Operation/Assembly Manual

1. Assembly/Disassembly

All parts to be printed exist in CAD

1. Frame

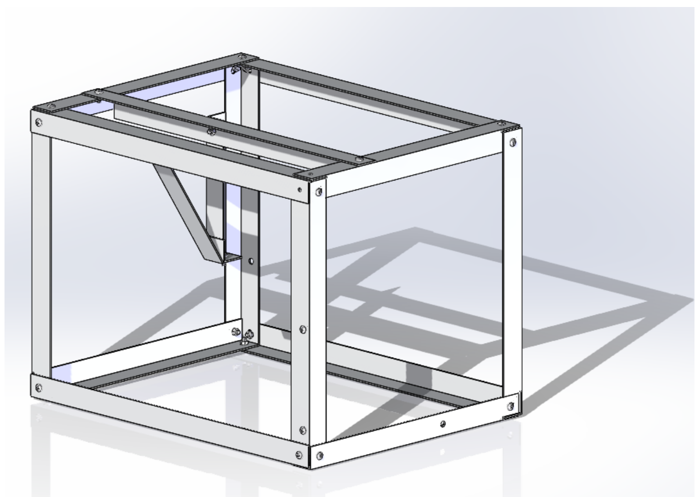


Figure 1：Frame

The first step needs to be putting the frame together, as all the parts will be mounted on and protected by the frame. At this stage, an L-shaped aluminum bar is used. After cutting the aluminum bar into 12in and 16in lengths, a drill is used to drill holes in the 8 joints of the rectangular frame and fix them with screws. A rectangular frame of 12in*12in*16in is formed. The horizontal aluminum bar at the top is also L-shaped and can be fit and mounted at the top as a beam by cutting off the sides. There is a vertical aluminum bar screwed to the beam that is intended for the camera. The camera will be fixed to the bottom of this aluminum bar. There is also an aluminum rod that links the camera attachment to the aluminum frame, this is for the stability of the camera.

2. Vacuum and Channel

Figure 2: Vacuum and Channel

The vacuum was purchased and the channels as well as the connectors were 3D printed. After the channel is printed, the channel is coated with carbon 3D conductive paint. Use a glass cutter to cut the ITO glass to a size similar to the shooting window reserved on the channel and glue the ITO glass to the channel. This way the entire channel is electrically conductive. At the bottom of the vacuum is a clamp that links the ball joint, which is fixed to the aluminum frame. This allows the channel angle to be adjustable.

3. Raspberry Pi and Relay

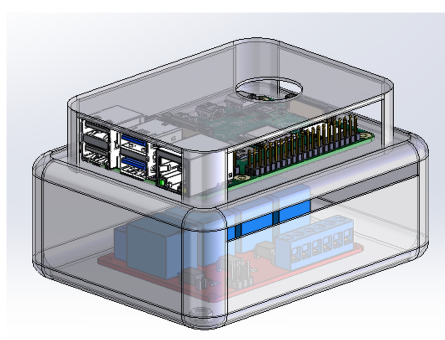


Figure 3: Raspberry Pi and Relay

The Raspberry Pi is placed in the original box while a small plastic box is then used to place the relay. The protective case of the Raspberry Pi has screw holes in all four corners. After placing the Raspberry Pi on the protective case of the relay, poke the four holes with a soldering iron in the same locations as the Raspberry Pi screw holes. This will hold them both together with four long screws. Finally use glass glue to glue them upside down to one of the top corners.

4. Fan

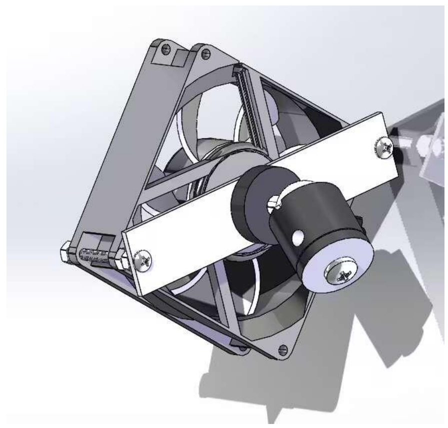


Figure 4: Fan

There are four screw holes on the corners of the fan. Use two long screws to attach an aluminum plate to the back of the fan. To prevent the aluminum plate from interfering with air flow, add a few extra nuts between the plate and the fan to increase the distance between the plate and the intake side of the fan. Attach a ball joint to the aluminum plate and screw it to the aluminum frame. This way the direction of the fan can be adjusted.

5. Camera

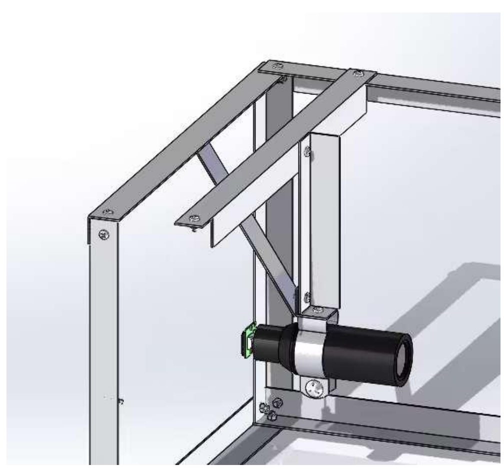


FIgure 5. Camera

After mounting the microscope on the Raspberry Pi camera, use a clip to hold the microscope and then mount the clip to the aluminum frame.

6. Battery

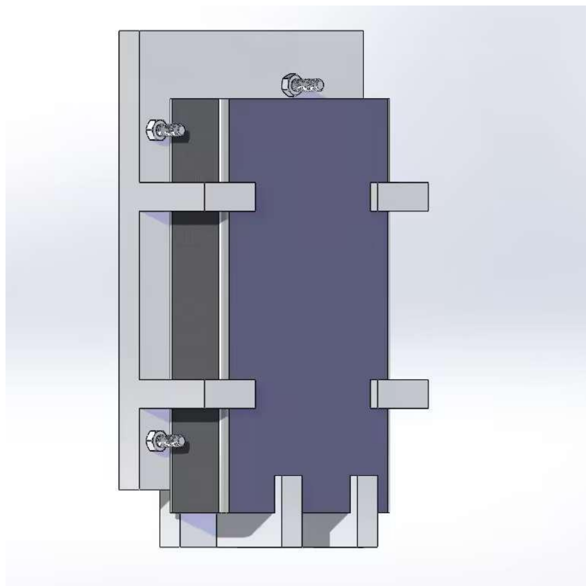


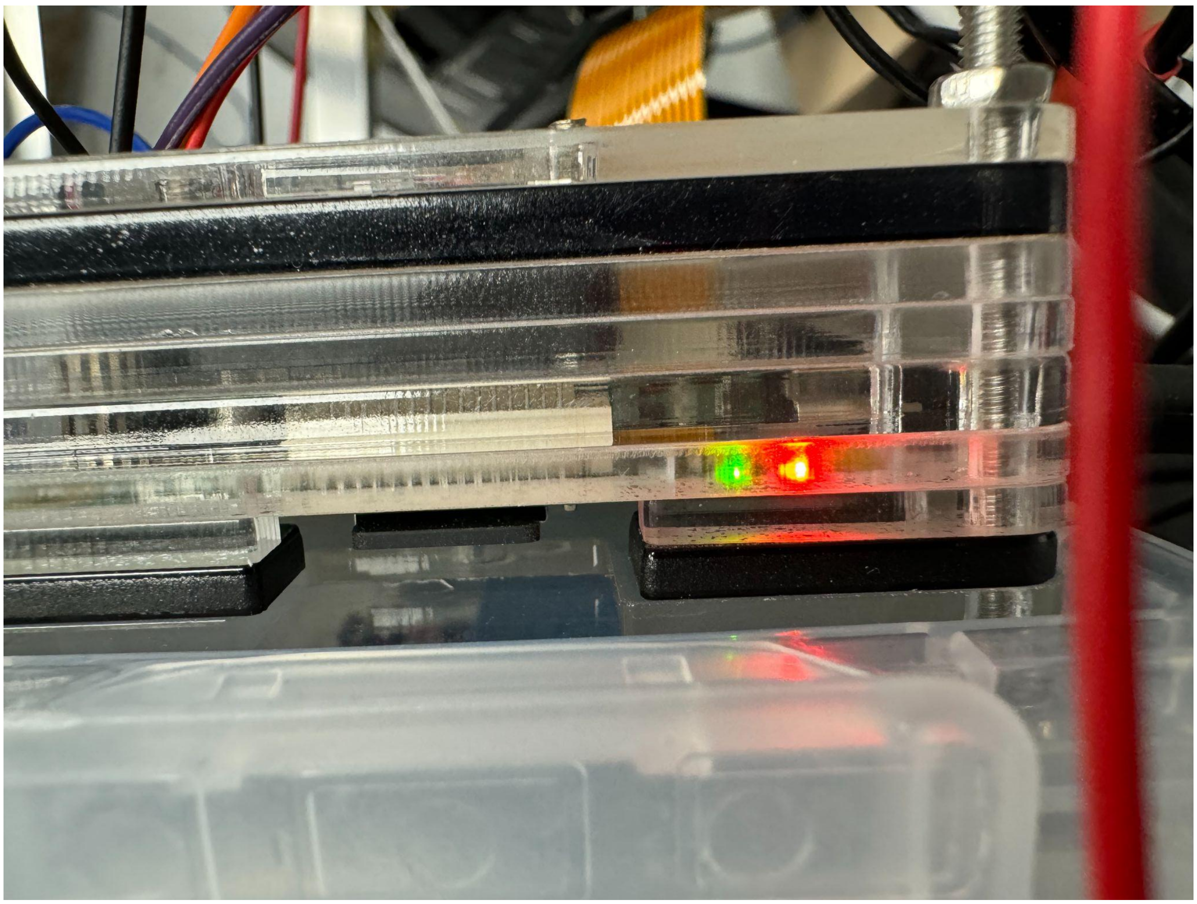
Figure 6: Battery

After printing out the battery retainer, secure the retainer to the aluminum frame with screws at the pre-drilled screw holes on the sides of the retainer. Adding another screw to the top of the battery makes it completely secure.

2. Operation Process

Before starting the operation, please ensure that the circuit connections are normal. The circuit connection should be as shown in the diagram below. Pay attention to any entangled or loose wires to prevent the risk of a short circuit. After confirming the circuit connections are normal, check that the fan and the Raspberry Pi are powered correctly. The power supply has two power interfaces; the cylindrical DC interface provides 12V for the relay and the fan, while the USB interface supplies 5V power to the Raspberry Pi and the camera module. The Raspberry Pi's power interface is a USB-C port. Once the circuit connections are complete, turn on the power switch, and you will see the red and green indicator lights on the Raspberry Pi light up.

If the Raspberry Pi lights up correctly, it means it has started correctly. If the Raspberry Pi only shows a red light without a green light, this means there is no video output from the Raspberry Pi. The video output of the Raspberry Pi is MINI HDMI. After connecting the Raspberry Pi to an external monitor, turn the power off and on again to restart. If both lights on the Raspberry Pi are lit at this time, it indicates that it is working normally. You can see the Raspberry Pi's startup screen on the monitor and then the Raspberry Pi desktop.



Control of the Raspberry Pi is divided into wired and remote connections.

Wired connection: Connect the Raspberry Pi to a monitor and external keyboard and mouse to establish a wired connection. Both the mouse and keyboard can be connected to any of the four USB ports on the Raspberry Pi.

Wireless connection: For initial use, it is recommended to first set up a wired connection. Please follow these steps for connecting:

Enable the VNC Server on the Raspberry Pi:

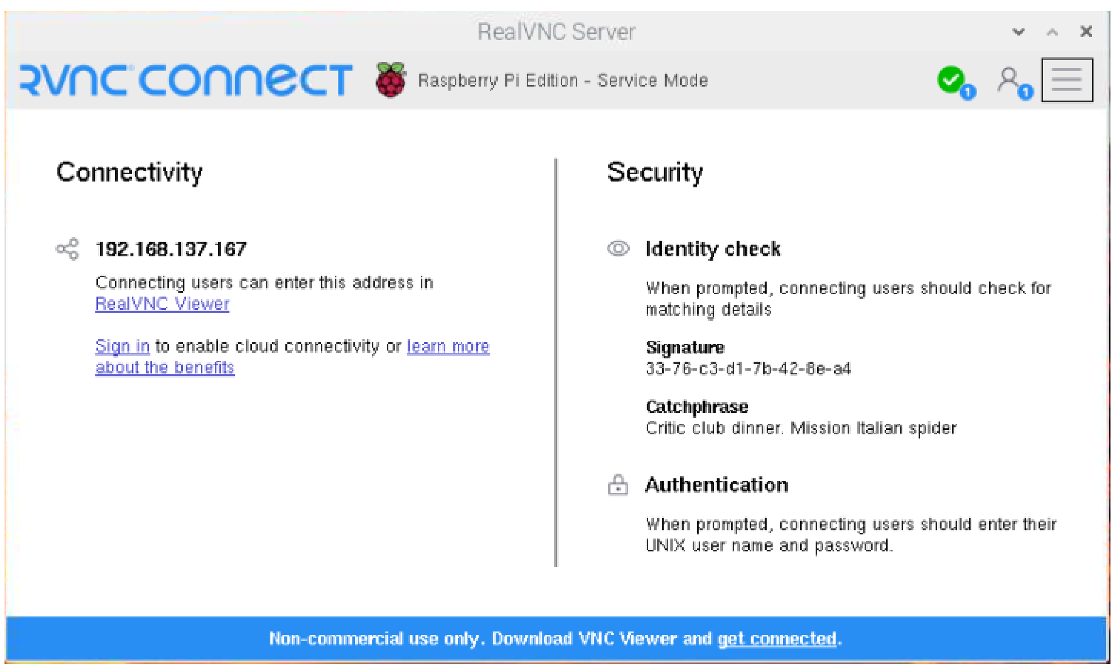
- First, ensure your Raspberry Pi has the VNC server installed. The Raspberry Pi operating system in this project comes pre-installed with RealVNC server.
- Open the Raspberry Pi terminal.
- Run code `'sudo raspi-config'`.
- Use the keyboard to navigate to '**Interfacing Options**'.
- Select VNC and enable it.
- Exit the configuration tool and restart the Raspberry Pi if necessary.

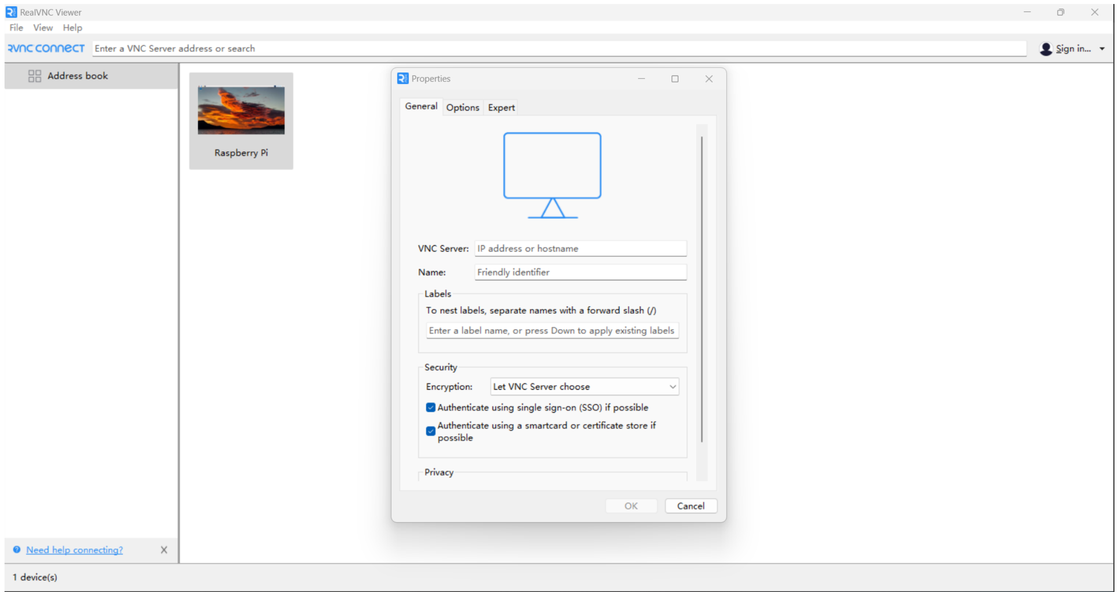
Install the VNC Client on Another Device:

- On the device you want to use for remote access to the Raspberry Pi, such as a laptop, install VNC Viewer:
- Visit the RealVNC website and download VNC Viewer.
- Install VNC Viewer on your computer.

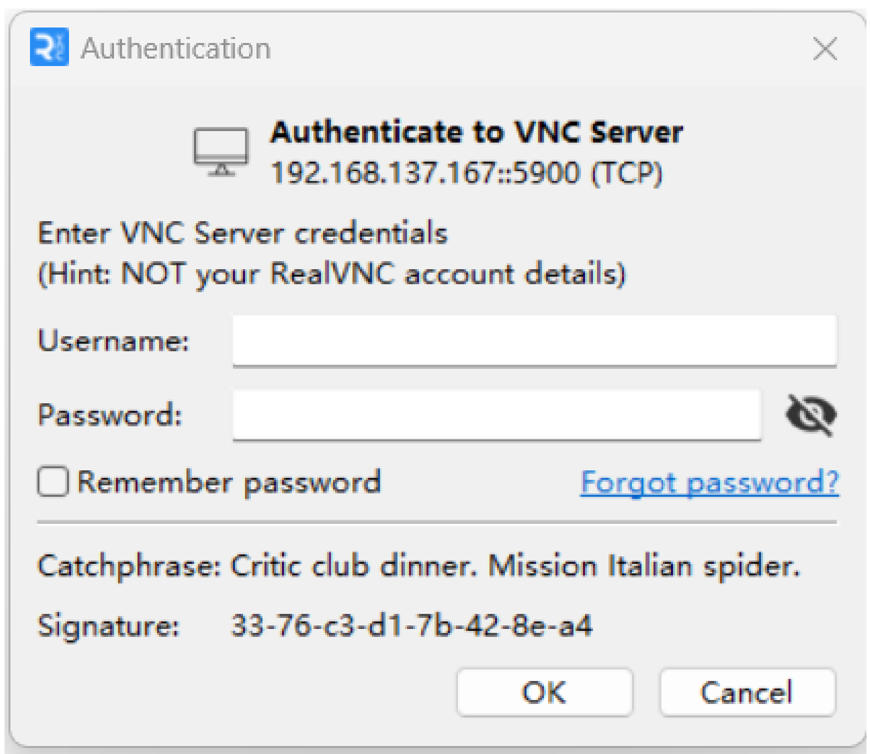
Connect to the Raspberry Pi:

- Ensure your Raspberry Pi and your computer are on the same network. This project uses the laptop's built-in network hotspot to connect the Raspberry Pi.
- Open VNC Viewer on your computer. Click on 'File' in the top left corner and select 'New Connection
- Enter the IP address of the Raspberry Pi. To find the IP address of the Raspberry Pi, enter '`hostname -I`' in the Raspberry Pi terminal.

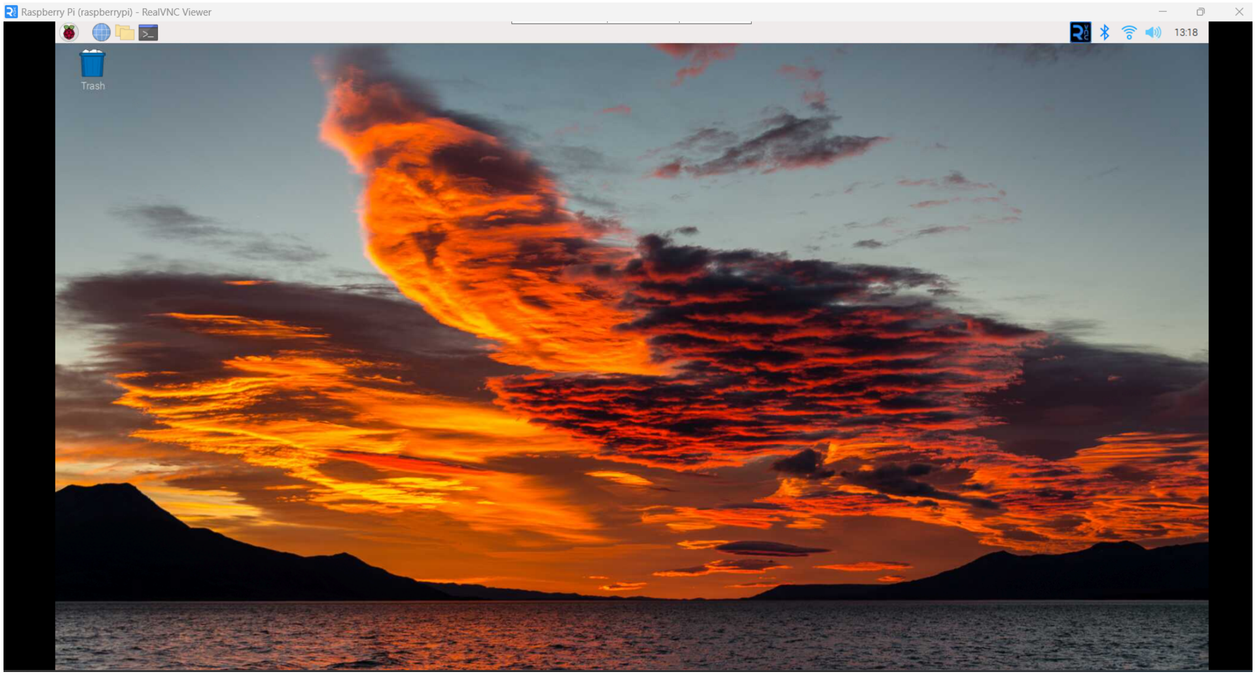




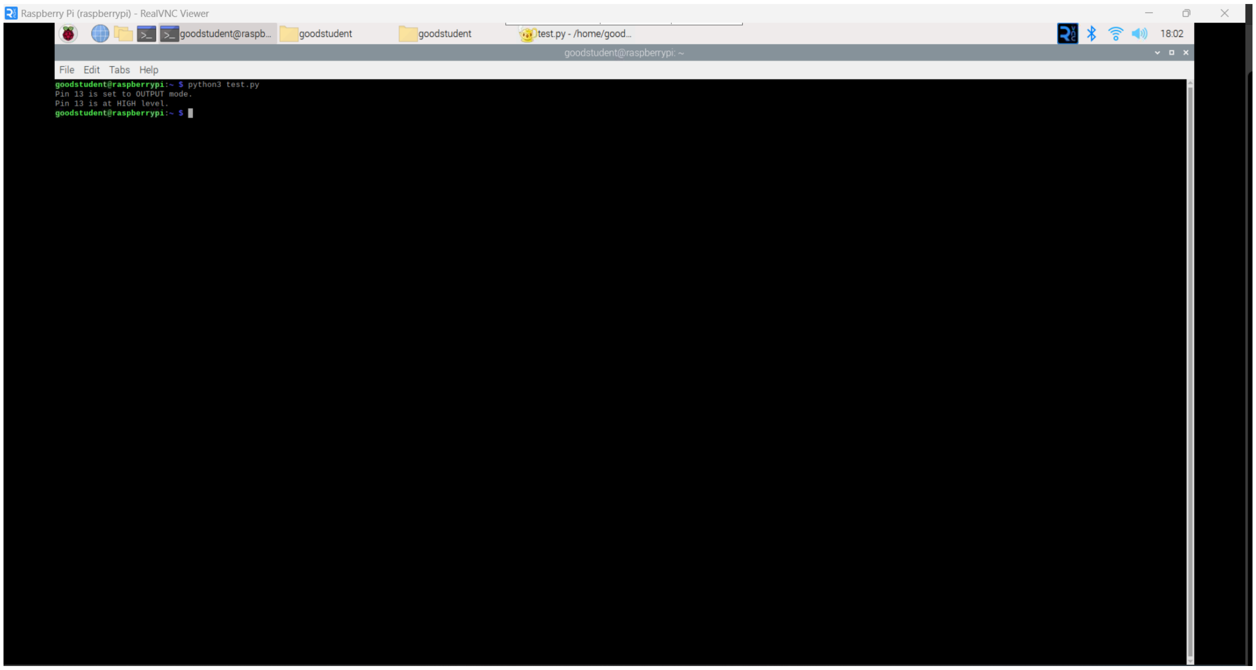
- Enter the username and password for your Raspberry Pi. In this project, the username is '**goodstudent**' and the password is the initial password '**Raspberry**'.



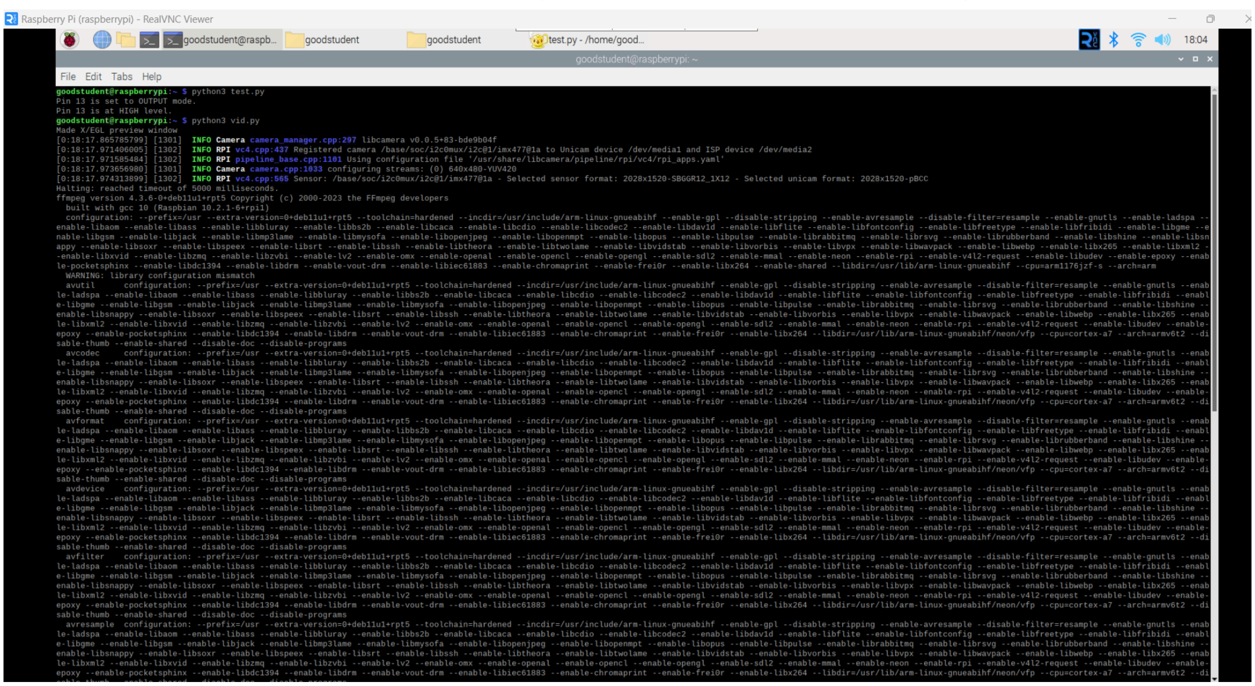
- Once connected successfully, you will be able to see and operate the desktop interface of the Raspberry Pi."

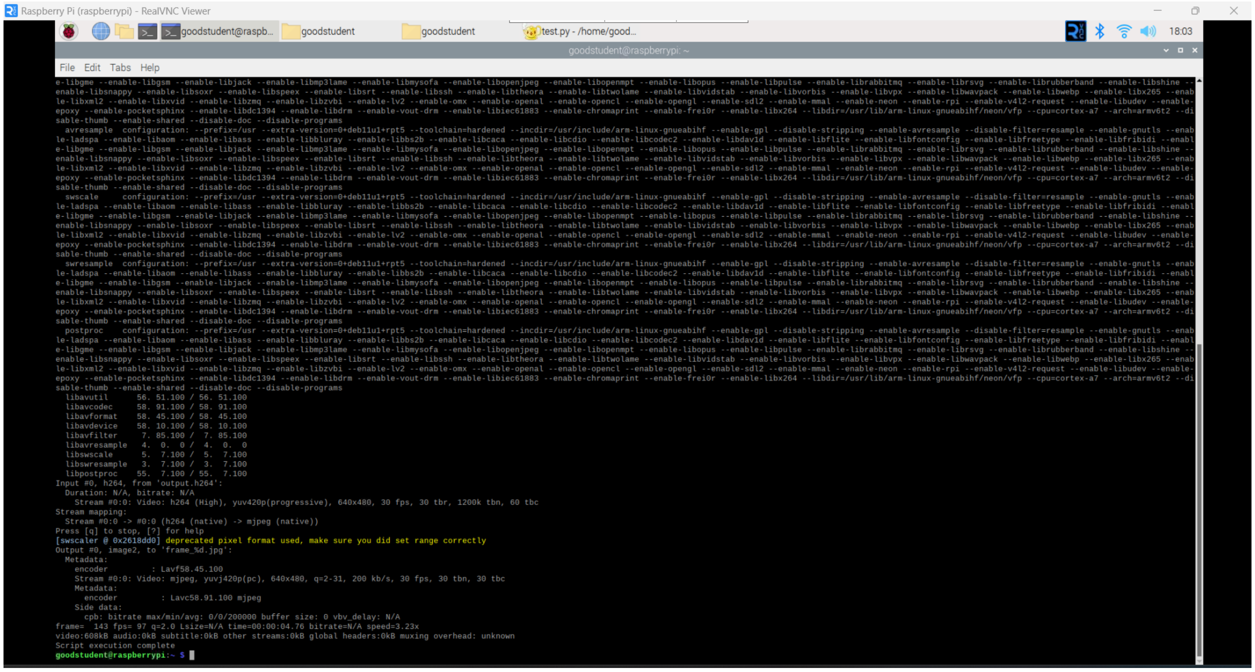


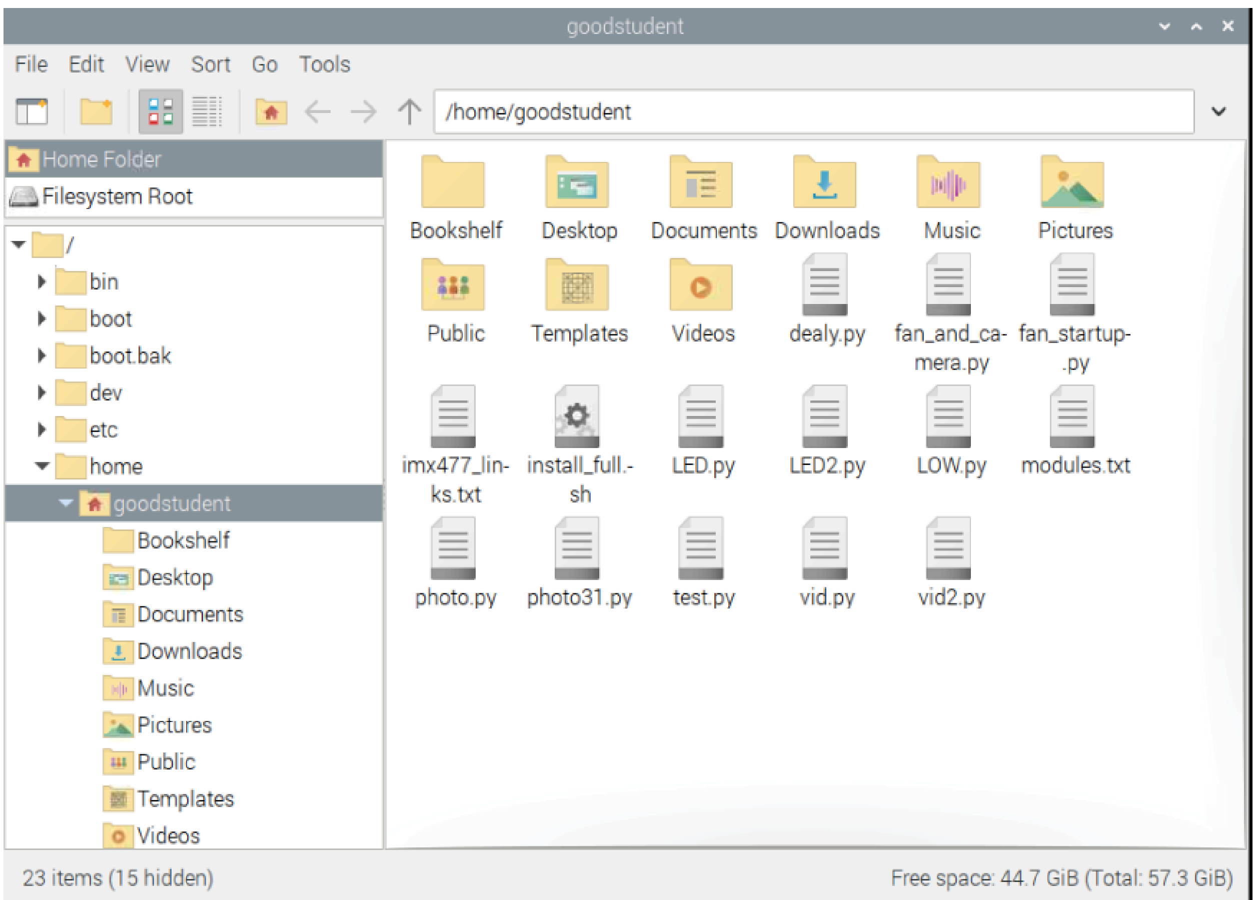
After establishing control of the Raspberry Pi, click on the top left corner to open the Raspberry Pi terminal. You can run scripts by entering code. The scripts are written in Python, so use the format `' python3 script.py'` to run them. After booting up, you can first enter `'python3 test.py'`. This script checks the voltage levels of all the Raspberry Pi pins and changes the pins that are by default in input mode to output mode. Once the test.py script has finished, you can start running the camera scripts. There are two camera scripts: vid.py records at a resolution of 630x480 at 30fps, and vid2.py records at a resolution of 1920x1080 at 24fps. For example, to run vid.py, enter `'python3 vid.py'` in the terminal, and the recording program will start running.

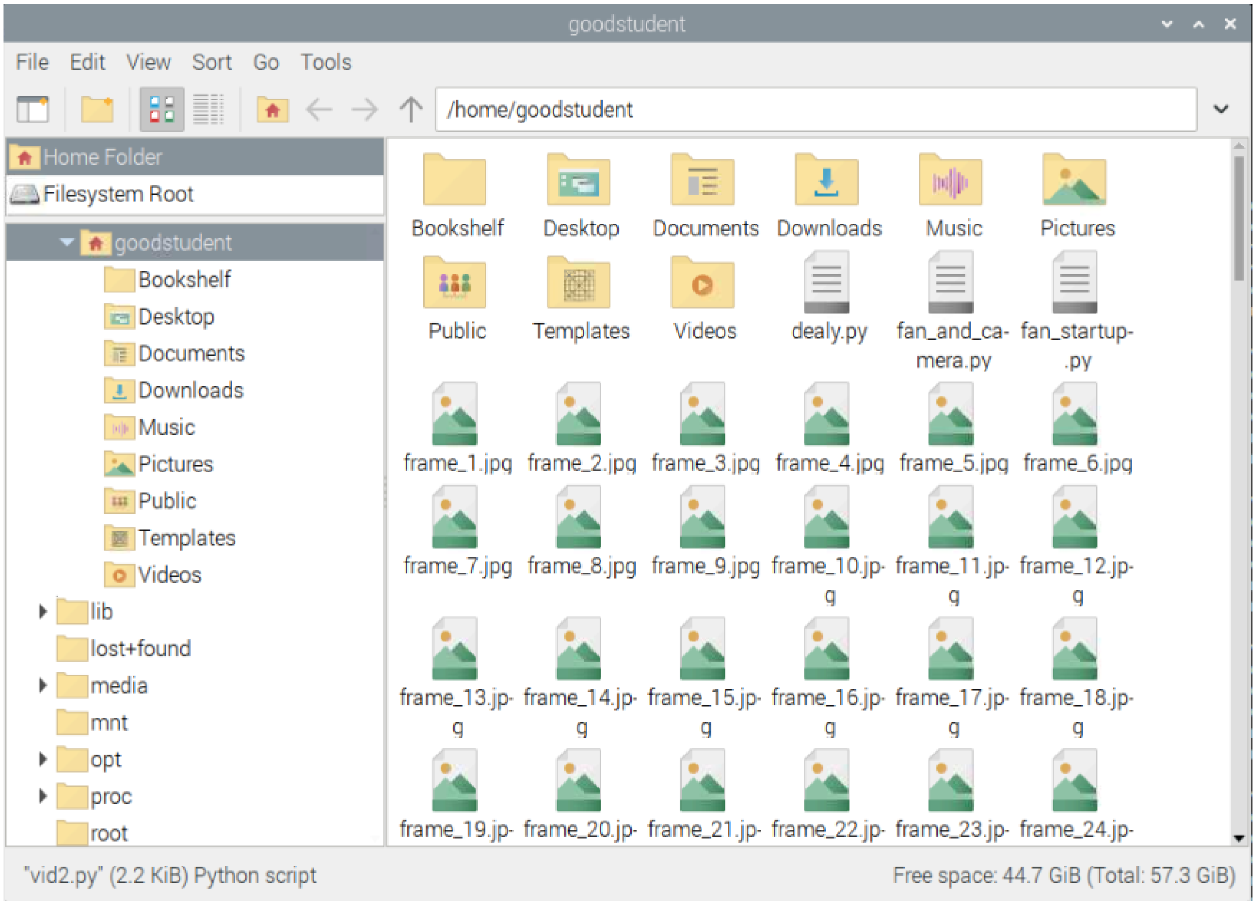


The program will control the fan to start running, and it will begin a 5-second video recording starting at the fifth second. After the recording ends, the fan and camera will automatically shut off, and the video file will be stored locally as `'output.h264'` in the default folder of the user named **Goodstudent.** The script will also use the ffmpeg command to extract frames from the video. A 5-second video shot at 30fps will extract 150 frames, numbered from 1 to 150 in jpg format, and stored in the same location. It is important to note that each time you run the vid.py script, the video and images recorded will have the same name and will overwrite the previous results. Please manually save the recording results after each run of the program.




Raspberry Pi (raspberrypi) - RealVNC Viewer
goodstudent@raspberrypi: ~
File Edit Tabs Help


goodstudent
File Edit View Sort Go Tools
/home/goodstudent
Home Folder
Filesystem Root
/
bin
boot
boot.bak
dev
etc
home
goodstudent
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
dealy.py
fan_and_camera.py
fan_startup.py
imx477_links.txt
install_full.sh
LED.py
LED2.py
LOW.py
modules.txt
photo.py
photo31.py
test.py
vid.py
vid2.py
23 items (15 hidden)
Free space: 44.7 GiB (Total: 57.3 GiB)



3. Maintenance Instructions

Maintenance is focused on ensuring a high level of transparency in the shooting window, the integrity of the 3D printed components, the battery and the proper functioning of the Raspberry Pi. Although the glass in the shooting window is a conductive material, there will still be dirt and dust particles that can adhere to the inner and outer walls and will need to be wiped down frequently with a dust-free cloth. Regular inspection of structural components for cracks, deformations, and breaks (especially in 3D printed components) is critical. Batteries should be monitored for signs of expansion or leakage and replaced as necessary. There is a risk of the screws loosening after a long period of time because of the fan and the vibration of the vacuum machine during operation. The screws on the entire unit should be checked regularly for stability.

The filter element in the vacuum machine should be cleared of soil dust attached to the filter element after each use, and should be replaced if it cannot be cleared after a period of time resulting in reduced suction power.

1. Lens Maintenance: The project uses high-transmittance conductive ITO coated glass (ITO coated glass) for the shooting window, which is costly, thus requiring maintenance of the lens. Use a cotton cloth or cotton swab to extend from the Channel outlet end to wipe off the sand and dust adhered to the lens, ensuring the cleanliness and high transmittance of the lens for capturing clear images. Since both the lens and Channel are conductive, they can be treated for static electricity elimination to prevent dust particles from adhering to the lens surface due to static electricity during flow.
2. Raspberry Pi Maintenance: The electronic components of the Raspberry Pi are quite sensitive, including the connection of wires, aging of circuit boards, and the stability of the camera module. After multiple uses, the circuit board may accumulate dust, and dust particles may penetrate the protective cover of the relay and the Raspberry Pi. Please regularly use WD-40, which is friendly to electronic components, to dust off, or use electronic spray cans containing difluoroethane or tetrafluoroethane to remove dust particles adhered to the electronic devices. Ensure that there are no dust particles on the Raspberry Pi's circuit board that could affect its operation and ensure the connections are functioning properly.
3. Maintenance of the Aluminum Frame: The aluminum frame is fixed with screws and nuts. During use, due to the operation of the fan and vacuum, the device will produce shaking and trembling. This high-frequency vibration can cause fatigue in the aluminum frame and also loosen the tightened screws. Since our safety factor is very high, the aluminum frame will not fail due to fatigue. Therefore, it is only necessary to regularly check and reinforce the tightness of the screws to prevent the loss of washers and nuts leading to structural disassembly. Additionally, the bracket of the Raspberry Pi camera is fixed with a combination of welding and glue. If necessary, more glue can be added for reinforcement to prevent failure.
4. Maintenance of 3D Printed Parts: The project uses 3D printed resin to make the Channel and sprays it with a conductive coating to prevent the accumulation of static electricity. The coating may slowly wear off during use due to friction from dust particles, and the 3D printed materials can slowly oxidize under sunlight due to outdoor use. To prevent failure, it is necessary to regularly apply a conductive coating to the surface for repairs. This not only maintains its conductive properties and uniform appearance but also prevents aging due to ultraviolet radiation. Additionally, this can strengthen the Channel's attachment to the Vacuum for increased stability.
5. Maintenance of the Lens: As a key component for capturing images, proper maintenance of the lens not only ensures the performance of the equipment but

also extends its lifespan. The lens module is separate from the shooting module and can be replaced. During use, it may experience a decrease in airtightness, scratches on the lens, and a decline in magnification accuracy. These damages are to some extent irreversible. The main way to maintain the lens is to remove surface dust, especially those adhering to the lens and the magnification adjustment knob. Using an air can to blow off dust particles can be effective, and a soft, fine-fiber cloth can also be used to wipe the lens for maintenance.

6. Vacuum Maintenance: The filter in the vacuum will decrease in breathability during use. After each use, the adhered and collected dust should be cleaned. If the filter's pores enlarge due to long-term use, or if excessive dust accumulation leads to clumping and sticking, the filter should be replaced. The exhaust port of the vacuum may become clogged due to reduced filtration efficiency of the filter, leading to decreased suction capability or affecting the motor module, potentially causing overheating. Therefore, regularly clean the vacuum's exhaust port to ensure smooth operation.

4. Potential failures and Solution

The following provides detailed and expert guidance on resolving potential faults in the whole device, focusing on practical solutions and the importance of balancing cost, complexity and system reliability.

- **Material Failure Mitigation**

  In the event of material failures, such as damage to the 3D printed resin, fan blades, or corrosion of the frame, immediate action is necessary. To mitigate these failures, we recommend using more durable materials and instituting a regular replacement schedule for vulnerable parts. While this approach may increase the project's cost and complexity, it is essential for long-term system reliability and performance.

- **Electronic Component Protection**

  For the protection of electronic components from soil dust, a dust proof enclosure for the Raspberry Pi and camera should be implemented. This measure has been tested and proven effective in preventing dust from damaging electronic equipment. Be aware that while this increases costs and maintenance requirements, it is crucial for safeguarding critical electronic components. Note that new risks may arise from these enclosures, such as potential heat dissipation issues affecting electronic components.

- **Material Strength and Durability**

  Addressing the brittle fracture of the 3D printing resin involves using a high-strength resin and advanced printing technology. Pressure and impact tests should be conducted to verify the robustness of these materials. For fan blades, use specially designed impact-resistant blades. While stronger materials may increase the cost and weight, achieving a balance between strength and weight is vital for optimal material selection.

- **Electronic Component and Operational Failures**

  To prevent electronic component failures like battery expansion and leakage, choose a high-quality battery with an intelligent charge/discharge system. Conduct thorough code reviews and testing for software components, such as those in the Raspberry Pi. Incorporating additional monitoring and protection circuitry is recommended, though it may increase the complexity and energy consumption of the design.

- **Dual Connection System for Reliability**

  For connection or operational failures, implement a dual connection system (wireless and wired) for the Raspberry Pi. This approach adds complexity but significantly enhances system reliability. Regular network stability tests are advised to ensure ongoing performance. Additionally, for issues like filter clogging, design an easy-to-replace filter system and adjust the filter size according to the actual dust size.

- **Mitigating Vibration and Structural Failures**

  To mitigate failures caused by wobbly shooting parts, add additional support structures and design a vibration damping mechanism, such as using shock-absorbing materials or a spring system. Conduct structural stability tests and vibration analysis to ensure the design can withstand additional loads without oscillating, thus ensuring image quality.

- **Fan Blockage and Blade Damage**

  If the fan experiences blockage or blade damage, possibly due to foreign objects, it can lead to significant hazards. Implement a dust cover around the fan for maintenance, and use fans designed for dusty environments to enhance tolerance. Regular durability and collision testing of the fan should be conducted to ensure safety.

- **Mechanical Failure - Brittle Fracture**

  This can occur due to the insufficient strength of the 3D printed resin, leading to cracks and fractures. To mitigate this, consider printing multiple models for selection, designing better mechanical models, or testing new resins.

- **Battery Swelling and Leakage**

  Overcharging, over-discharging, or exposure to abnormal environmental temperatures can lead to battery swelling or leakage. Regularly monitor charging and discharging currents, and use high-quality batteries to prevent these issues.

- **Creation of Eddy Currents in Collection Channel**

  The inner wall of the collection channel should be streamlined to prevent soil dust fluid from becoming turbulent, which can affect the efficiency of dust collection. Optimizing the channel design is crucial to maintaining the accuracy of the collection process.

- **Unstable Power Supply**

  To address the instability of the power supply, especially with smaller batteries, consider using more efficient batteries or adding a backup power system. This approach may add weight and cost but is critical for system stability.

- **Raspberry Pi Software and Network Issues**

  Issues with software errors or network instability in the Raspberry Pi can lead to significant operational failures. Thorough testing of all equipment, multiple runs of software for stability checks, and the preparation of backup plans are essential for preventing these failures.

- **Pump Overheating**

  Poor heat dissipation, long-term use, or internal damage can cause the pump to overheat. To mitigate this, consider positioning the pump in the fan's air duct for cooling, or create a sunshade to avoid direct sunlight. Regular testing of the pump's tolerance is also crucial to prevent failures during experiments.

- **Wobbly Shooting Parts**

  If the soil dust collection channel is susceptible to wobbling, this can blur the captured images. Adding a ball joint between the vacuum and the frame can make the collection device's angle adjustable, but careful design is needed to avoid excess movement.

- **Frame Corrosion or Oxidation**

  The frame can become brittle or soft when it comes into contact with corrosive chemicals. Selecting the right materials, like aluminum alloy or stainless steel, is necessary to ensure safety, durability, and minimal weight.

5. Appendix

Code for script ‘vid.py’

```
import RPi.GPIO as GPIO
import time
import subprocess
import threading

# Define GPIO and PWM parameters
FAN_PIN = 13  # Fan connected to physical pin 13 (GPIO27)
LED_PIN = 38  # LED connected to physical pin 38 (GPIO20)
FAN_SPEED = 75  # Represented as a percentage
PWM_FREQUENCY = 100  # PWM frequency
VIDEO_DURATION = 5000  # Video length of 5000 milliseconds (5 seconds)

# Use BOARD numbering
GPIO.setmode(GPIO.BOARD)
GPIO.setup(FAN_PIN, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED_PIN, GPIO.OUT, initial=GPIO.LOW)

pwm = GPIO.PWM(FAN_PIN, PWM_FREQUENCY)
pwm.start(FAN_SPEED)  # Start the fan

# Define a function to record video and blink LED
def record_video_and_blink_led():
    # Start recording video
    process = subprocess.Popen(["libcamera-vid", "-t", str(VIDEO_DURATION), "-o",
"output.h264"])
```

```
    # Wait for two seconds before starting LED blinking
    time.sleep(3)

    # Blink twice per second for the next 3 seconds
    for _ in range(5):
        for _ in range(2):  # Blink twice per second
            GPIO.output(LED_PIN, GPIO.HIGH)
            time.sleep(0.02)  # Blink duration of 100 milliseconds
            GPIO.output(LED_PIN, GPIO.LOW)
            time.sleep(0.48)  # Remaining time to complete the half second
        time.sleep(0.1)  # Interval between two blinks per second

    # Wait for the video recording process to end
    process.wait()

try:
    # Run the fan for the first 3 seconds
    time.sleep(3)

    # Create and start a thread to record video and blink LED simultaneously
    video_led_thread = threading.Thread(target=record_video_and_blink_led)
    video_led_thread.start()

    # Wait for the thread to complete
    video_led_thread.join()

    # Continue running the fan for the remaining time (total of 9 seconds, 3 seconds
already run)
    time.sleep(1)
```

```
    # After video recording is complete, execute the ffmpeg command for frame
extraction
    subprocess.run(["ffmpeg", "-i", "output.h264", "-qscale:v", "2", "-vf", "fps=30",
"frame_%d.jpg"])
finally:
    # Turn off PWM and clean up GPIO configuration
    pwm.ChangeDutyCycle(0)  # Set PWM to 0% to turn off the fan
    pwm.stop()
    GPIO.cleanup()

print("Script execution complete")
```

Code for script 'vid2.py'

```
import RPi.GPIO as GPIO
import time
import subprocess
import threading

# Define GPIO and PWM parameters
FAN_PIN = 13  # Fan connected to physical pin 13 (GPIO27)
LED_PIN = 38  # LED connected to physical pin 38 (GPIO20)
FAN_SPEED = 75  # Represented as a percentage
PWM_FREQUENCY = 100  # PWM frequency
VIDEO_DURATION = 5000  # Video length of 5000 milliseconds (5 seconds)

# Use BOARD numbering
GPIO.setmode(GPIO.BOARD)
GPIO.setup(FAN_PIN, GPIO.OUT, initial=GPIO.LOW)
```

```
GPIO.setup(LED_PIN, GPIO.OUT, initial=GPIO.LOW)

pwm = GPIO.PWM(FAN_PIN, PWM_FREQUENCY)
pwm.start(FAN_SPEED)  # Start the fan

# Define a function to record video and blink LED
def record_video_and_blink_led():
    # Start recording video with 1920x1080 resolution
    process = subprocess.Popen(["libcamera-vid", "-t", str(VIDEO_DURATION), "-o",
"output.h264", "--width", "1920", "--height", "1080"])

    # Wait for two seconds before starting LED blinking
    time.sleep(3)

    # Blink twice per second for the next 3 seconds
    for _ in range(5):
        for _ in range(2):  # Blink twice per second
            GPIO.output(LED_PIN, GPIO.HIGH)
            time.sleep(0.02)  # Blink duration of 100 milliseconds
            GPIO.output(LED_PIN, GPIO.LOW)
            time.sleep(0.48)  # Remaining time to complete the half second
        time.sleep(0.1)  # Interval between two blinks per second

    # Wait for the video recording process to end
    process.wait()

try:
    # Run the fan for the first 3 seconds
    time.sleep(3)

    # Create and start a thread to record video and blink LED simultaneously
```

```
    video_led_thread = threading.Thread(target=record_video_and_blink_led)
    video_led_thread.start()

    # Wait for the thread to complete
    video_led_thread.join()

    # Continue running the fan for the remaining time (total of 9 seconds, 3 seconds
already run)
    time.sleep(1)

    # After video recording is complete, execute the ffmpeg command for frame
extraction
    subprocess.run(["ffmpeg", "-i", "output.h264", "-qscale:v", "2", "-vf", "fps=30",
"frame_%d.jpg"])
finally:
    # Turn off PWM and clean up GPIO configuration
    pwm.ChangeDutyCycle(0)  # Set PWM to 0% to turn off the fan
    pwm.stop()
    GPIO.cleanup()

print("Script execution complete")
```