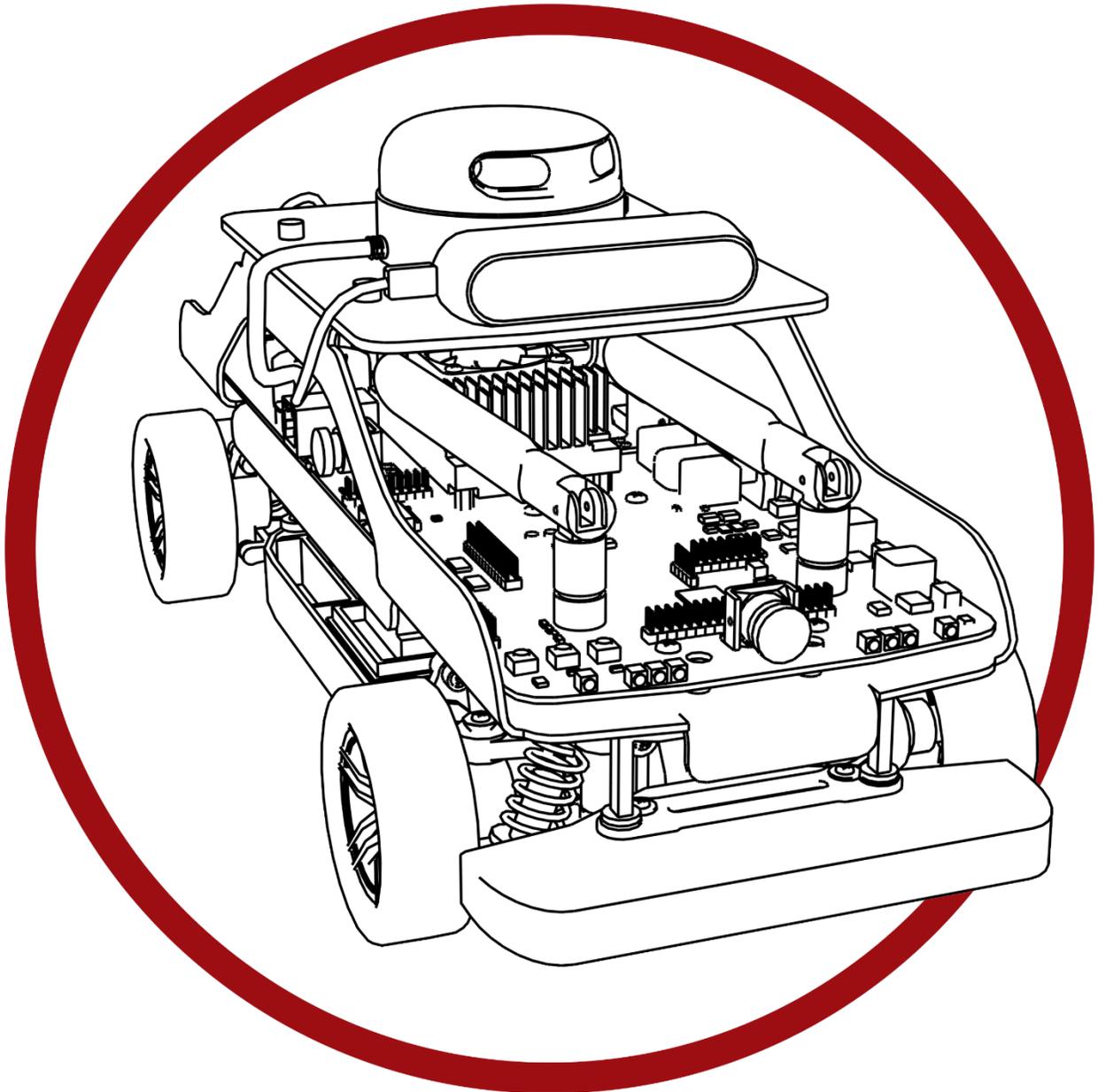


Self-Driving Car Research Studio



360 Vision – Python

V 1.1 (November 2020)

Table of Contents

I. System Description	3
II. Running the example	3
III. Details	4

I. System Description

In this example, we will capture images from the four CSI cameras at the same resolution and frame rate. These will be stitched together, and passed to a display module.

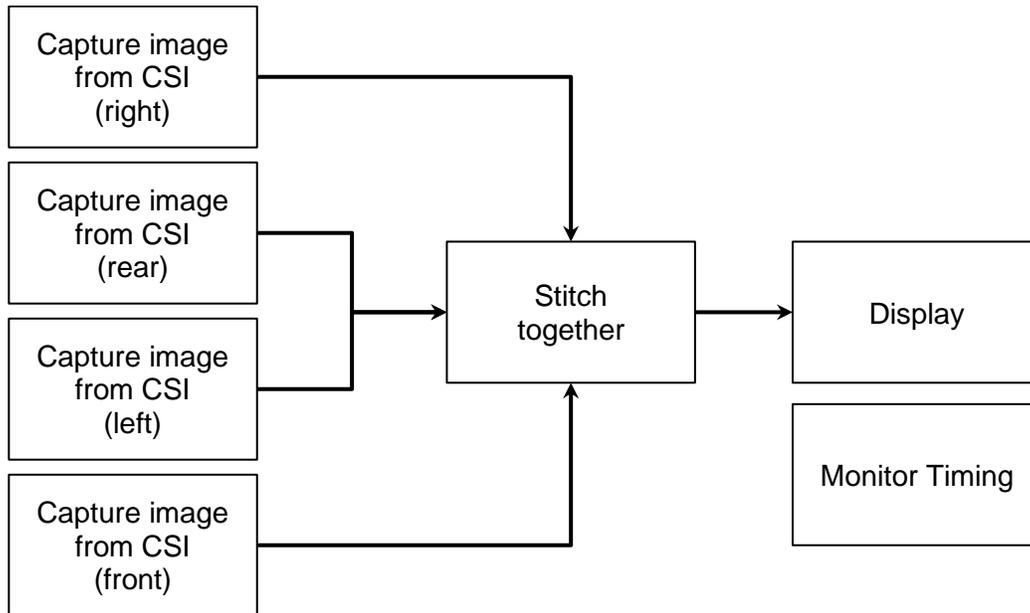


Figure 1. Component diagram

II. Running the example

Check the user guide **V - Software - Python** for details on deploying python scripts to the QCar as applications. The output in the `cv2.imshow()` function should look like Figure 2.



Figure 2. 360 view output showing images stitched together

III. Details

In this example, we move the front camera feed to the centre of the 360 degree video stream and split the rear camera stream to the left and right extremes. This aligns the driving orientation of the vehicle with the camera feed, making first-person-view driving easier. We also place black padding around the camera feeds to ease differentiating them, as seen in the snippet below.

```
# Stitch images together with black padding
horizontalBlank = np.zeros((20, 4*imageWidth+120, 3), dtype=np.uint8)
verticalBlank   = np.zeros((imageHeight, 20, 3), dtype=np.uint8)

imageBuffer360 = np.concatenate(
    ( horizontalBlank, np.concatenate
      (
        ( verticalBlank,
          myCam2.image_data[:,320:640],
          verticalBlank,
          myCam3.image_data,
          verticalBlank,
          myCam4.image_data,
          verticalBlank,
          myCam1.image_data,
          verticalBlank,
          myCam2.image_data[:,0:320],
          verticalBlank
        ), axis = 1
      ),
    horizontalBlank )
    , axis=0 )
```

The script runs in a **while** loop at 30Hz The **cv2.waitKey(msSleepTime)** will pause the loop for the time difference between the **sample time** and **computation time** (please refer to **V - Software - Python Section B. Timing** for more information.)