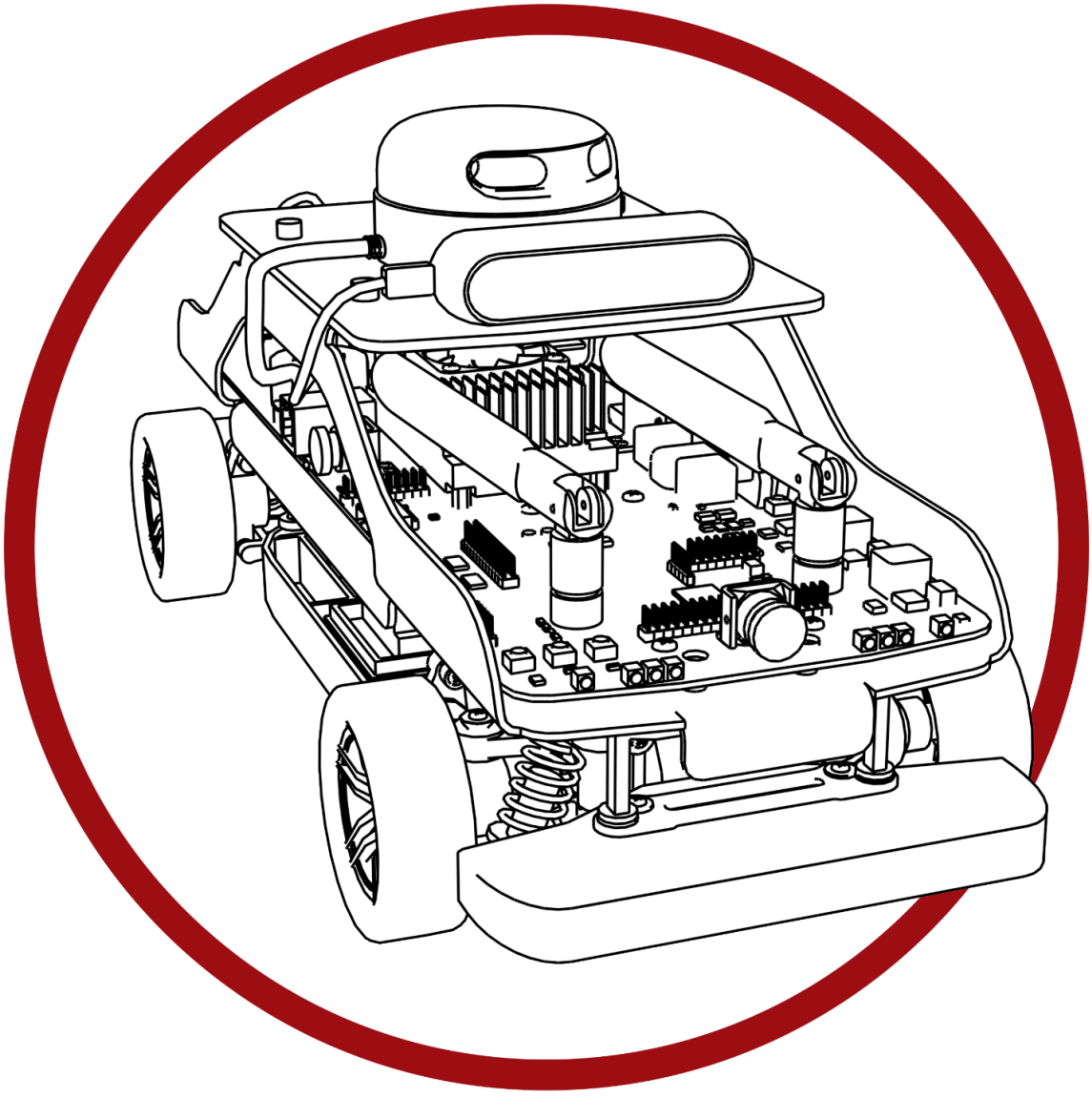
QUANSER
INNOVATE·EDUCATE

Self-Driving Car Research Studio



360 Vision - Simulink

V 1.1 (November 2020)

Table of Contents

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.

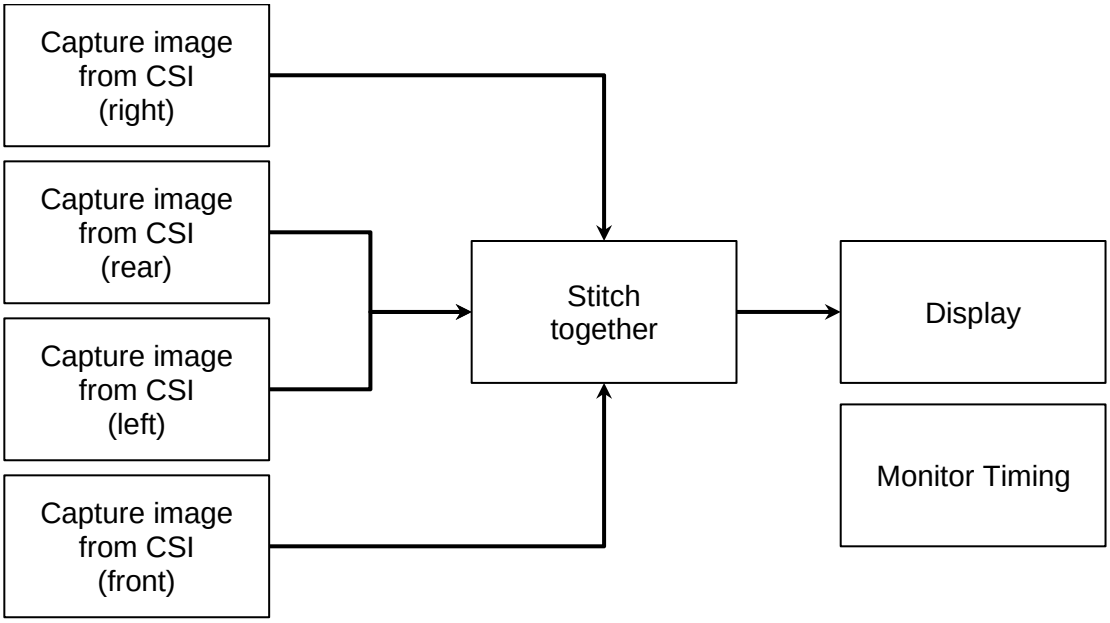


Figure 1. Component diagram

In addition, a timing module will be monitoring the entire application's performance. The Simulink implementation is displayed in Figure 2 below.

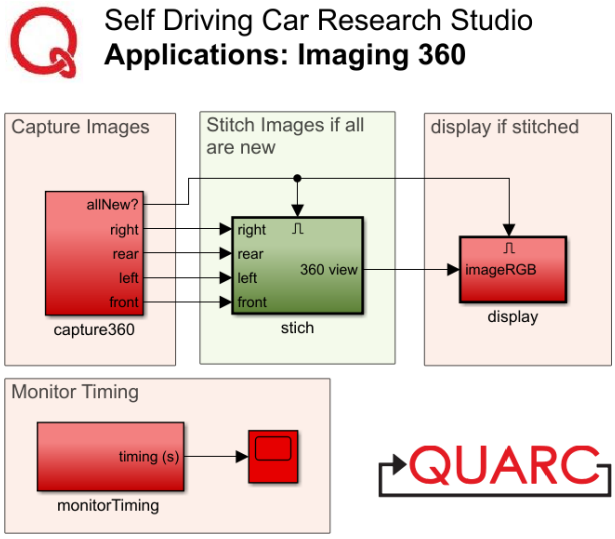


Figure 2. Simulink implementation of 360 Vision

II. Running the example

Check the user guide **IV - Software - Simulink** for details on deploying Simulink models to the QCar as applications. The output in the **Video Display** block should look like Figure 3. (The order of stitching is **Right, Rear, Left,** and **Front.)**

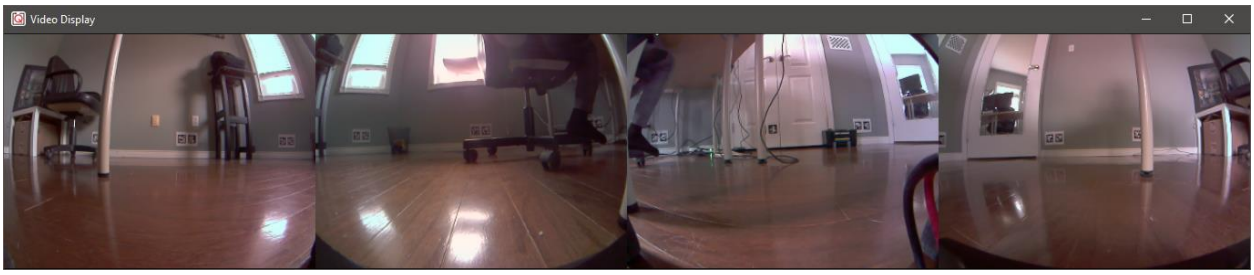


Figure 3. 360 view output showing images stitched together

III. Details

The implementation has a few important elements that help boost performance. The model is deployed at 30 Hz with each camera streaming a 640 x 480 RGB (3 channels) unsigned integer image (8 bits or 1 byte per data sample). The data rate can be estimated as,

$$30\ \frac{steps}{s} \times (640 \times 480 \times 3)\ \frac{samples}{image} \times 1\ \frac{byte}{sample} \times 4\ \frac{images}{step}$$

This ends up as $105.47\ Mbps$ of data streamed over Wi-Fi. However, asynchronous image acquisition between the four capture loops might yield cases when at least 1 of the 4 images might not be new, and this step can be skipped.

In the implementation provided, the images are stitched together only when all the images are new. The **Video Capture** block in the **capture360** module also outputs a **new** signal that is high (**1**) when the image is new. The four **new** signals are passed into a logical AND, which is then used to stitch images and display them, improving performance.

Note that a simple matrix concatenate method is used here to stitch the images together side by side. Alternatively, you may consider debarreling the images first, and using feature matching to remove translational and rotational discrepancies in the placements of the 4 cameras.