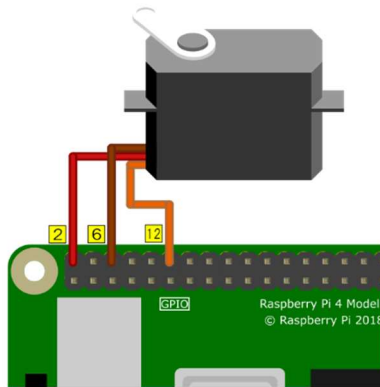


Raspberry Pi Web ストリーミングカメラ 設定手順

1. サーボモーターのセットアップとテスト

配線

1. サーボモーターの配線を行います：
 - VCC (赤) → Raspberry Pi の 5V ピン
 - GND (濃紺) → Raspberry Pi の GND ピン
 - 制御信号線 (オレンジ) → Raspberry Pi の GPIO18 ピン



pigpio ライブラリのインストール

サーボモーター制御には pigpio ライブラリを使用します。コマンドプロンプトを開き、以下のコマンドを実行することで、pigpio ライブラリをインストールします。

```
“sudo apt install pigpio python3-pigpio”
```

pigpio の起動

インストールした pigpio を起動します。コマンドプロンプトを開き、以下のコマンドを実行します。

```
“sudo pigpiod”
```

サーボモーターテスト用プログラム

1. サーボモーターテストプログラム “test_servo.py” を github からダウンロードし、Raspberry Pi に保存します。
 2. プログラムを実行して、サーボモーターが-90 度→90 度→0 度→-30 度→30 度→0 度の順番に動作することを確認します。
 3. サーボモータに付属のサポートを取り付けると動きが識別しやすくなります。動かない時は配線等を確認してください。
-

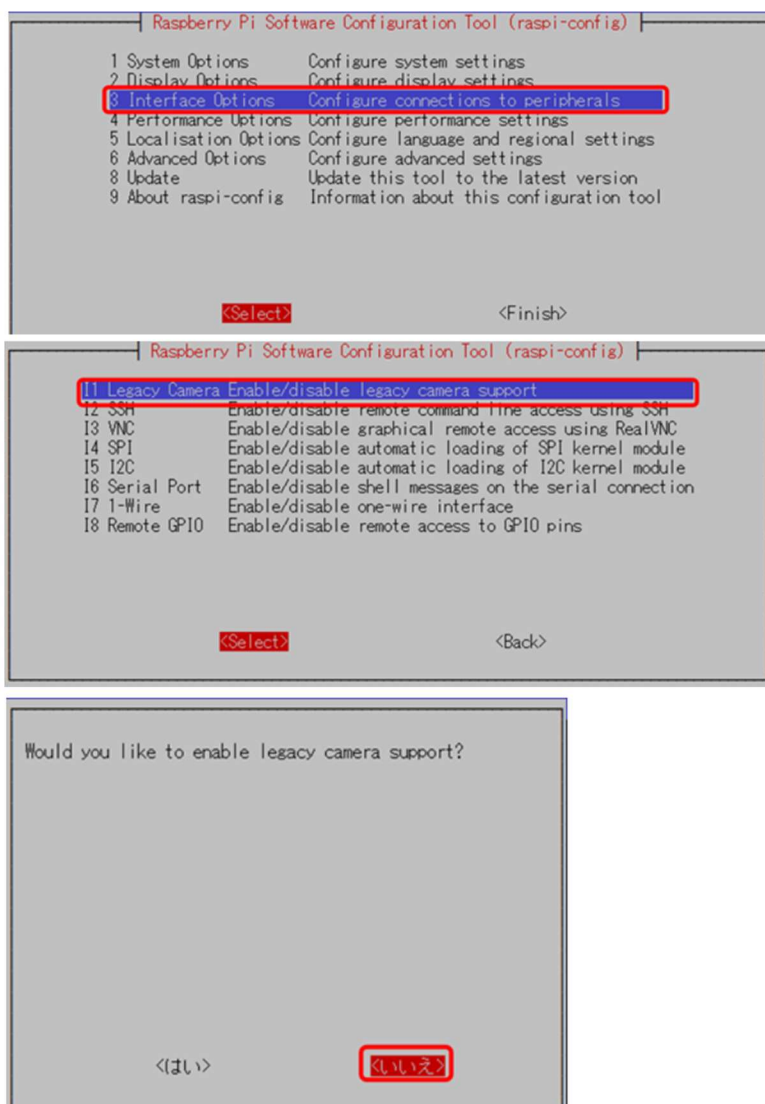
2. カメラのセットアップとテスト

カメラの接続

1. Raspberry Pi の電源を切り、カメラを接続します。カメラケーブルの銀色の接点が HDMI ポートの方向に向くようにします。

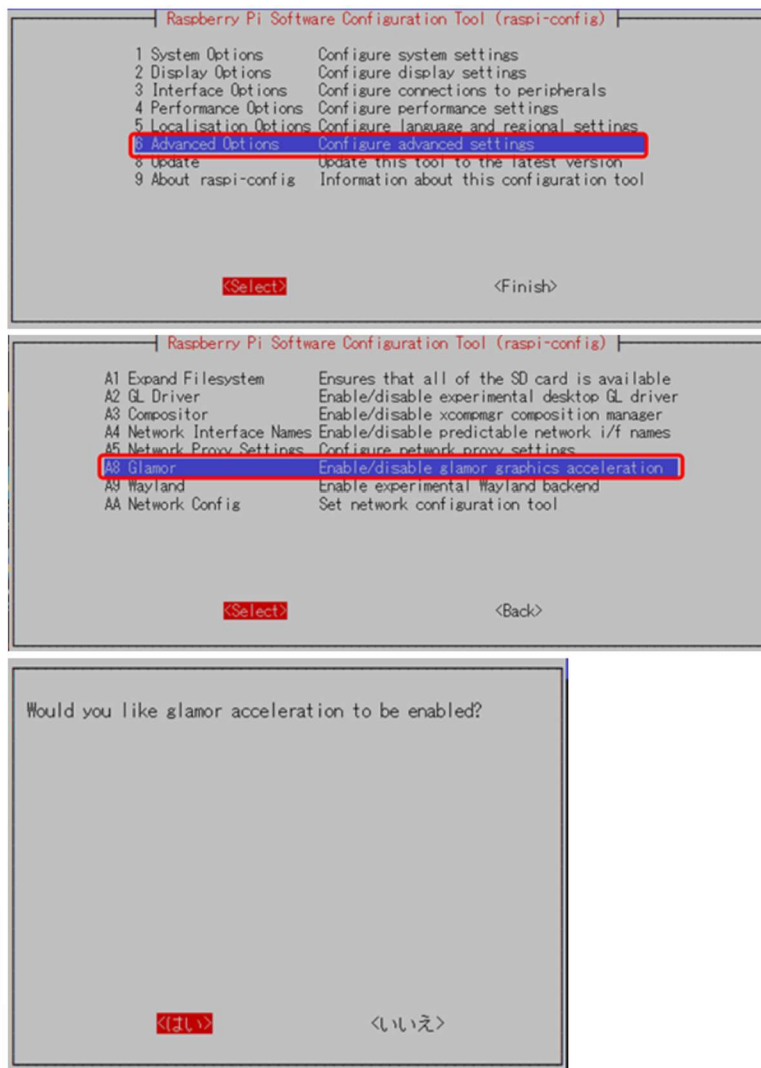
libcamera モードへの切り替え

1. （以降の説明は Raspberry Pi OS が Bullseye であることを前提としています。）
Raspberry Pi OS Bullseye では、デフォルトで libcamera モードが使用されるようになっていますが、古い設定ではレガシーカメラモードが有効になっている可能性があります。以下の手順で、libcamera モードに切り替えます。
2. ターミナルを開き、`sudo raspi-config` コマンドを実行します。下図の順番に操作し、**Legacy camera support を無効**にし、Raspberry Pi を再起動します。



Glamor の有効化

1. Glamor は、GPU によるハードウェアアクセラレーションを活用した X ウィンドウシステムの描画を高速化する技術です。OpenCV などを使用したカメラ映像の処理を効率化するために、Glamor を有効にします。
2. ターミナルを開き、`sudo raspi-config` コマンドを実行します。下図の順番に操作し、Glamor acceleration を有効にし、Raspberry Pi を再起動します。



カメラモジュールの指定

1. “/boot/config.txt”にカメラモジュールを指定します。”/boot/”フォルダに移動し、スーパーユーザー権限で”config.txt”を開きます。
2. ファイルの末尾に下記行を追記します。
“dtoverlay=ov5647”
3. Raspberry Pi を再起動します。
4. ターミナルを開き、“libcamera-hello --list-cameras”を実行します。下図のように ov5647 が認識されていれば完了です。
5. ターミナルで“libcamera-hello”を実行します。カメラの接続がうまくできている場合、プレビューが 5 秒間表示されます。

```
yaopi@camera:~$ libcamera-hello --list-cameras
Available cameras
-----
0 : ov5647 [2592x1944] (/base/soc/i2c0mux/i2c@1/ov5647@36)
    Modes: 'SGBRG10_CS12P' : 640x480 [58.92 fps - (16, 0)/2560x1920 crop]
    1296x972 [43.25 fps - (0, 0)/2592x1944 crop]
    1920x1080 [30.62 fps - (348, 434)/1928x1080 crop]
    2592x1944 [15.63 fps - (0, 0)/2592x1944 crop]
```

OpenCV および picamera2 のインストール

次に、ターミナルで以下のコマンドを実行し、必要なライブラリをインストールします。

“sudo pip3 install opencv-python”

“sudo apt install python3-picamera2”

“sudo apt install libatlas3-base”

カメラテスト

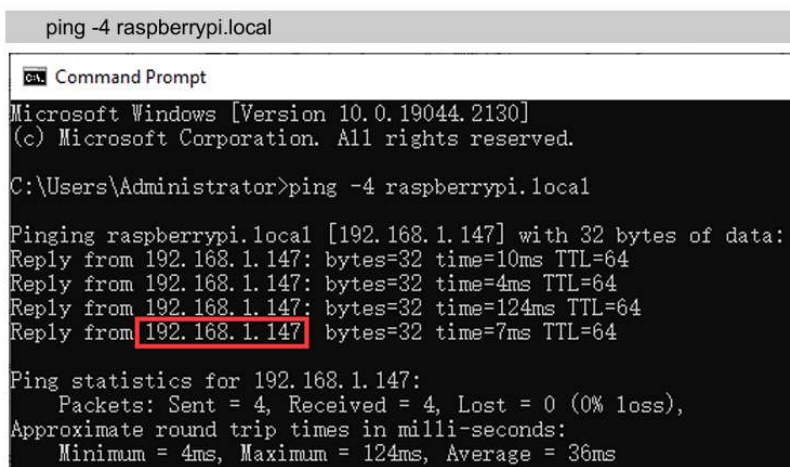
1. カメラテストプログラム “test_camera.py” を github からダウンロードし、Raspberry Pi に保存します。
 2. プログラムを実行して、プレビューが表示され、また”test_camera.py”のフォルダに”test.jpg”という名前で画像ファイルが作成されることを確認して下さい。
 3. import cv2 で numpy のモジュールに関するエラーが表示される場合、numpy のバージョンと opencv のバージョンとの組み合わせが悪い場合があります。
numpy を apt-get python3-numpy でインストールしている場合、numpy のバージョンが古すぎる可能性があります。apt-get でインストールした numpy をアンインストールし、pip3 でインストールし直しましょう。
pip3 で numpy==2.X.Y のバージョンをインストールした場合、numpy が新しすぎる可能性もあります。pip3 install numpy==1.XX.YY でバージョンを指定します。
-

3. Web ストリーミングカメラの使い方

プログラムの準備

1. サーボモータとカメラのテストが完了したら、次に Web ストリーミングカメラのプログラムを作成します。
2. ターミナルを開き、以下のコマンドを実行してパッケージをインストールします。
“sudo apt install python3-flask”
3. Raspberry Pi を WiFi に接続します。
4. ターミナルで“ping -4 raspberrypi.local”を実行し、Raspberry Pi の IP アドレスを確認します。

```
ping -4 raspberrypi.local
```



```
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping -4 raspberrypi.local

Pinging raspberrypi.local [192.168.1.147] with 32 bytes of data:
Reply from 192.168.1.147: bytes=32 time=10ms TTL=64
Reply from 192.168.1.147: bytes=32 time=4ms TTL=64
Reply from 192.168.1.147: bytes=32 time=124ms TTL=64
Reply from 192.168.1.147: bytes=32 time=7ms TTL=64

Ping statistics for 192.168.1.147:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 124ms, Average = 36ms
```

Then 192.168.1.147 is my Raspberry Pi IP.

実行

1. プログラム “webcamera_v2-0.py” を github からダウンロードし、Raspberry Pi に保存します。
2. プログラムを実行します。WEB サーバーが立ち上がります。
3. Raspberry Pi と同一の WiFi に接続した機器で Web ブラウザを立上げ、Raspberry Pi にアクセスします。url 欄に “IP アドレス:8000” と打ち込むことでアクセスが可能です。
例 : `http://192.168.1.100:8000/`
4. ブラウザにアクセスすると、カメラの映像がリアルタイムで表示されます。
5. **left** と **right** のボタンを押すと、サーボモータが 30 度回転します。サーボモーターとカメラとを連結することで、カメラの画角を WEB ブラウザから操作できるようになります。