

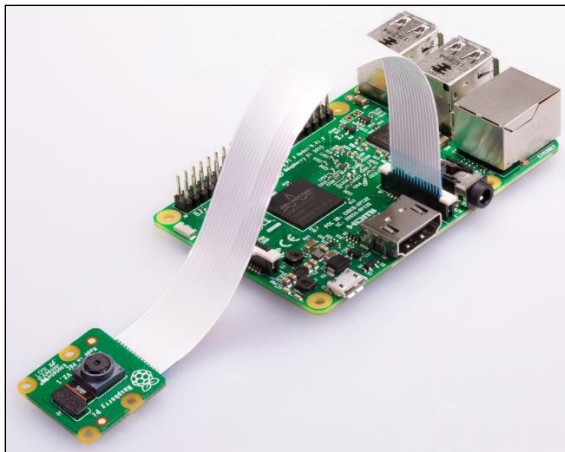
GoPiGo Raspberry Pi Camera

Contents

GoPiGo Raspberry Pi Camera	1
Raspberry Pi Camera	1
Remove Top and Redboard	2
Mount the Camera	9
Enable the Raspberry Pi Camera Port.....	10
Control the Camera at the Terminal.....	11
Control the Camera with Python.....	11
Take a Photo in a Loop Every 2 Seconds.....	12
Take a Photo Every 5 Minutes	13

Raspberry Pi Camera

You will have to disassemble your GoPiGo a bit to get to the Raspberry Pi to connect the Pi camera. The photos shows how the camera connects to the Pi.



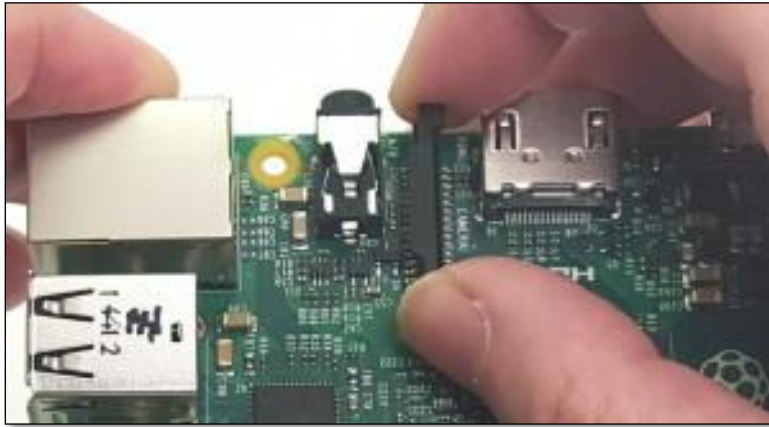
Remove Top and Redboard

1. Remove the GoPiGo top and the red board as shown.
2. Be careful pulling the red board off of the pins. You may have to rock it left and right to get it to come loose

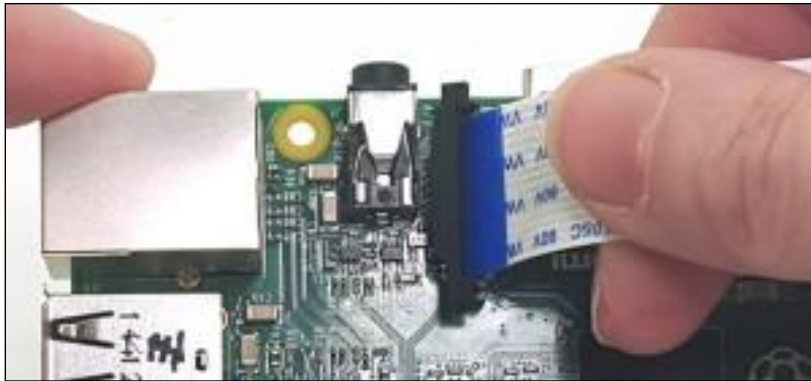


3. Before continuing, watch the first couple of minutes of this walkthrough that shows how to connect the Pi camera to the Pi.
<https://www.youtube.com/watch?v=R7BlvxPCII4&list=PLGXEJ4Ye1qCN2En7hVi-O9Z8uVJqqqMJq>

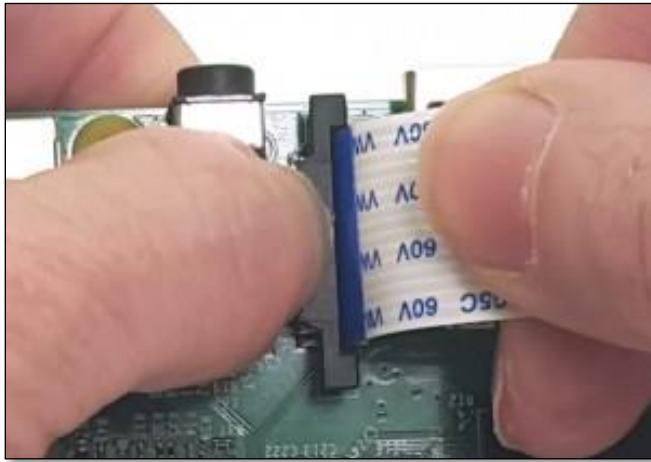
4. Open the Camera Port on the Raspberry Pi. The camera port is between the audio port and the HDMI port.
5. To open the port, use two fingers and lift the ends up slightly. Please note that there is another port on the Pi board that looks just the same, but is not meant for the camera.



6. The camera cable has to be inserted with the right orientation: the blue has to face the Ethernet port, and the silver side is facing the HDMI port. Insert the cable so that almost no blue is showing. This photo shows the beginning of the insertion.



7. Close the Camera Port.
8. To close the port and keep the cable snugly in place, push the top of the port while holding the cable with the other hand. Because you're pushing it down, more blue will be exposed. That's okay.



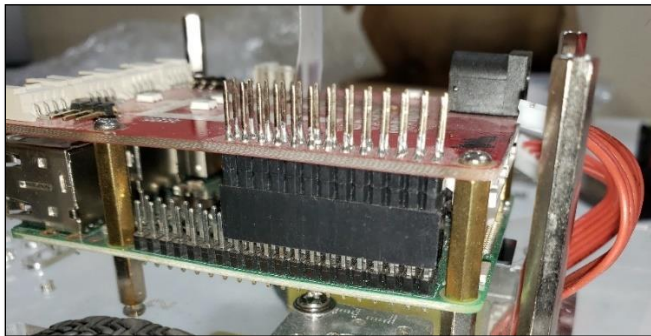
9. The completed connection.



10. Verify the connection by giving a very, very, gentle tug on the cable. No force is needed at all. If the cable is not properly held in place, it will slide out of the port. You will feel a resistance if it's inserted solidly. Just a little pull is enough.



11. Replace the red board. Push it carefully down on the pins.



12. Make sure the Pi camera cable is pulled through the slot as shown.

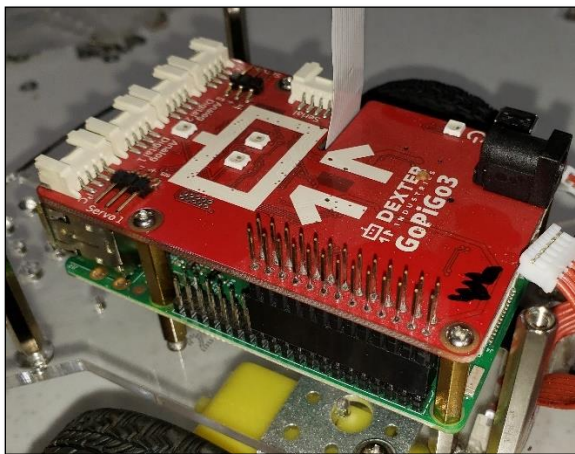


13. Plug in the motor controllers as shown. This allows enough room to remove the MicroSD card easily.

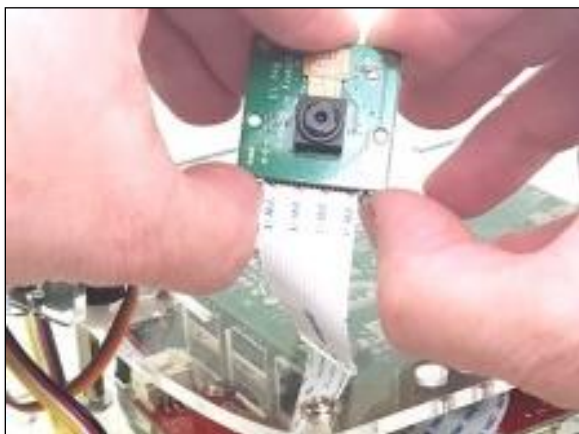
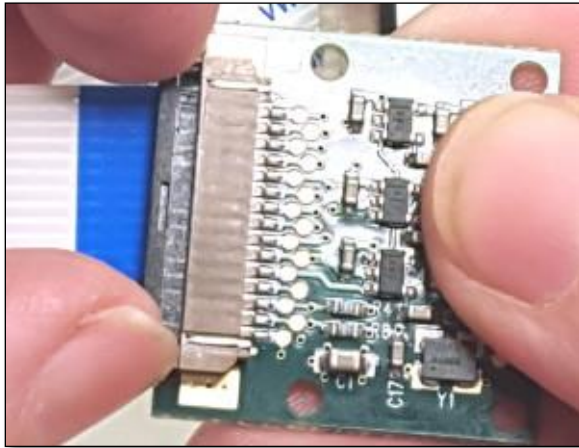


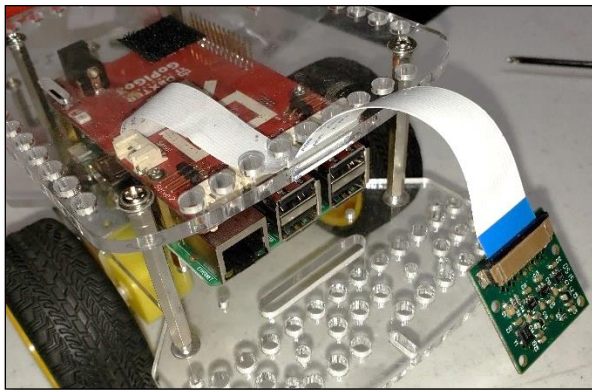
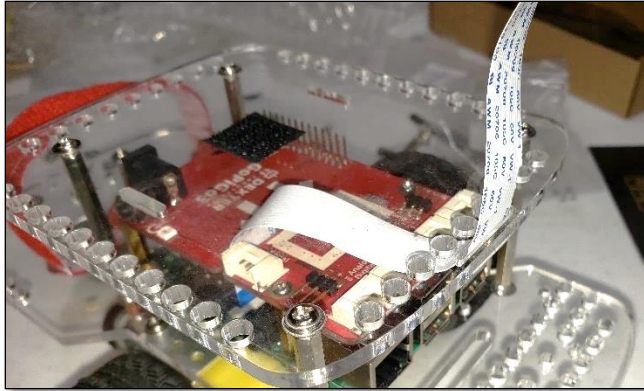
14. Carefully tighten the screws.

WARNING: If you tighten the screws on the circuit board too tightly, you can damage the Pi or the red board.



15. On the back of the camera, you will find a similar cable port. It opens and closes the same way as the one on the Raspberry Pi. You may want to have someone hold the cable in place while connecting it. It is a bit tricky to hold the cable and push in the top of the port to secure the cable.

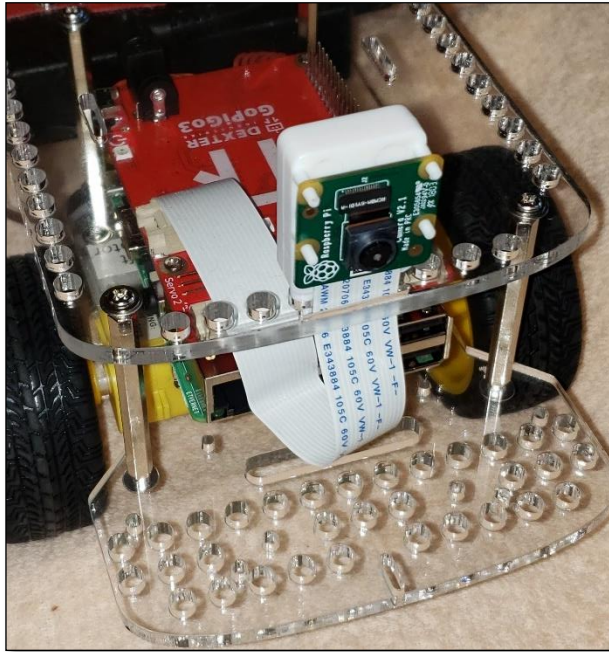




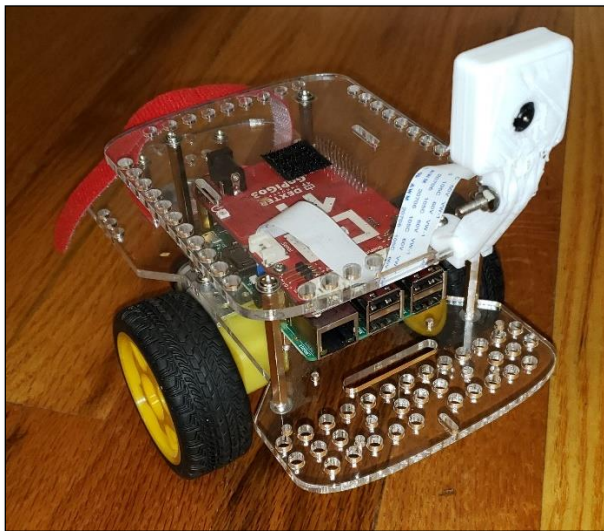
Mount the Camera

There are a couple of mounting options for the camera.

GoPiGo Sensor/Camera Mount: This picture is using the GoPiGo camera/sensor mount. The problem is that if you also have the Distance Sensor mounted, the camera is blocked.



3D printed camera holder: It is higher for a better view, and is out of the way of the Distance Sensor.

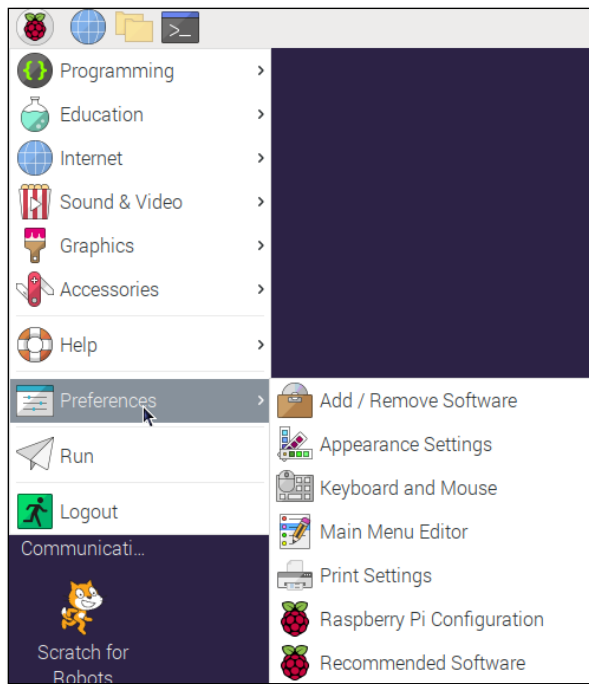


The camera can also be placed on the servo unit.

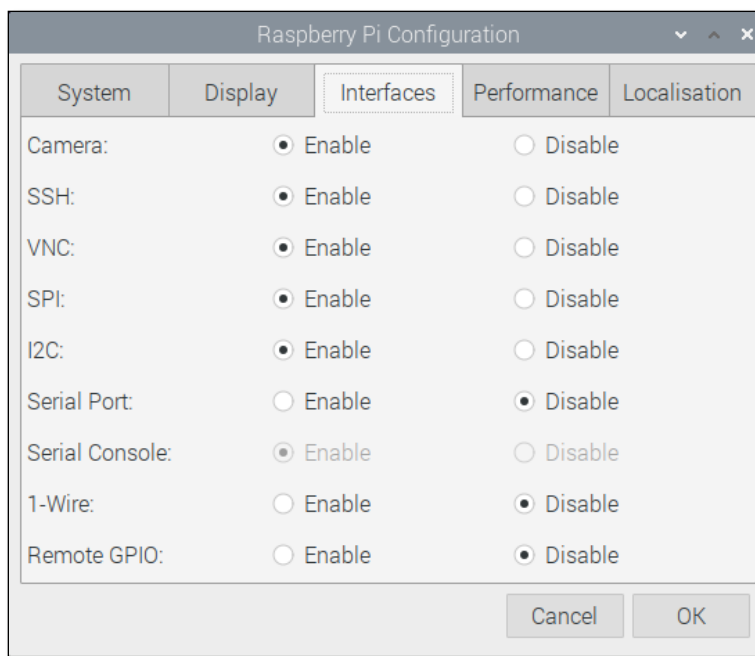
Enable the Raspberry Pi Camera Port

The first step is to boot the GoPiGo and enable the camera port.

1. Go to the **Raspberry** menu → **Preferences** → **Raspberry Pi Configuration**.



2. Enable all ports as shown. Reboot when prompted.



Control the Camera at the Terminal

Your camera is enabled. Let's test it out by the command line.

1. Open a terminal session. Run the following command to take a still image on the desktop.

```
raspistill -o Desktop/image.jpg
```

2. When the terminal prompt returns, there should be an image on your desktop. Double Click the image file to open it.

3. Let's record a short video. Run the following command.

```
raspivid -o Desktop/video.h264
```

4. Double Click the video to see it.

Control the Camera with Python

Let's use the PiCamera library to control the camera. Create the following python file in your Code folder. If the program runs successfully, you will see picamera_test.jpg on your desktop.

```
1 # -----
2 # Filename: camera_test.py
3 # Author: William A Loring
4 # Created: 10/17/21
5 # Purpose: Test PiCamera library with Pi Camera
6 # Original program from RaspberryPi.org
7 # https://picamera.readthedocs.io/en/release-1.13/recipes1.html#
8 # -----
9
10 from picamera import PiCamera # Import the PiCamera library
11 from time import sleep       # Import sleep
12
13 camera = PiCamera()          # Initialize a PiCamera object
14
15 # You won't see the preview through VNC
16 # The preview is needed to warm up the camera
17 print("Warming up camera")
18 camera.start_preview()
19 # The preview is needed to warm up the camera and sense light levels
20 sleep(2)
21
22 # Capture the image to a file
23 print("Capture the image")
24 camera.capture('/home/pi/Desktop/picamera_test.jpg')
25
26 # Shutdown the camera
27 print("Shut down the camera")
28 camera.stop_preview()
```

Take a Photo in a Loop Every 2 Seconds

```
1 # -----
2 # Filename: camera_5_images.py
3 # Author: William A Loring
4 # Created: 10/17/21
5 # Purpose: Take 5 images with Pi Camera
6 # Original program from RaspberryPi.org
7 # https://picamera.readthedocs.io/en/release-1.13/recipes1.html#
8 # -----
9
10 from picamera import PiCamera # Import the PiCamera library
11 from time import sleep # Import sleep
12
13 camera = PiCamera() # Initialize a PiCamera object
14 camera.resolution = (1024, 768) # Lower the image resolution
15
16 # You won't see the preview through VNC
17 # The preview is needed to warm up the camera
18 print("Warming up camera")
19 camera.start_preview()
20
21 # The variable i counts how many times the loop has run, from 0 to 4
22 # The images get saved as image0.jpg, image1.jpg, and so on.
23 for i in range(5):
24     # Sleep is needed to sense light levels
25     sleep(2)
26     # Capture the image to a file
27     print("Capture image %s" % i)
28     camera.capture('/home/pi/Desktop/image%s.jpg' % i)
29
30 # Shutdown the camera
31 print("Shut down the camera")
32 camera.stop_preview()
```

Take a Photo Every 5 Minutes

```
1 # -----
2 # Filename: camera_5_minutes.py
3 # Author:   William A Loring
4 # Created:  10/17/21
5 # Purpose:  Take a image every five minutes
6 # Original program from
7 # https://picamera.readthedocs.io/en/release-1.13/recipes1.html#
8 # -----
9
10 from picamera import PiCamera    # Import the PiCamera library
11 from time import sleep          # Import sleep
12
13 camera = PiCamera()              # Initialize a PiCamera object
14 camera.resolution = (1024, 768)  # Lower the image resolution
15
16 # You won't see the preview through VNC
17 # The preview is needed to warm up the camera
18 print("Warming up camera")
19 camera.start_preview()
20 sleep(2)
21
22 # Loop until program is stopped
23 for filename in camera.capture_continuous('img{counter:03d}.jpg'):
24     print('Captured %s' % filename)
25     sleep(300) # Wait 5 minutes or whatever time you choose
```