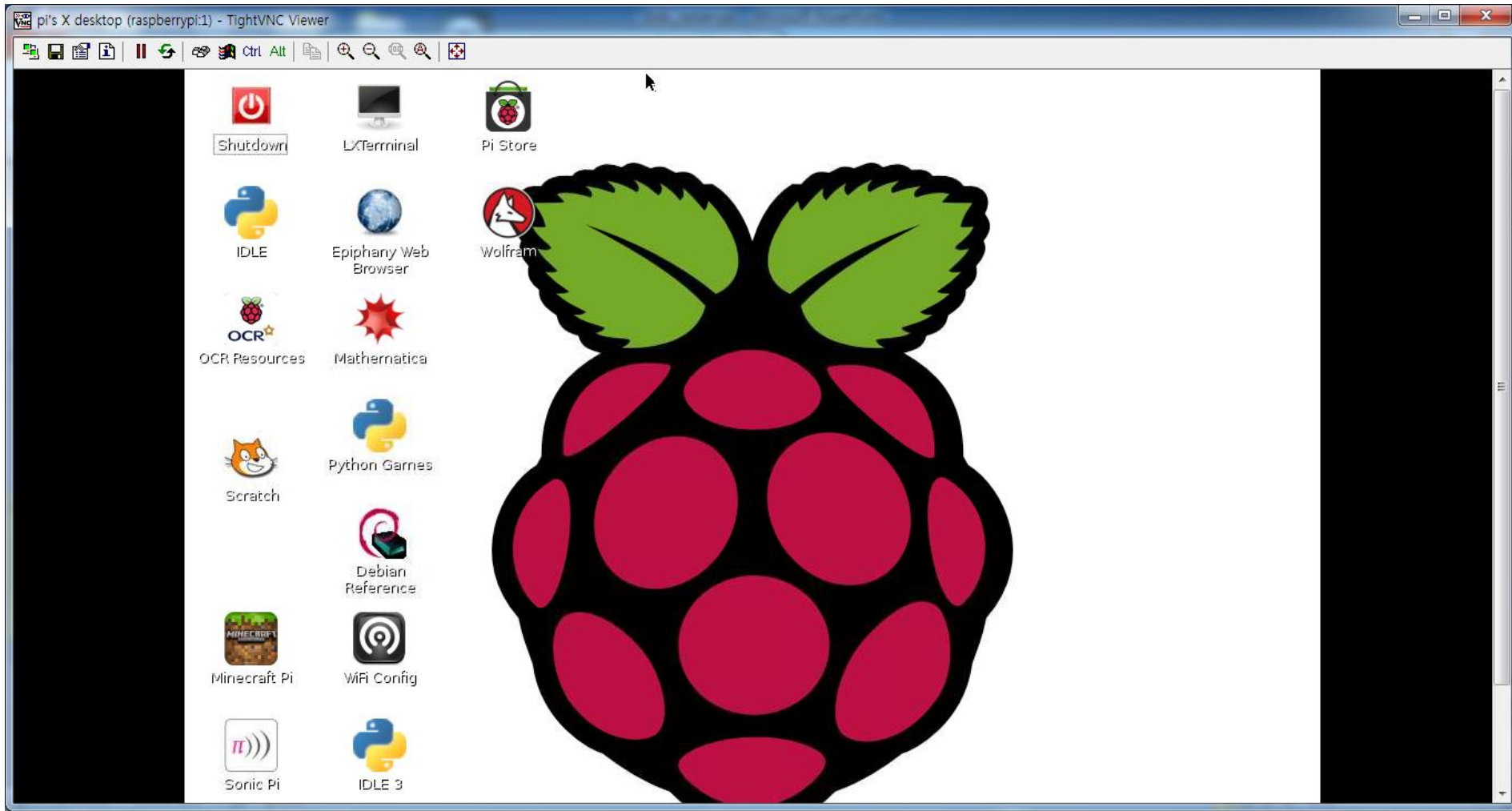


VNC on Pi

vncviewer 암호는 이전에 vncserver 를 설치할 때 입력한 `vncserver` 를 입력한다.



VNC on Pi



SSH

우선 사용자 sghong 를 추가해 본다.

```
$sudo useradd -m -d /home/sghong sghong
```

```
$sudo passwd sghong
```

```
$ssh sghong@
```

```
pi@raspberrypi:~ $ sudo useradd -m -d /home/sghong sghong
pi@raspberrypi:~ $ sudo passwd sghong
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $ █
```

```
sghong@sghong-desktop:~$ ssh sghong@172.20.12.63
The authenticity of host '172.20.12.63 (172.20.12.63)' can't be established.
ECDSA key fingerprint is dd:c7:c6:90:bc:4a:3b:6b:0b:34:bd:85:a8:bf:89:f2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.20.12.63' (ECDSA) to the list of known hosts.
sghong@172.20.12.63's password:

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: Thu Oct  6 01:51:42 2016 from 172.20.12.213
sghong@raspberrypi:~ $ █
```


SAMBA

우선 samba 관련된 패키지를 설치한다.

```
$sudo apt-get install samba samba-common-bin
```

```
pi@raspberrypi /etc/samba $ sudo apt-get install samba samba-common-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  tdb-tools
Suggested packages:
  openbsd-inetd inet-superserver smbldap-tools ldb-tools ctdb
The following NEW packages will be installed:
  samba samba-common-bin tdb-tools
0 upgraded, 3 newly installed, 0 to remove and 9 not upgraded.
Need to get 6,119 kB of archives.
After this operation, 36.1 MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

samba 에 연결할 계정을 만든다.

```
pi@raspberrypi ~ $ sudo adduser sghong
Adding user `sghong' ...
Adding new group `sghong' (1004) ...
Adding new user `sghong' (1001) with group `sghong' ...
Creating home directory `/home/sghong' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for sghong
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
```



WARNING: The following packages cannot be authenticated!

이런 메시지가 보이면 아래 명령을 실행시켜서 시스템을 최신 상태로 update 한다.

```
$sudo apt-get update
```

SAMBA (cont.)

Samba 계정을 만든다.

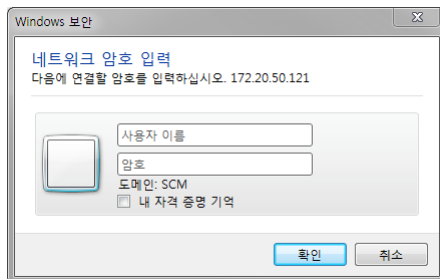
```
$sudo smbpasswd -a sghong
```

/etc/samba/smb.conf 파일에 아래 사항을 추가한다.

```
[sghong]
comment=sghong
path=/home/sghong
writeable = yes
valid users = sghong
browseable = yes
create mask = 0644
directorymasks = 0755
```

재부팅 한다.

이후, 윈도우에서 [\\172.20.20.32\sghong](#) 으로 접속한 이후, 아래와 같이 네트워크 암호를 물어보면 sghong 계정으로 Samba 를 접속할 수 있다.



Screen capture on Pi

스크린 캡처 프로그램을 설치한다.

```
$sudo apt-get install scrot
```

```
$scrot -cd 5
```

5초를 카운트 한 후의 현재 스크린을 캡처한다.

scrot 를 실행한 디렉토리에 파일이 저장된다.

Reference

<http://www.raspberrypi.org/camera-board-available-for-sale/>

<http://www.rasplay.org/>

Getting

Started with

Raspberry Pi

Matt Richardson and Shawn Wallace