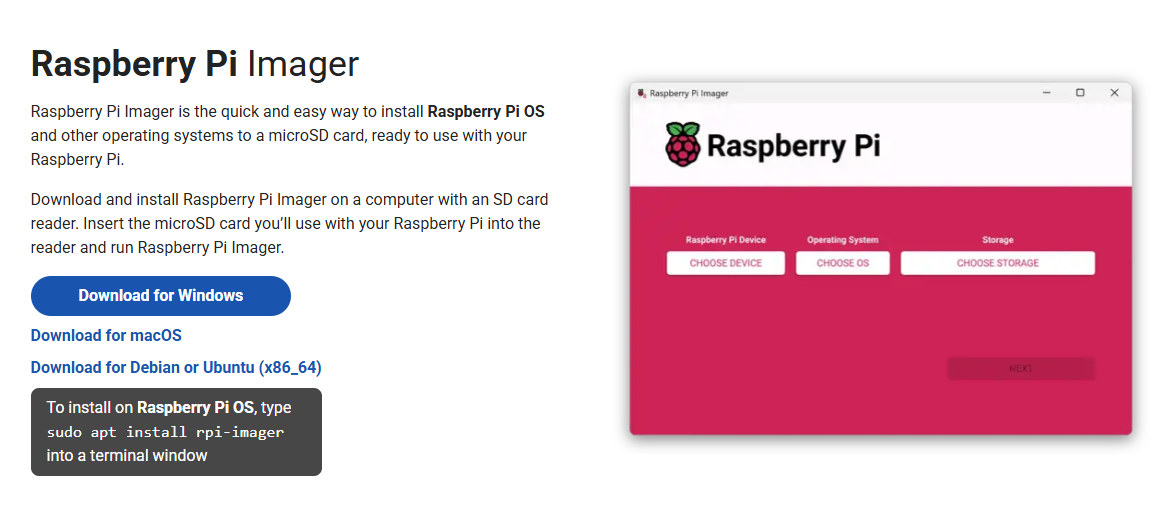
1. **Raspberry pi setup**
2. Install and launch Raspberry pi imager from web site to your computer

(<https://www.raspberrypi.com/software/>)



1. Download Legacy OS (<https://downloads.raspberrypi.com/raspios_oldstable_armhf/images/raspios_oldstable_armhf-2024-03-12/>) and write Raspberry pi OS (Legacy) to SD card
2. Install SD card to Raspberry pi board
3. Connect keyboard, mouse and display to Raspberry pi board
4. Turn Raspberry pi ON
5. Setup nationality, language, user name, password and wifi connection
6. Shutdown and remove mouse, keyboard and display
7. Turn Raspberry pi ON, When desktop is available, Connect your computer to Raspberry pi remotely using via VNC viewer (Need to enter IP address, user name and password in advance) ([Download VNC Viewer by RealVNC®](https://www.realvnc.com/en/connect/download/viewer/?lai_sr=5-9&lai_sl=l))
8. To stabilize network connection, Parameter is change in network preference

Configure : wlan0

□Automatically configure empty option

☑ Disable IPv6

IPv4 Address : Raspberry pi IP address

Please fill the Router, DNS servers and DNS search items.

Check list

□Can you connect to Raspberry pi using VNC?

1. **Supervisor application**
2. In linux terminal, enter “ sudo apt-get install supervisor “
3. In linux terminal, enter “ cd /etc/supervisor “
4. In linux terminal, enter “ ls “
5. Update permissions for all files listed using “ sudo chmod 777 supervisord.conf “
6. Update permissions for all files listed using “ sudo chmod -R 777 conf.d “
7. Open file manager and enter “ /etc/supervisor “ at directory
8. Rewrite the contents of supervisord.conf to the same context as attached folder and save it. supervisord.conf file directory : " https://github.com/mrobert7/MSB\_imaging/blob/main/Supervisor/supervisord.conf "
9. Create a file named “ CU.conf ” in the conf.d folder, paste the contents of the attached CU.conf and save it. (Use the username set on the Raspberry pi.)

CU.conf file directory : " https://github.com/mrobert7/MSB\_imaging/blob/main/Supervisor/conf.d/CU.conf "

1. Create a folder in home/pi with the name Experiment and create files in the folder with the names “ CU.py ” and “ CU.log ”.
2. In linux terminal, enter “ pip3 install ischedule “
3. Reboot
4. Check to see if any errors are identified when executing “ sudo supervisorctl status CU:CU.py ” in linux terminal.

Check list

□Is “CU.py” executable in Supervisor?

1. **Setup Camera module**
2. In linux terminal, enter “ sudo geany “
3. Open boot/config.txt file by geany and write “ dtoverlay=arducam-64mp “ at the bottom of the script in the config.txt file (16mp→imx519)

Please check→ <https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/Quick-Start-Guide/#cm3cm4>

1. In linux terminal, enter “ wget -O install\_pivariety\_pkgs.sh https://github.com/ArduCAM/Arducam-Pivariety-V4L2-Driver/releases/download/install\_script/install\_pivariety\_pkgs.sh “
2. In linux terminal, enter “ chmod +x install\_pivariety\_pkgs.sh “
3. In linux terminal, enter “ ./install\_pivariety\_pkgs.sh -p libcamera\_dev “
4. In linux terminal, enter “ ./install\_pivariety\_pkgs.sh -p libcamera\_apps “
5. Reboot
6. To check camera operation, in terminal, enter “ libcamera-hello “

(Please check whether camera is accurately connected to pi computer)

Check list

□Is the camera working?

1. **Ready for experiment script**
2. Make a “ slotimaging “ folder in the new USB .
3. Make a “ Data “ folder in the “ slotimaging “ folder .
4. Create a “ well1 “ “ well2 ““ well3 ““ well4 ““ well5 ““ well6 ““ well7 ““ well8 ““ well9 “folder in the “ Data “ folder in the USB.
5. Create a “ Data1 “ “ Data2 ““ Data3 ““ Data4 “folder in the each well folder.
6. Download all sh files to Experiment folder

sh file directory : https://github.com/mrobert7/MSB\_imaging/tree/main/Experimental\_script\_imaging\_operation "

※Rewrite the contents of sh file to use as an imaging parameter script.

Example (White trans image)

libcamera-still

--width 2312 # Width pixel of captured image

--height 1736 # height pixel of captured image

--datetime -n -o /media/raspberry4/Raspi\_USB/slotimaging/well1/Data1 # Directory to destination folder

--contrast 1.0 # Image contrast

--brightness 0.1 # Image brightness

--shutter 500 # Shutter speed (unit is µs)

--autofocus-mode manual # Manual focus

--lens-position 13 # Focus adjustment

--awbgains 1.5,2.0　#RGB

--lens position 14

1. Rewrite contents of “ CU.py “ to the same context as attached folder and save it

※Rewrite the contents of CU.py to use as an experimental script

※Refer to the comment out of the attached CU.py when changing the amount of rotation of the stepping motor and controlling the lighting of the relay module.

CU.py file directory : https://github.com/mrobert7/MSB\_imaging/blob/main/Experimental\_script\_imaging\_operation/CU.py

1. When you start taking pictures over time, enter “ sudo supervisorctl start CU:CU.py ” in the linux terminal.
2. When you stop taking pictures over time, enter “ sudo supervisorctl stop CU:CU.py ” in the linux terminal.

**List of things to check before starting imaging**

□Can you connect and search the net server from Raspberry pi?

□Have you set up your Raspberry pi (contents of this paper)?

□Is the camera working?

□Are the images taken by the camera saved in the specified folder?

□Is the camera in focus?

□Is “CU.py” executable in Supervisor?

□Does the LED array board used light up?

□Are the LEDs controlled to turn on and off via relay modules?

□Is the LED light source and optical filter exchangeable via stepper motors?

□Is the position of the sample for photography controlled correctly by the stepping motor?

□Is the table with the shooting sample in the initial position?

□Is the power supply for the stepping motor and relay module connected (5V)?

□Is the external power supply for the LEDs connected (18V)?