
Turn Your Raspberry PI into Low-cost CCTV camera

IoT Basic

Install Raspbian

You can also install Windows 10 OS or Android OS on Raspberry Pi 4, and you may actually install and use these OS familiar to you. However, the most stable and efficient is to install and use Raspbian, a Linux-based Raspberry PI OS officially supported by the Raspberry Pi Foundation.

In the past, the process of installing Raspbian for this Raspberry Pi was quite complicated, but recently, a tool called Raspberry pi Imager is supported to make the installation work very simple and convenient. In other words, it is supported to install a tool for making an SD card for Raspberry PI OS on one's PC and to make it immediately whenever necessary.

Install Raspbian - Prerequisites

- Micro SD card to install Raspberry PI OS: 8 GB old SD card is also possible, but since it will be used for OS, it is better to use the latest one with a speed of 16 GB or more.
- Download the Raspberry pi Imager installer: Access the following site and download the installer. (In Internet Explorer, this site is not displayed normally, so you need to access it with another web browser such as Internet Explorer or Chrome)

Install Raspbian - Download

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 40-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

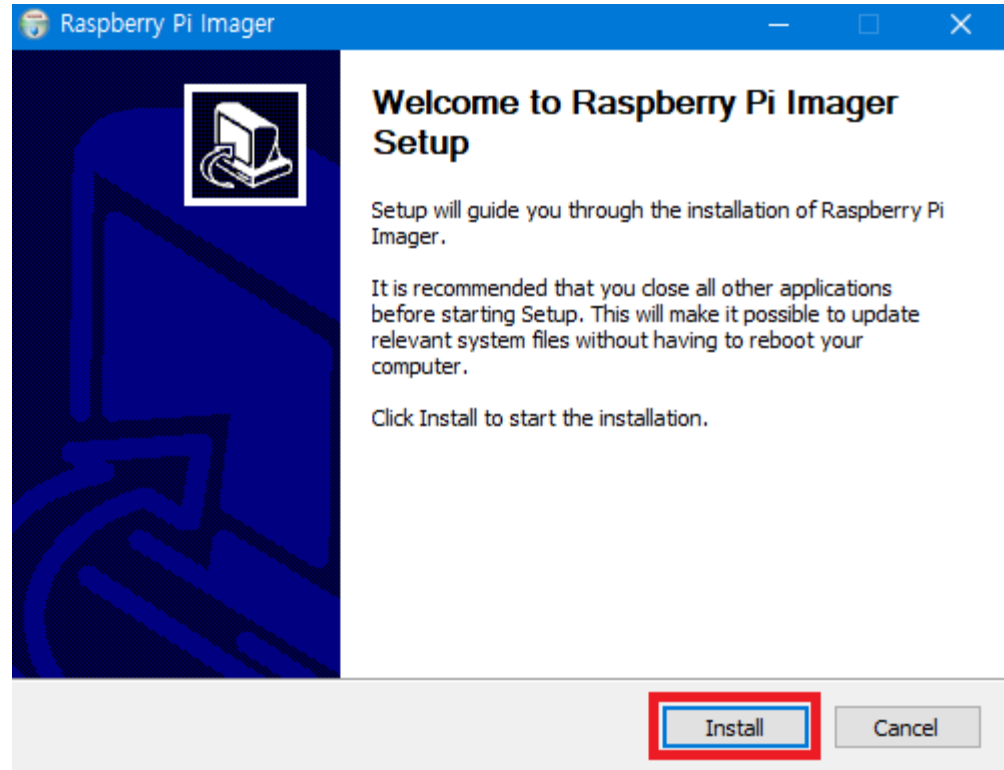
[Download for macOS](#)

[Download for Ubuntu for x86](#)



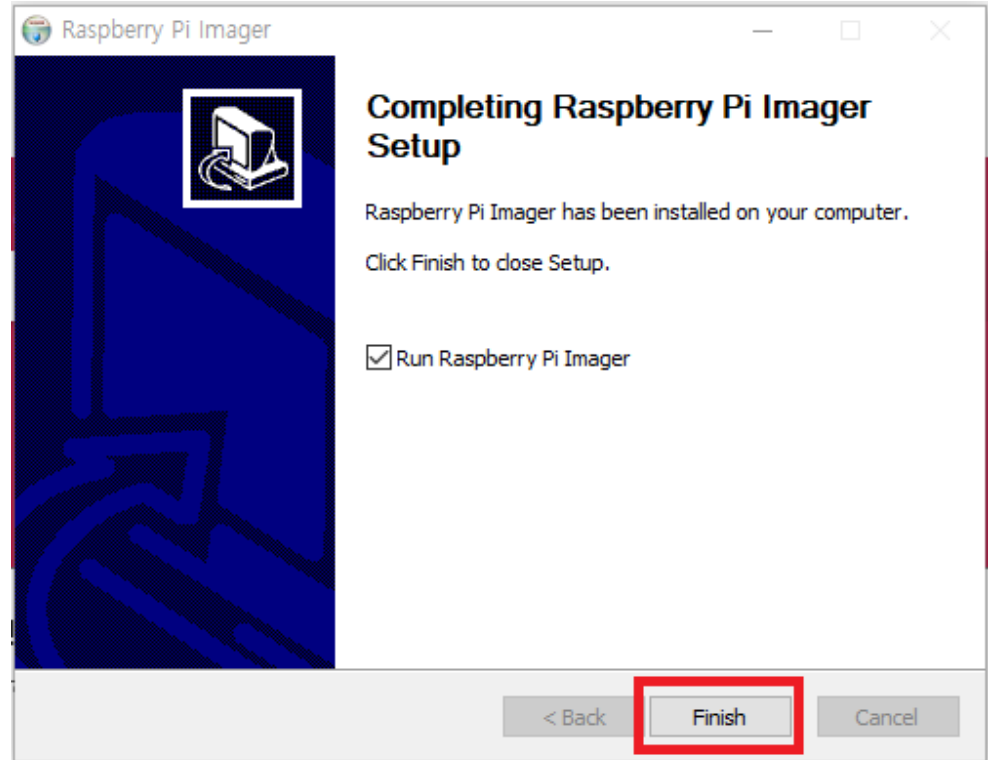
Install Raspbian - Raspberry Pi Imager

When the downloaded Raspberry pi Imager installer is started, the Raspberry pi Imager program is installed on the PC through the following process.



Install Raspbian - Raspberry Pi Imager

When the downloaded Raspberry pi Imager installer is started, the Raspberry pi Imager program is installed on the PC through the following process.



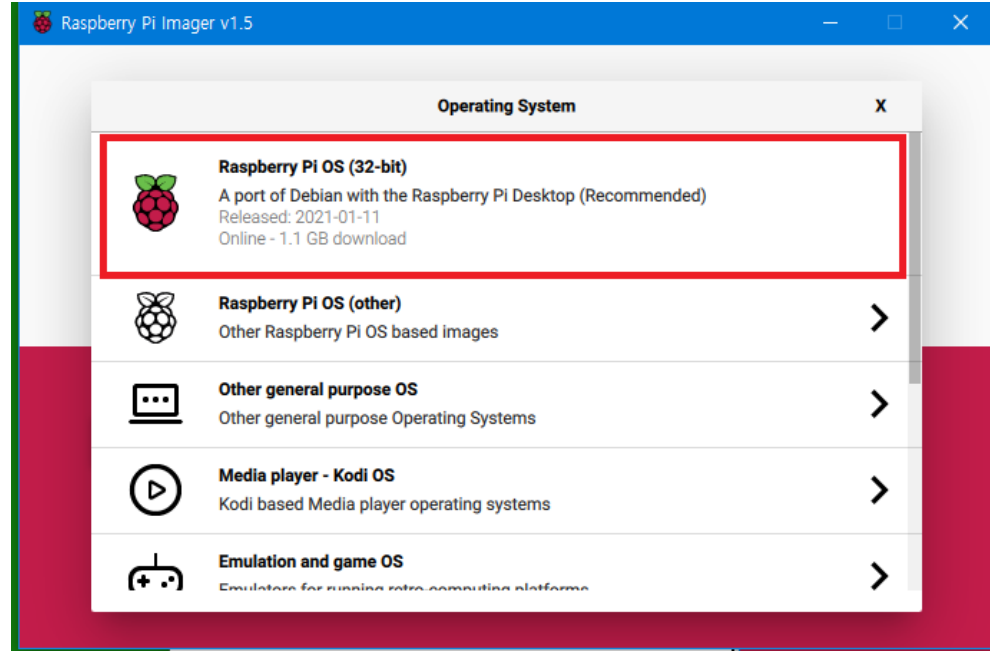
Install Raspbian - Creating a micro SD card for Raspberry PI OS

When the Raspberry pi Imager program installed on the PC is started, the following screen is displayed.



Install Raspbian - Creating a micro SD card for Raspberry PI OS

Click the CHOOSSE OS button to select the OS type to install here.



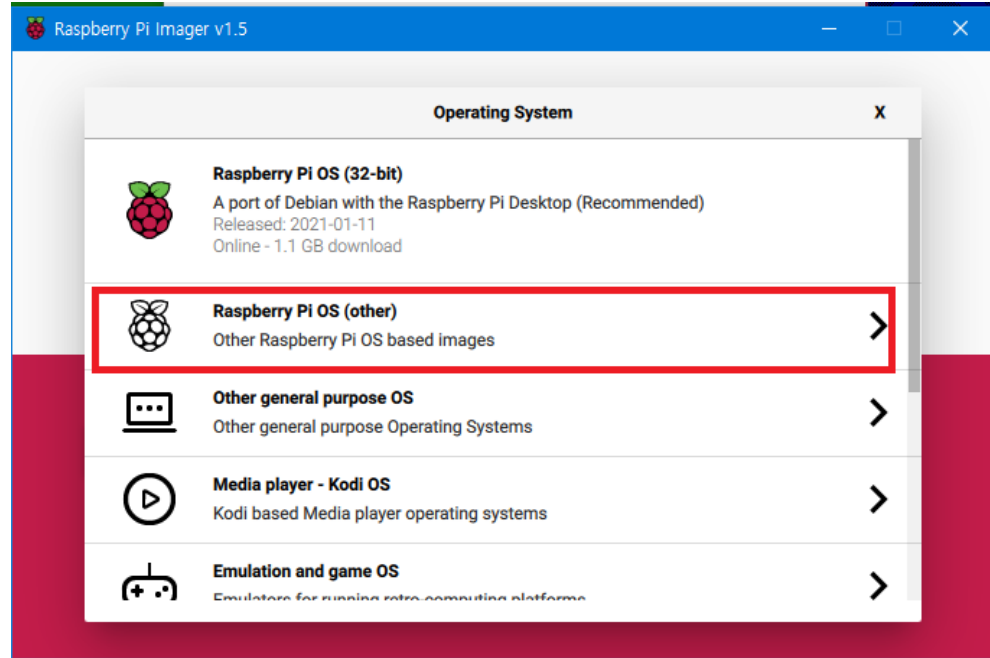
Install Raspbian - Creating a micro SD card for Raspberry Pi OS

To install in a general standard type, select Raspberry Pi OS (32bit) at the top, and the selected OS is displayed as shown below.



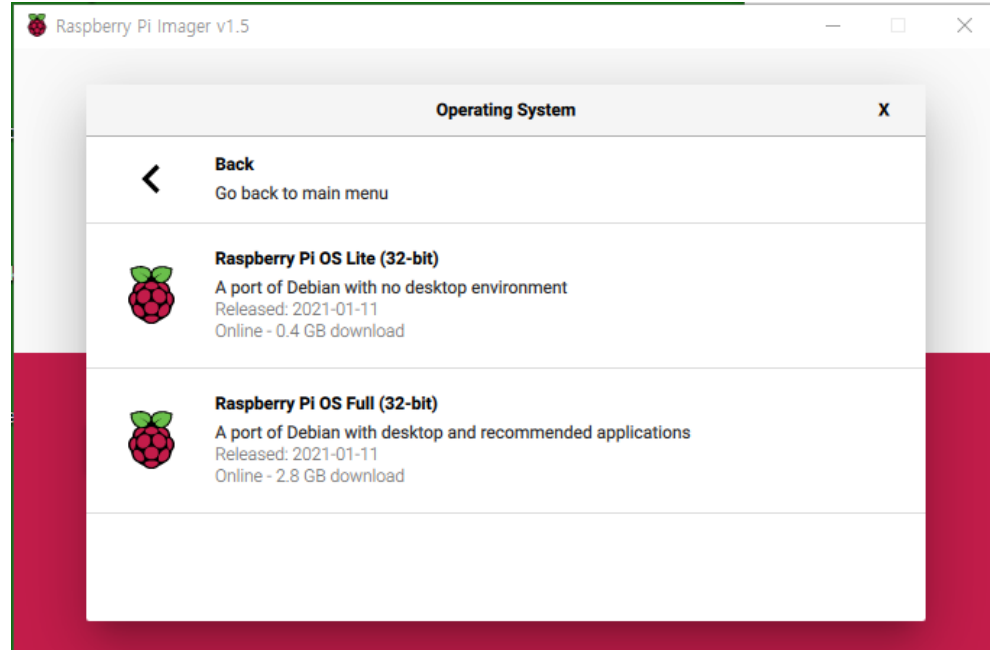
Install Raspbian - Creating a micro SD card for Raspberry PI OS

If you want to select a model other than the standard OS, select the second menu, Raspberry PI OS (other) as follows.



Install Raspbian - Creating a micro SD card for Raspberry PI OS

If you select Raspberry PI OS (other), you can choose the Lite version with reduced size or the Full version including various recommended applications in the standard type.



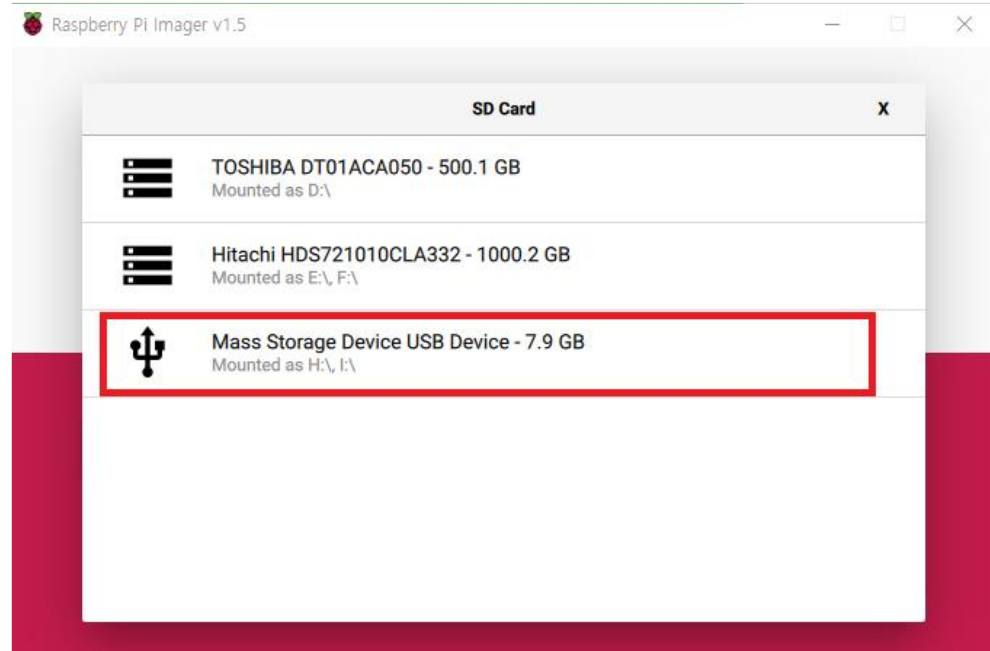
Install Raspbian - Creating a micro SD card for Raspberry Pi OS

After selecting the OS type, select the SD card drive to install.



Install Raspbian - Creating a micro SD card for Raspberry PI OS

If you click the CHOOSE SD CARD button, a list of drives is displayed on the PC as follows. Select the drive with the SD card installed among them.



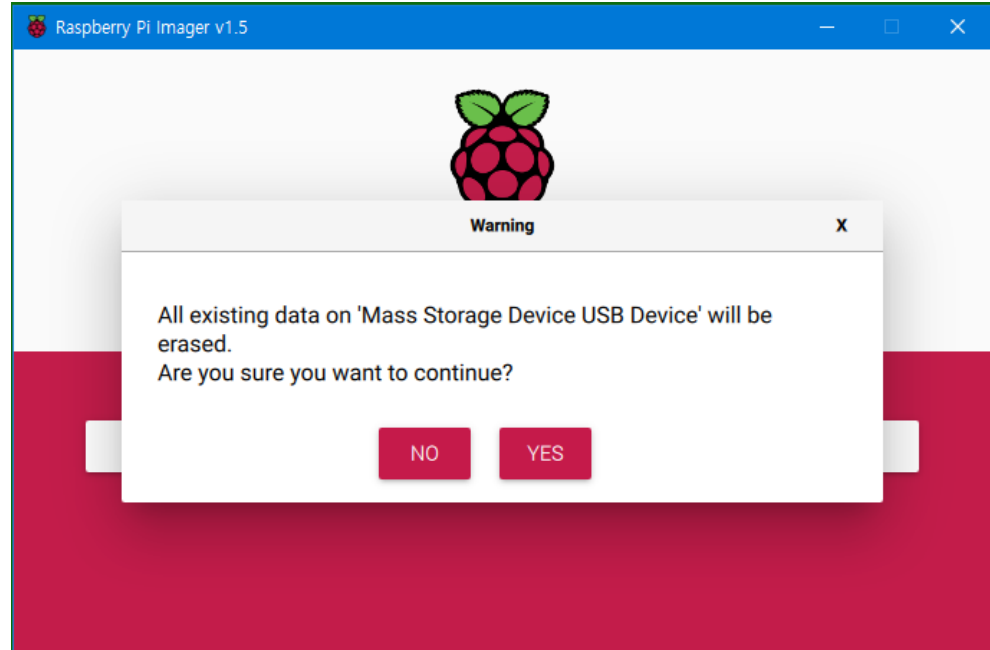
Install Raspbian - Creating a micro SD card for Raspberry Pi OS

If a drive with an SD card is selected, the name of this drive is displayed and the WRITE button is activated.



Install Raspbian - Creating a micro SD card for Raspberry PI OS

With this, the preparation for the installation work is completed and the OS installation work starts when the WRITE button is pressed.



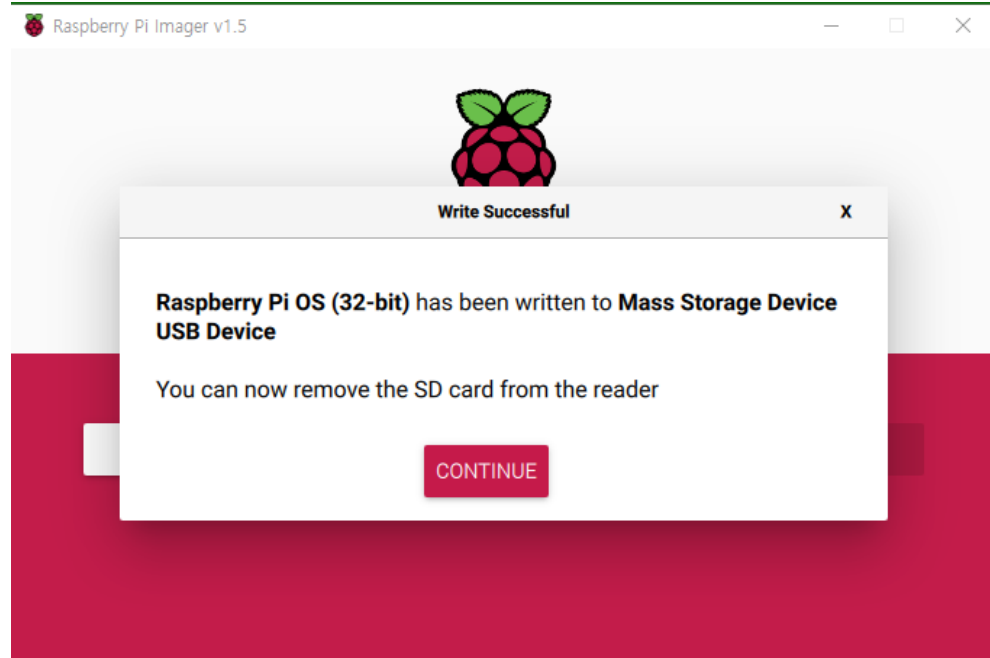
Install Raspbian - Creating a micro SD card for Raspberry Pi OS

If a warning message is displayed and confirmed as above, SD production for Raspberry Pi OS starts as follows.



Install Raspbian - Creating a micro SD card for Raspberry PI OS

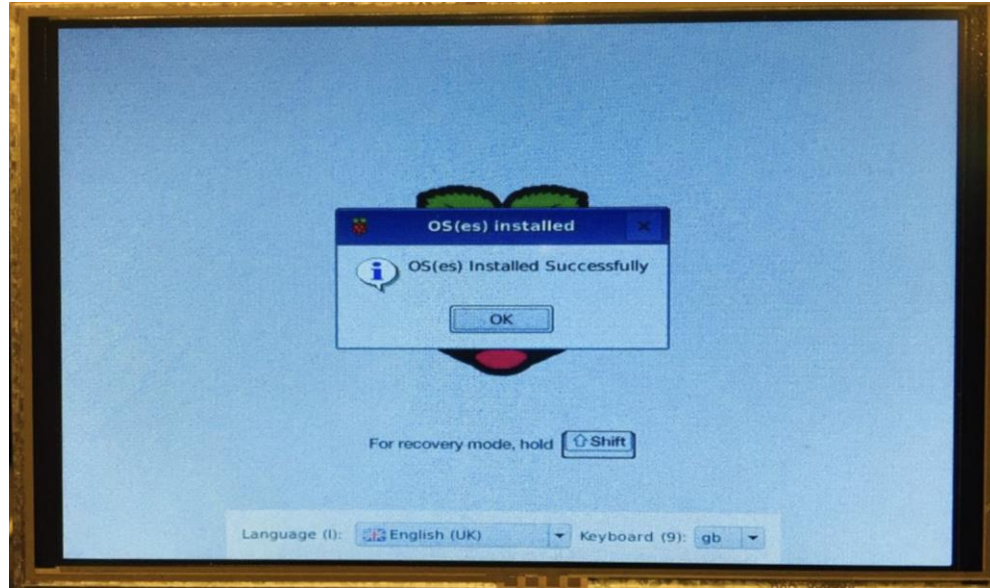
This task takes quite a bit of time, and it takes about 10 minutes depending on the writing speed of the SD card or the speed of the PC.



Install Raspbian - A few things to set up

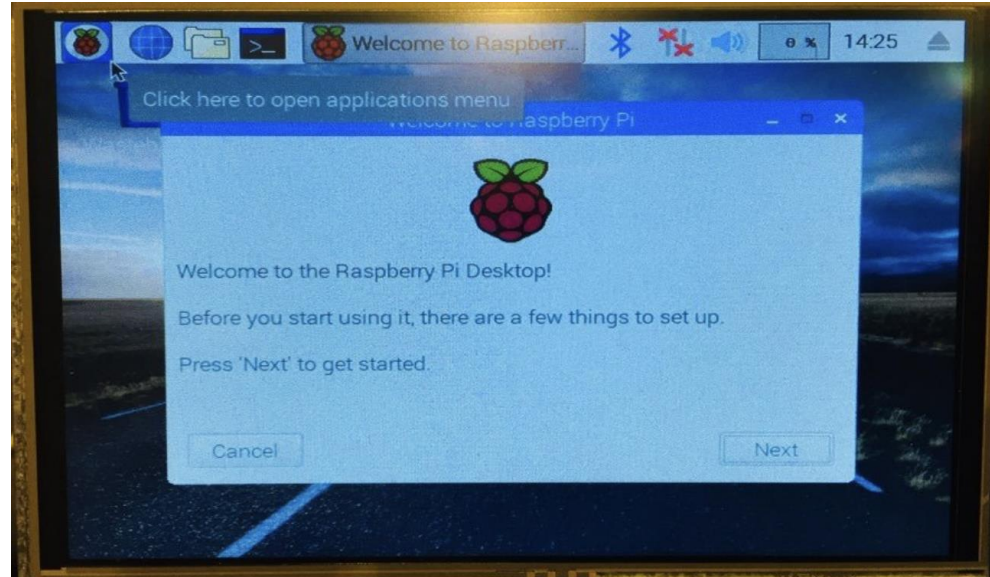
Raspberry Pi OS has been installed on the SD card, so simply inserting this SD card into the slot of the Raspberry device will start the Raspberry Pi.

When the installation process is finished, a pop-up will appear and click the OK button to complete the installation.



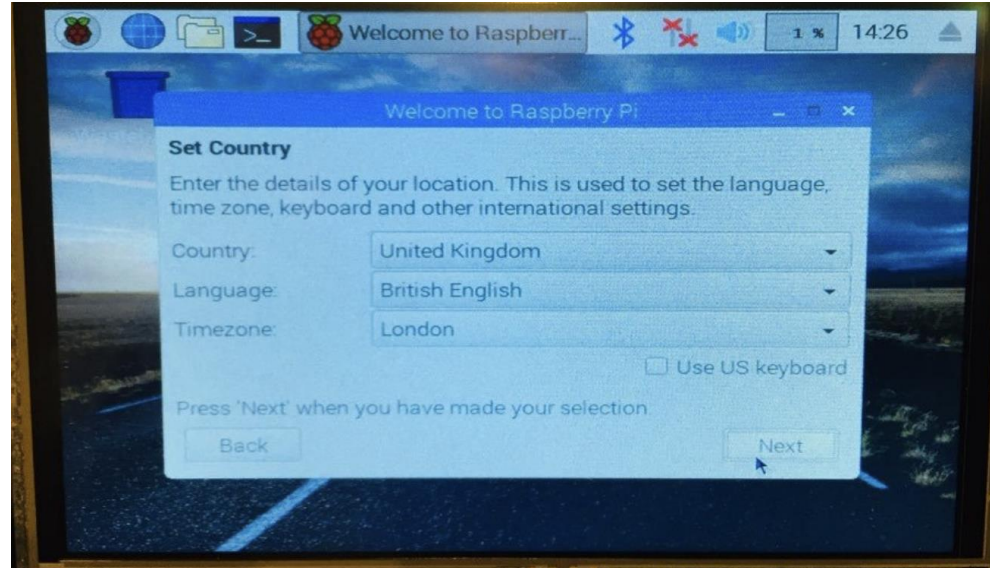
Install Raspbian - A few things to set up

After installation, a window asking if you want to perform some tasks for environment setting appears. Click the Next button to continue.



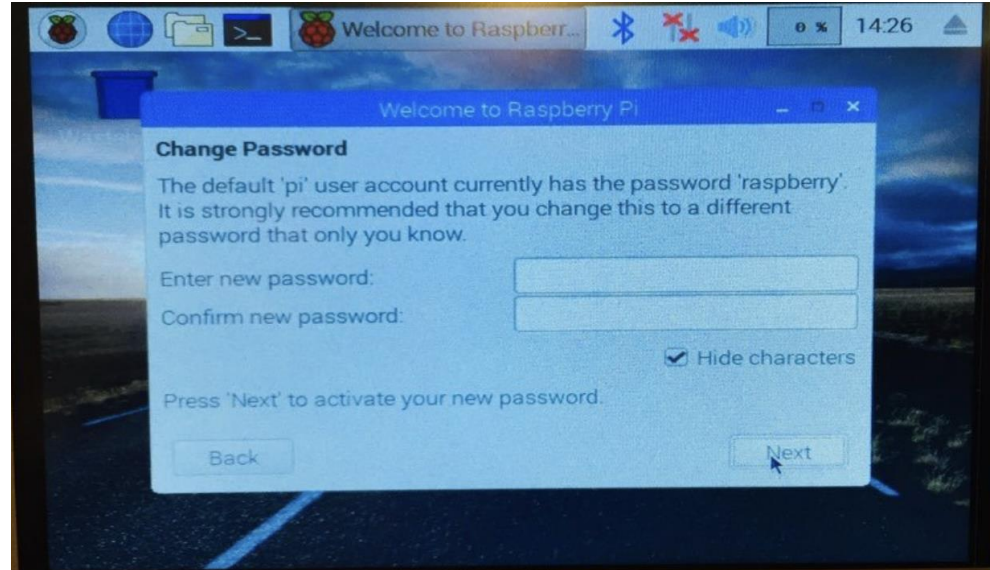
Install Raspbian - A few things to set up

You need to select a keyboard and region, but since Raspberry Pi was developed by the British Raspberry Pi Foundation, it is set to United Kingdom by default. Therefore, select [Use US Keyboard] and click the [Next] button to continue.



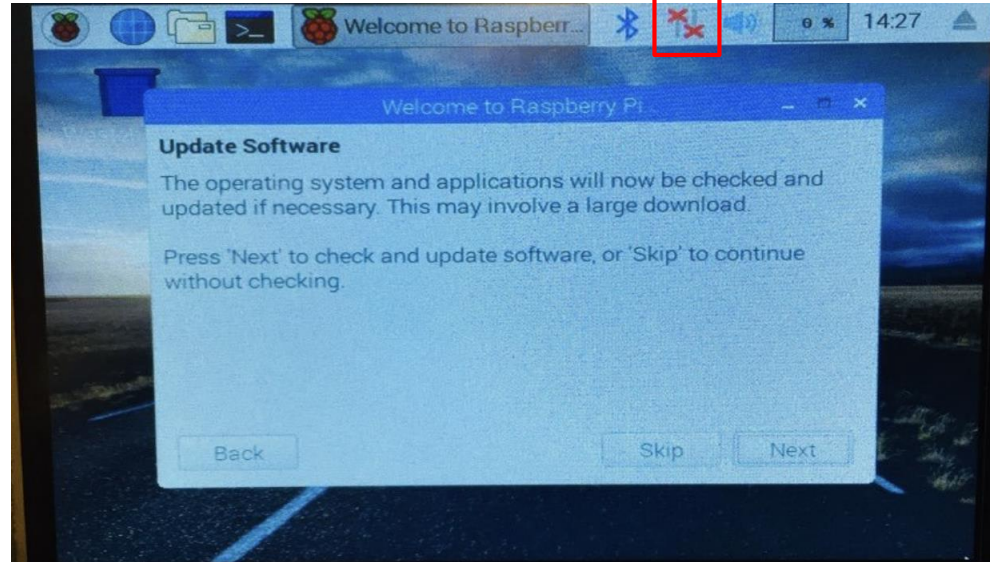
Install Raspbian - A few things to set up

The default account to access the Raspberry Pi is "pi", and the password is "raspberrypi". If you want to change your password, just enter your new password in the input window.



Install Raspbian - A few things to set up

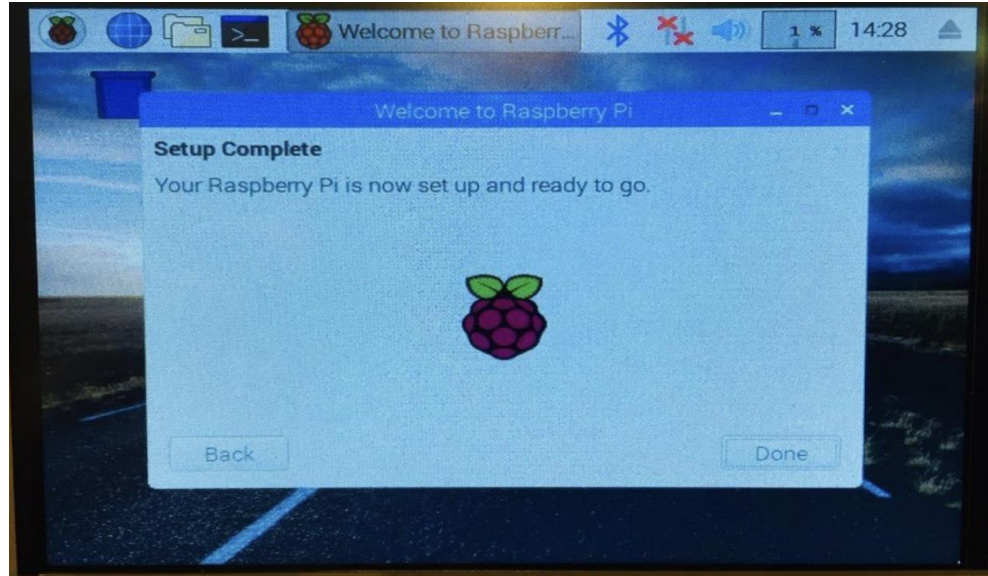
Since Raspberry Pi 3 supports wireless LAN, you can connect to an available wireless LAN by clicking the red X mark in the upper right corner.



Install Raspbian - A few things to set up

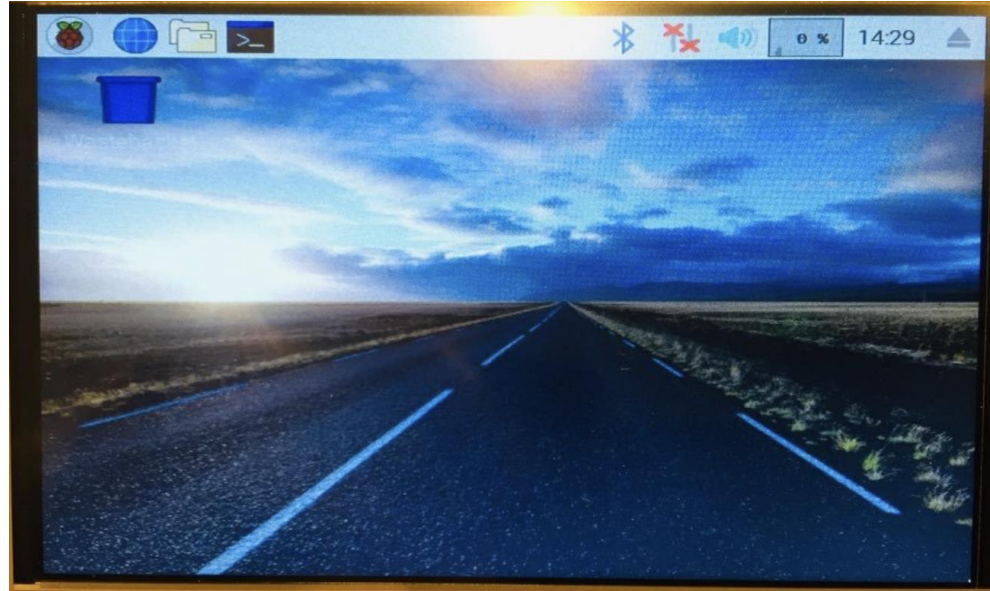
Click the [Next] button to update Raspbian and the installed software while connected to the wireless LAN.

However, it is recommended that you proceed when you have time to spare, as it may take more than an hour to update the precautions.



Install Raspbian - A few things to set up

After installing the Raspbian operating system and completing the software update, you can see the following initial screen.



Introduction of Raspberry Pi Camera Module V2

There are many modules available for Raspberry Pi, but the camera module is one of the most popular items when thinking about a Raspberry Pi project. There are many projects that use a camera to shoot images or videos.

The camera module V2 (Raspberry Pi Camera Module V2) officially released by the Raspberry Pi Foundation was released in May 2016 along with the infrared camera module V2. Both the Camera Module V2 and the Infrared Camera Module V2 have 8MP resolution, and the infrared camera module can take pictures even in low-light conditions (night photography).

Returning to the camera module V2 and looking at the specifications, the V1 5MP resolution was upgraded to the V2 8MP resolution, and the image sensor was replaced with Sony's IMX219 8MP sensor.

Based on the Raspberry Pi Foundation description, image quality, color fidelity, and performance in low light have been upgraded. In addition, VideoCore IV is said to be able to obtain good quality images with the built-in ISP and Auto White Balancing function.

And, Camera Module V2 and Infrared Camera Module V2 are fully compatible with all Raspberry Pi series (ZERO, A, B - 1, 2, 3).

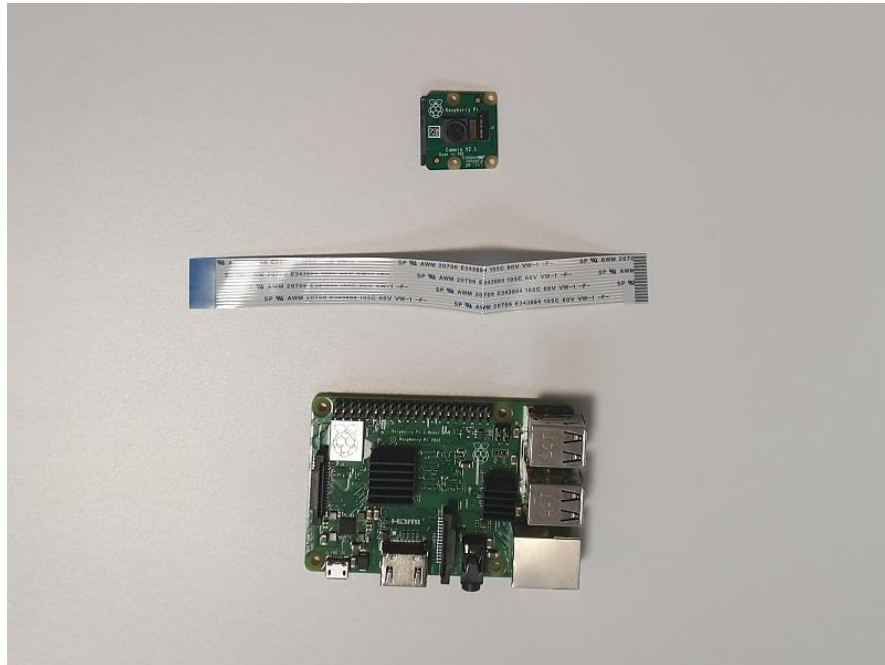
Introduction of Raspberry Pi Camera Module V2

Raspberry Pi Camera Module V2 Main Specifications

Product Name	Raspberry Pi Camera Module
Image Sensor	Sony IMX 219 PQ CMOS image sensor in a fixed-focus module.
Resolution	8-megapixel
Still picture resolution	3280 x 2464
Max image transfer rate	1080p: 30fps (encode and decode) 720p: 60fps
Connection to Raspberry Pi	15-pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2).
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -20° to 60° Stable image: -20° to 60°
Lens size	1/4"
Dimensions	23.86 x 25 x 9mm
Weight	3g
국내가격	28,000~33,000원

Raspberry Pi camera installation

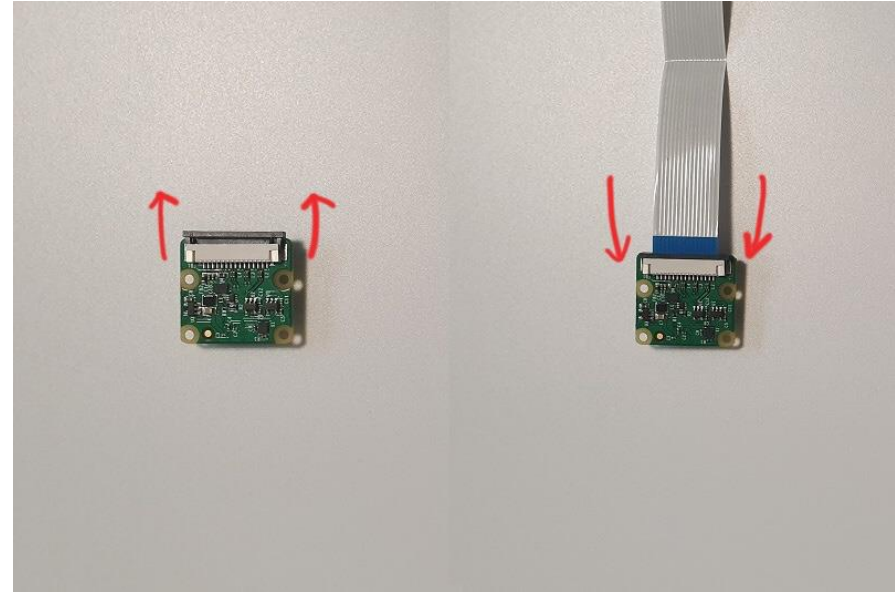
This is a preparation for using the Camera Module V2 officially released by the Raspberry Pi Foundation. Of course, you must have a camera module. You need a Raspberry Pi to operate the camera module. Camera Module V2 can be run on any Raspberry Pi, so any Raspberry Pi is fine. I prepared a Raspberry Pi 3 B+. And, you need a connector cable to connect the Raspberry Pi body and the camera module. A connector cable provides a CSI interface. The specifications of the cable to connect to Raspberry Pi 3 B+ and the cable to connect to Raspberry Pi Zero are different. After deciding which Raspberry Pi to connect to, check the cable specifications as well.



Raspberry Pi camera installation

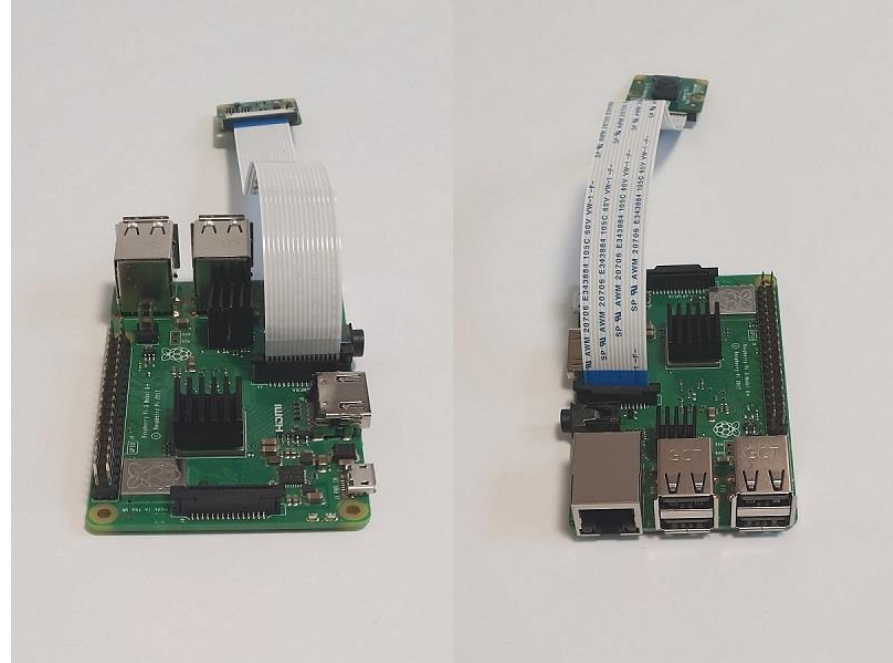
First, connect the cable to the camera module as shown in the picture below. Here, we tested with Raspberry Pi ZERO. There is a clip (?) that serves to hold the cable in the cable insertion part of the camera module. Pull out the clip before inserting the cable, then insert the cable and reinsert the clip to secure the cable.

If you are confused about the cable insertion direction, it is convenient to classify the direction when inserting the cable into the camera or Raspberry Pi as follows. At the end of the cable, one side has a metal side that can carry electrical signals and the other side doesn't. The Raspberry Pi connector side also has an electrical circuit that can transmit electrical signals. Just align and insert the cable and the connector side so that the electrical signal can be transmitted.



Raspberry Pi camera installation

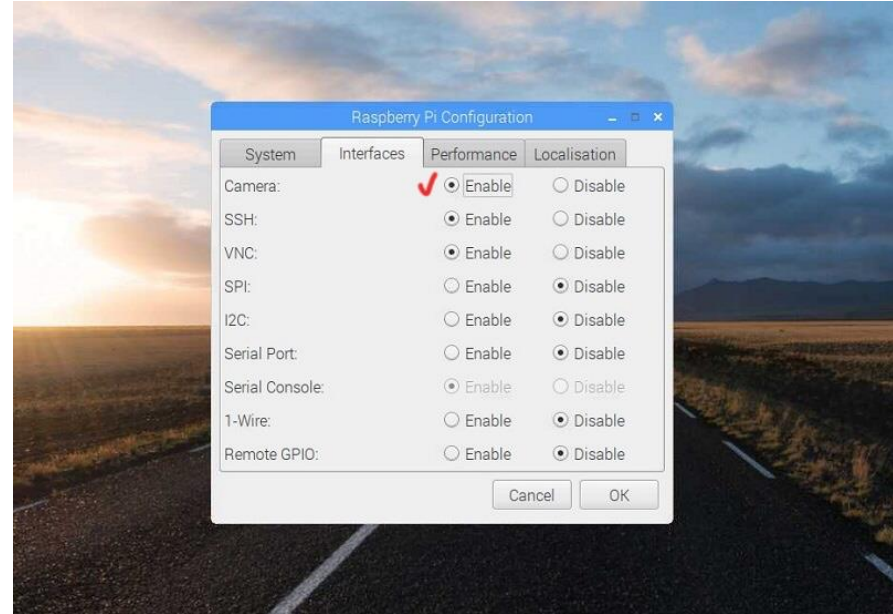
The same fixing clip exists on the connector side of the Raspberry Pi. You can connect the cables in the same way as when fixing the camera module and cable.



Camera module interface setting

Next, boot the Raspberry Pi and configure the settings for using the camera module. Select Preference->Raspberry Pi Configuration->Interfaces from the menu.

Select OK after setting Camera to Enable in Interfaces. Then, a reboot guide appears and after rebooting, Camera Interface is now set.



Camera module interface setting

To check if the camera operation is normal, take a picture with the raspistill command as shown below.

```
#!/bin/bash  
pi@raspberrypi:~ $ raspistill -o image.jpg
```

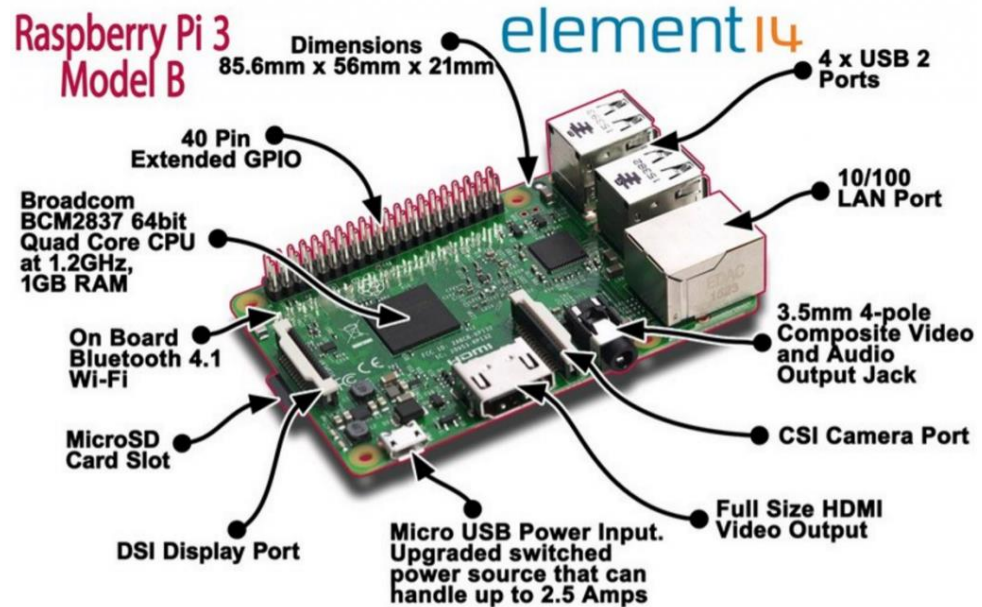
If you check the image.jpg file in that path, you can check the photo file that was taken.

Low-cost CCTV Camera - Hands-on Lab

Under this lab, I will showcase how to install the latest Docker 18.09.0 on Raspberry Pi. I will test drive BuildKit on top of this credit-card sized device and finally showcase how to turn this small Pi into low-cost HD surveillance camera using a single Docker Image.

Low-cost CCTV Camera - Hands-on Lab(Prerequisites)

1. Raspberry Pi 4
2. Micro-SD card reader
3. Any Windows/Linux/macOS
4. HDMI cable
5. Internet
Connectivity(WiFi/Broadband/
Tethering using Mobile) – to
download Docker 18.09.0
package
6. Keyboard & mouse connected
to Pi's USB ports



Low-cost CCTV Camera - Hands-on Lab

Verifying Raspbian OS Version

```
root@raspberrypi:~# cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
root@raspberrypi:~# curl -sSL https://get.docker.com/ | sh
# Executing docker install script, commit: 40b1b76
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/raspbian/gpg" | apt-key add -qq - >/dev/null
Warning: apt-key output should not be parsed (stdout is not a terminal)
+ sh -c echo "deb [arch=armhf] https://download.docker.com/linux/raspbian stretch edge" >
/etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sh -c docker version
Client:
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:57:21 2018
 OS/Arch:           linux/arm
 Experimental:      false
```

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
Server: Docker Engine - Community
```

```
Engine:
```

```
Version:      18.09.0
```

```
API version:  1.39 (minimum version 1.12)
```

```
Go version:   go1.10.4
```

```
Git commit:   4d60db4
```

```
Built:        Wed Nov  7 00:17:57 2018
```

```
OS/Arch:      linux/arm
```

```
Experimental: false
```

If you would like to use Docker as a non-root user, you should now consider adding your user to the "docker" group with something like:

```
sudo usermod -aG docker your-user
```

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run containers which can be used to obtain root privileges on the docker host.

Refer to <https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface> for more information.

Low-cost CCTV Camera - Hands-on Lab

Installing Docker 18.09

```
** DOCKER ENGINE - ENTERPRISE **
```

If you're ready for production workloads, Docker Engine - Enterprise also includes:

- * SLA-backed technical support
- * Extended lifecycle maintenance policy for patches and hotfixes
- * Access to certified ecosystem content

```
** Learn more at https://dockr.ly/engine2 **
```

ACTIVATE your own engine to Docker Engine - Enterprise using:

```
sudo docker engine activate
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Docker Version

```
root@raspberrypi:~# docker version
Client:
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:57:21 2018
 OS/Arch:           linux/arm
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:           18.09.0
  API version:       1.39 (minimum version 1.12)
  Go version:        go1.10.4
  Git commit:        4d60db4
  Built:             Wed Nov  7 00:17:57 2018
  OS/Arch:           linux/arm
  Experimental:      false
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

Test Drive Nginx App on Pi Box

```
root@raspberrypi:~# docker run -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
9c38b5a8a4d5: Pull complete
1c9b1b3e1e0d: Pull complete
258951b5612f: Pull complete
Digest:
sha256:dd2d0ac3fff2f007d99e033b64854be0941e19a2ad51f174d9240dda20d9f534
Status: Downloaded newer image for nginx:latest
d812bf50d136b0f78353f0a0c763b6b08ecc5e7ce706bac8bd660cdd723e0fcd
root@raspberrypi:~#
```

Low-cost CCTV Camera - Hands-on Lab

Test Drive Nginx App on Pi Box

```
root@raspberrypi:~# curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```


Low-cost CCTV Camera - Hands-on Lab

BuildKit on Raspberry Pi

```
root@raspberrypi:~# export DOCKER_BUILDKIT=1
root@raspberrypi:~# git clone https://github.com/ajeetraina/hellowhale
Cloning into 'hellowhale'...
remote: Enumerating objects: 28, done.
remote: Total 28 (delta 0), reused 0 (delta 0), pack-reused 28
Unpacking objects: 100% (28/28), done.
root@raspberrypi:~# cd hellowhale/
root@raspberrypi:~/hellowhale# ls
Dockerfile  html  README.md  wrapper.sh
root@raspberrypi:~/hellowhale# docker build -t ajeetraina/hellowhalecom .
[+] Building 7.9s (5/8)
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 129B                                              0.0s
=> [internal] load .dockerignore                                                  0.2s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/nginx:latest                  0.0s
=> [1/3] FROM docker.io/library/nginx:latest                                    0.0s
=> => resolve docker.io/library/nginx:latest                                    0.0s
=> [internal] helper image for file operations                                    0.1s
=> => resolve docker.io/docker/dockerfile-copy:v0.1.9@sha256:e8f159d3f00        7.5s
=> => sha256:b13ecc473b58ad8d80fba73ae6de690f6fcbe341bdaca42 736B / 736B    0.0s
=> => sha256:fabe16b757ee155dfd7210795199962d1b35e22b3437d06 767B / 767B    0.0s
```

Low-cost CCTV Camera - Hands-on Lab

BuildKit on Raspberry Pi

```
root@raspberrypi:~/helloworld# time docker build -t ajeetraina/helloworld .
[+] Building 0.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 31B                                                0.0s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/nginx:latest                  0.0s
=> [internal] helper image for file operations                                  0.0s
=> [1/3] FROM docker.io/library/nginx:latest                                    0.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 317B                                                  0.0s
=> CACHED [2/3] COPY wrapper.sh /                                                0.0s
=> CACHED [3/3] COPY html /usr/share/nginx/html                                  0.0s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.0s
=> => writing image sha256:5aee990f7e24e7c0f486ed01b4c1f8696ff307f836af1      0.0s
=> => naming to docker.io/ajeetraina/helloworld                                0.0s
real                                0m0.615s
user                                0m0.204s
sys                                  0m0.082s
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Dockerd

```
root@raspberrypi:~/hellowhale# systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
  Active: active (running) since Tue 2019-02-26 13:01:04 IST; 38min ago
    Docs: https://docs.docker.com
  Main PID: 2437 (dockerd)
    CPU: 1min 46.174s
  CGroup: /system.slice/docker.service
          └─2437 /usr/bin/dockerd -H unix://
          └─2705 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8
          └─4186 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8
```

Low-cost CCTV Camera - Hands-on Lab

Verifying if armv7 hello-world image is available or not

```
docker run --rm mplatform/mquery hello-world
Unable to find image 'mplatform/mquery:latest' locally
latest: Pulling from mplatform/mquery
db6020507de3: Pull complete
5107afd39b7f: Pull complete
Digest: sha256:e15189e3d6fbcee8a6ad2ef04c1ec80420ab0fdcf0d70408c0e914af80dfb107
Status: Downloaded newer image for mplatform/mquery:latest
Image: hello-world
* Manifest List: Yes
* Supported platforms:
- linux/amd64
- linux/arm/v5
- linux/arm/v7
- linux/arm64
- linux/386
- linux/ppc64le
- linux/s390x
- windows/amd64:10.0.14393.2551
- windows/amd64:10.0.16299.846
- windows/amd64:10.0.17134.469
- windows/amd64:10.0.17763.194
```

Low-cost CCTV Camera - Hands-on Lab

Verifying hellowhale Image

```
root@raspberrypi:~# docker run --rm mplatform/mquery ajeetraina/hellowhale
Image: ajeetraina/hellowhale
* Manifest List: No
* Supports: amd64/linux
```

Low-cost CCTV Camera - Hands-on Lab

Verifying Prometheus Image

```
root@raspberrypi:~# docker run --rm mplatform/mquery rycus86/prometheus
Image: rycus86/prometheus
* Manifest List: Yes
* Supported platforms:
  - linux/amd64
  - linux/arm/v7
  - linux/arm64
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Cloning the Repository:

```
$ git clone https://github.com/collabnix/docker-cctv-raspbian  
$ cd docker-cctv-raspbian
```

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Building the Docker Image:

```
docker build -t collabnix/docker-cctv-raspbian .
```


Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Configuring Camera Interface:

Before you execute `run.sh`, you need to configure Camera Interface by running the below command:

```
# raspi-config
```

It will open up command-line UI window, choose Interfacing , select Camera and enable it. Save and exit the CLI window.

Low-cost CCTV Camera - Hands-on Lab

Running Low-cost HD surveillance Camera system using Docker Container -

Running the Docker container:

```
root@raspberrypi:~/rpi-motion# sudo modprobe bcm2835-v4l2  
root@raspberrypi:~/rpi-motion# ls
```

```
$sh run.sh
```

Low-cost CCTV Camera - Hands-on Lab

That's it. Browse over to <http://192.168.1.5:8082>(either using Win Laptop or macbook) to open up CCTV cam which initiates the video streaming instantly. Cool, isn't it?

