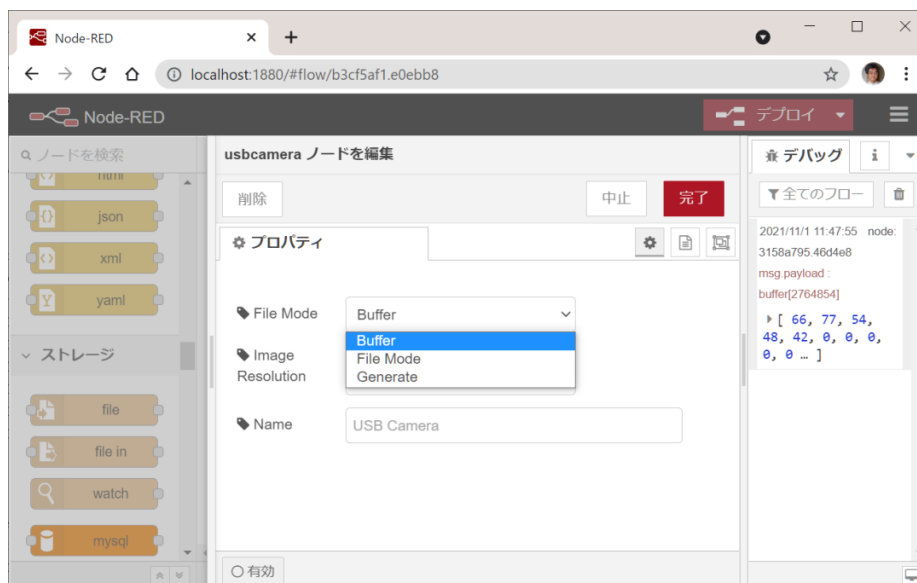
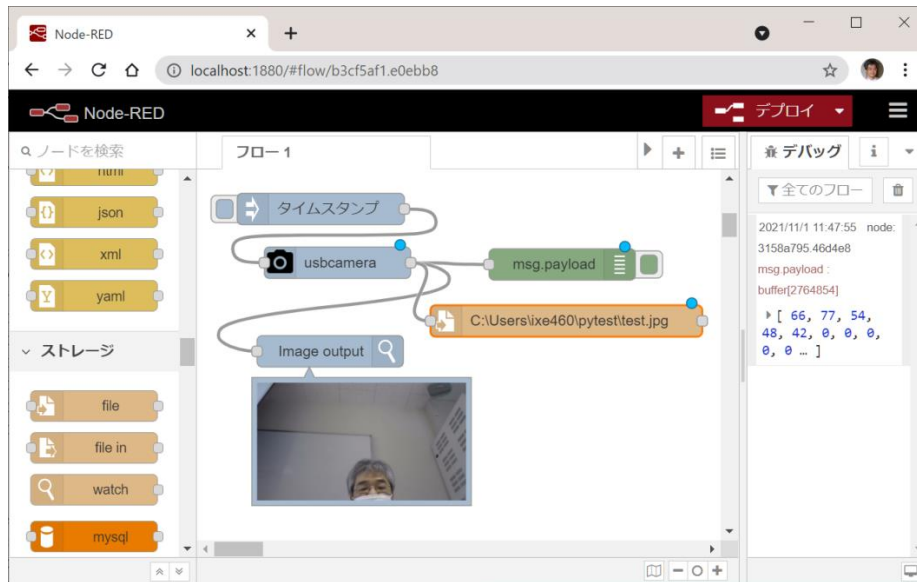


追加資料 カメラ関連で2つの例を紹介します。

○USBCAMERA ノードを使う

外部からトリガのかけられるカメラキャプチャに **USBCAMERA** ノードがあります。

追加のモジュール **node-red-contrib-usbcamera** をインストールすると利用できます。そのあとに表示等を接続する場合はモードを **Buffer** に設定します。



○Python と組み合わせて画像の中心の色情報を表示する

Python と組み合わせてみます。Python のプログラムで画像を撮影し、その中心の場所の画素情報を Node-RED 側に UDP で送信します。Node-RED 側はその値をグラフ表示します。

・ Python 側

リスト (Python) カメラから画像を取り込み、中央の画素値を UDP で送信する

```
import cv2
import socket

ap = ('127.0.0.1',5000)

client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

#client.sendto(b'Hello UDP', ap)

camera = cv2.VideoCapture(0) # カメラ Ch.(ここでは 0)を指定

# 撮影＝ループ中にフレームを 1 枚ずつ取得 (q キーで撮影終了)
while True:
    ret, frame = camera.read() # フレームを取得
    print(type(frame)) # 'numpy.ndarray'
    print(frame.shape) # y,x,z
    print(frame[240][320][0]) # B
    print(frame[240][320][1]) # G
    print(frame[240][320][2]) # R

    client.sendto(frame[240][320],ap)

    cv2.imshow('camera', frame) # フレームを画面に表示

    # キー操作があれば while ループを抜ける
    if cv2.waitKey(1000) & 0xFF == ord('q'): # 0 stop
        break

# 撮影用オブジェクトとウィンドウの解放
camera.release()
cv2.destroyAllWindows()
```

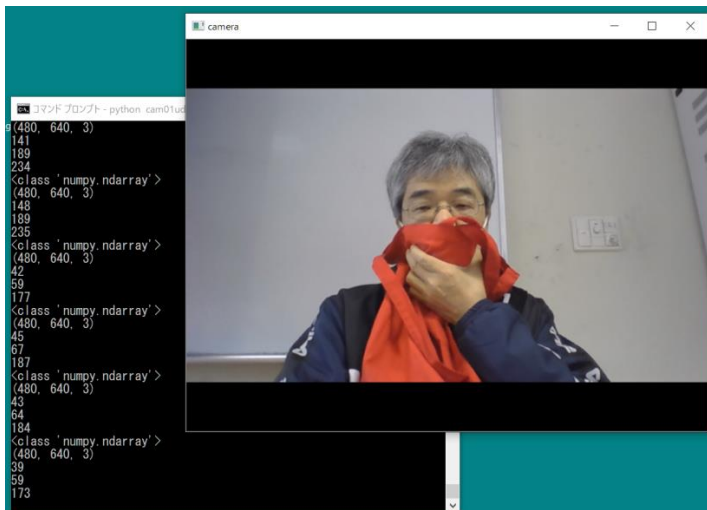


図 実行の様子

• Node-RED 側

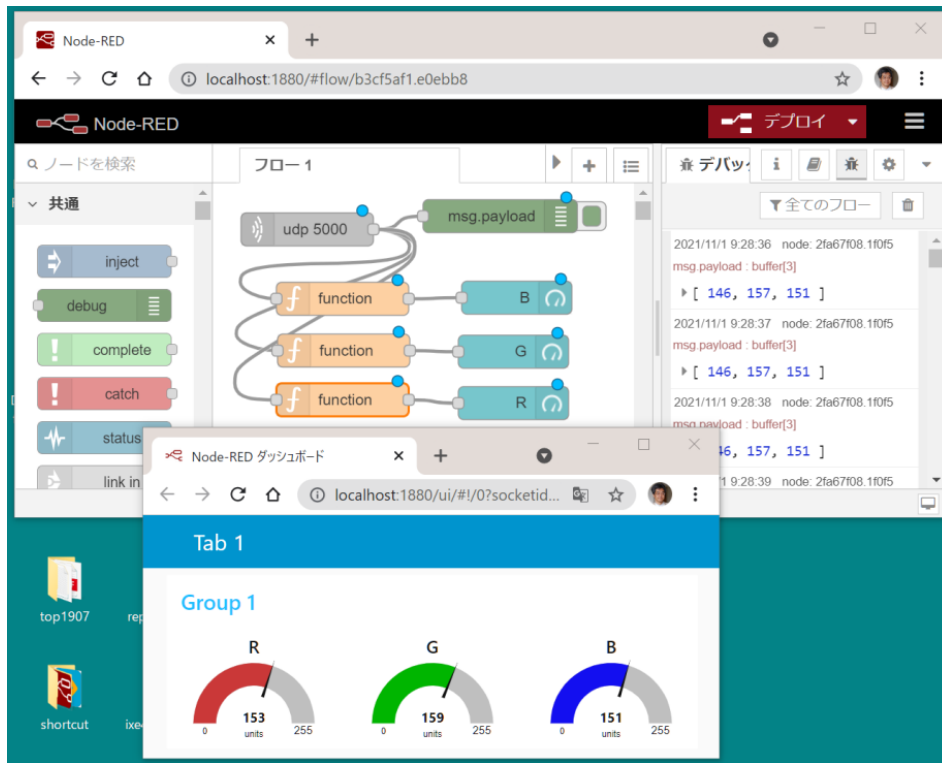


図 Node-RED 側

B (青) 用 function ノード
`msg.payload = msg.payload[0];`
`return msg;`

G(緑), R (赤) は `msg.payload[0]` の値を 1, 2 に変更する。