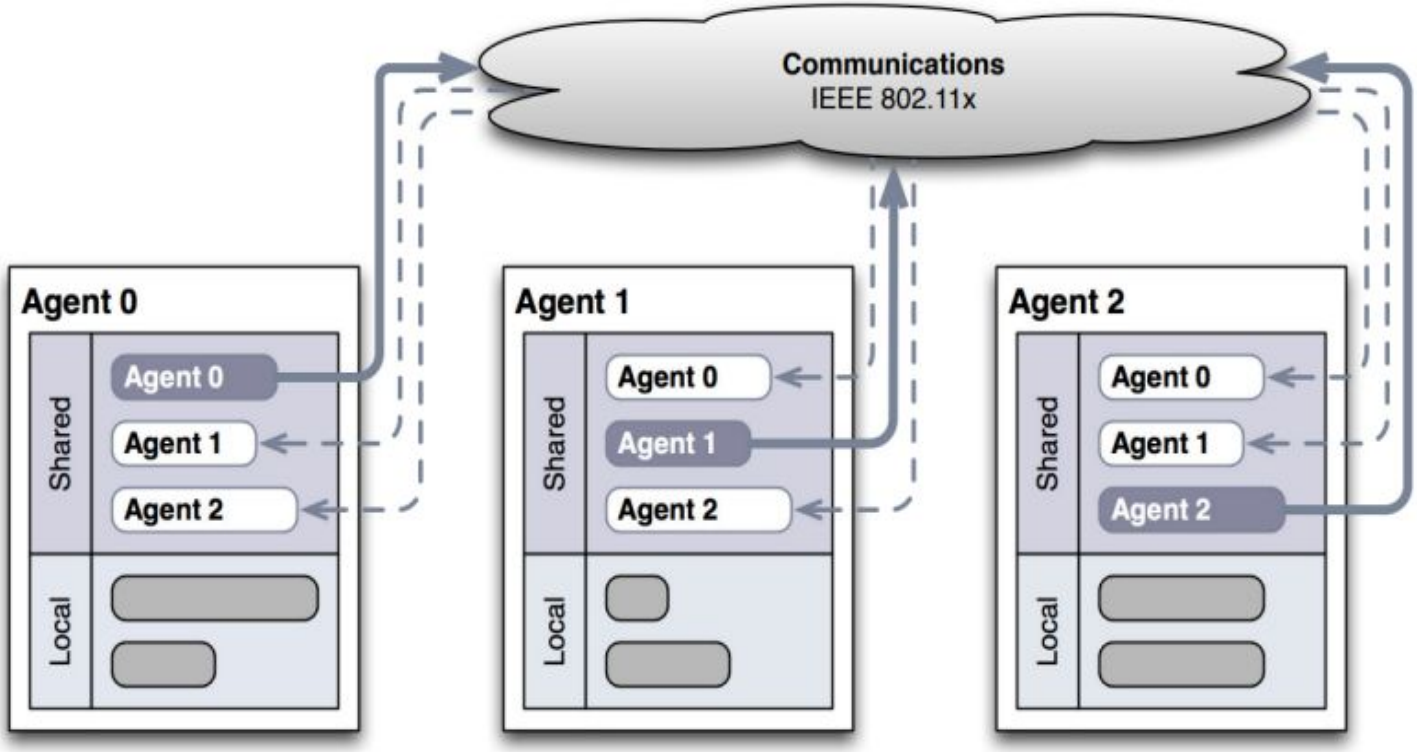
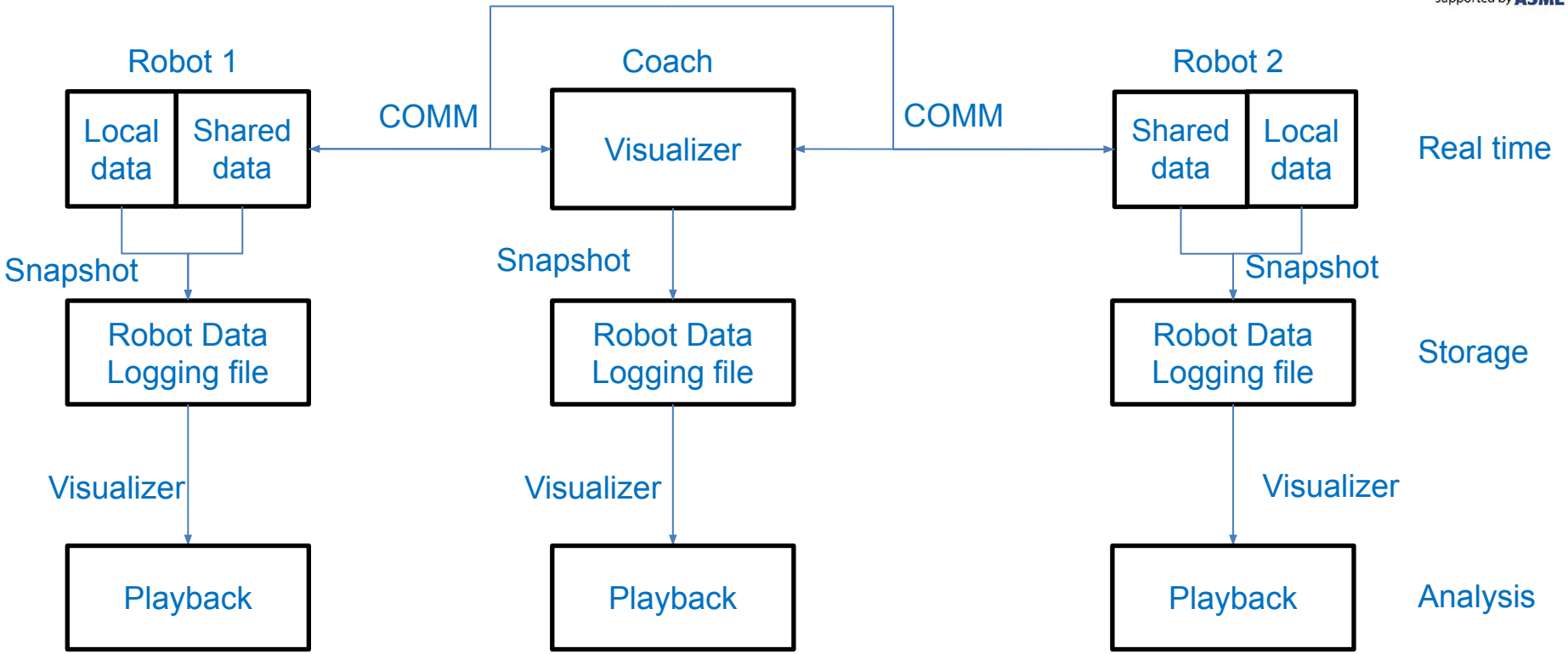
Scientific and Engineering presentation

Real Time DataBase

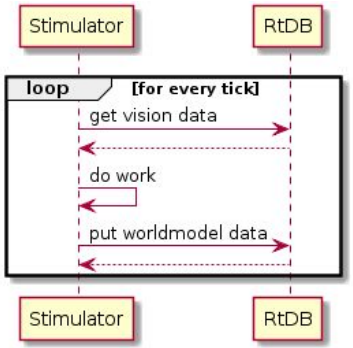
FALCONS
supported by ASML

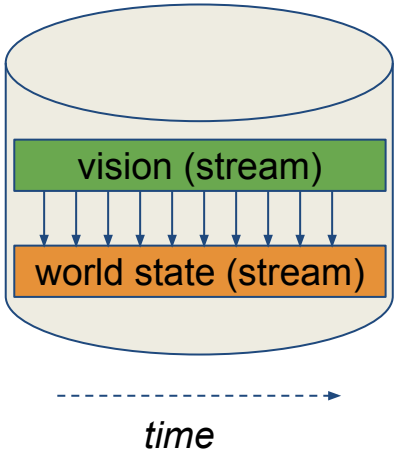


Improvement logging



Improvement Stimulation



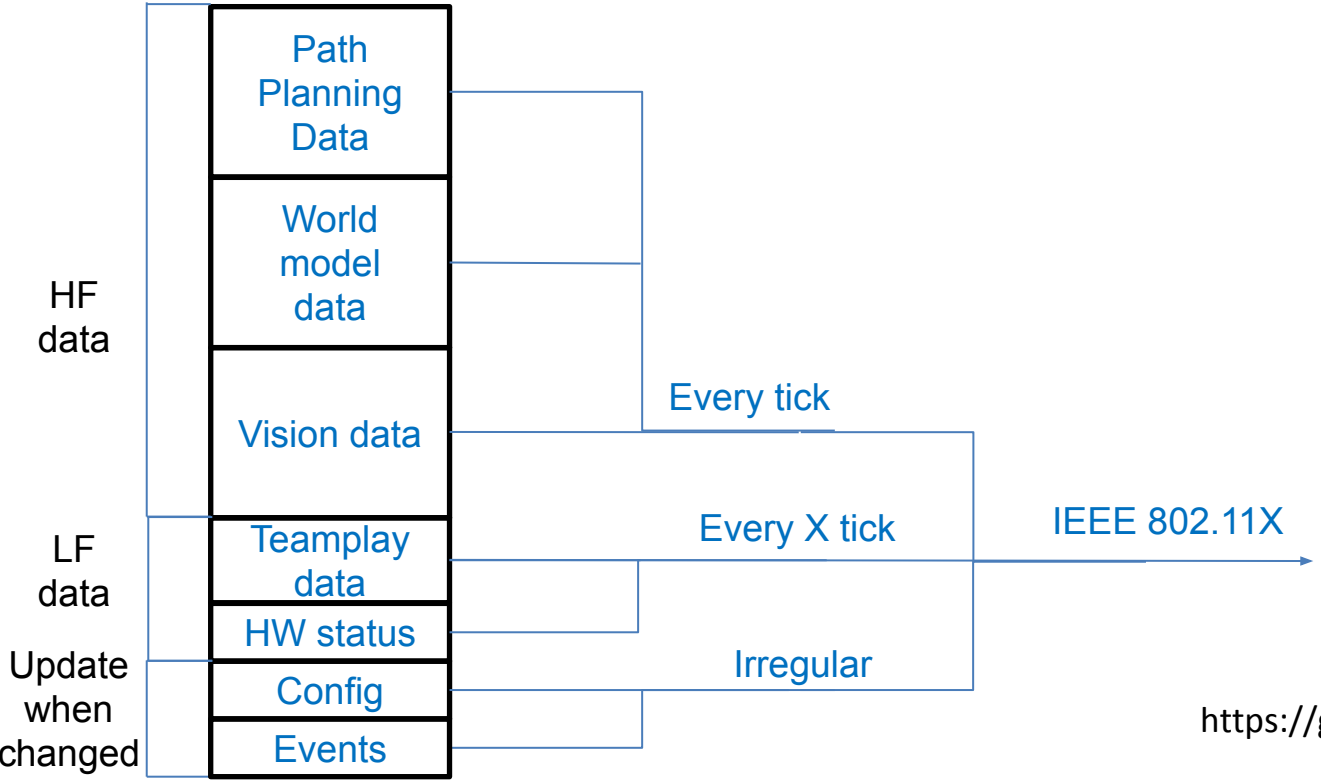


Reference Robot Data Logging file

SW update

New Robot Data Logging file

Improvement Comm2



Reduction by a factor of two achieved, more possible

https://github.com/Falcons-Robocup

Vision

FALCONS
supported by ASML

Synchronizing Asynchronous Cameras

Vision Introduction

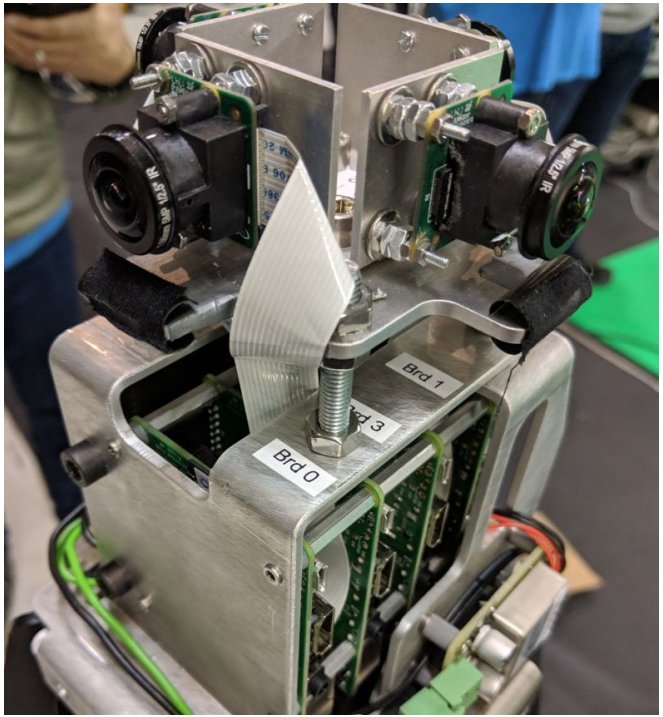
4 Raspberry pi cameras (+ 4 Raspberry pi boards)

- low budget (mobile phone market)
- rolling shutter
- **no** synchronization input
- camera access through i2c
- camera driver (GPU) **closed source**

Used for

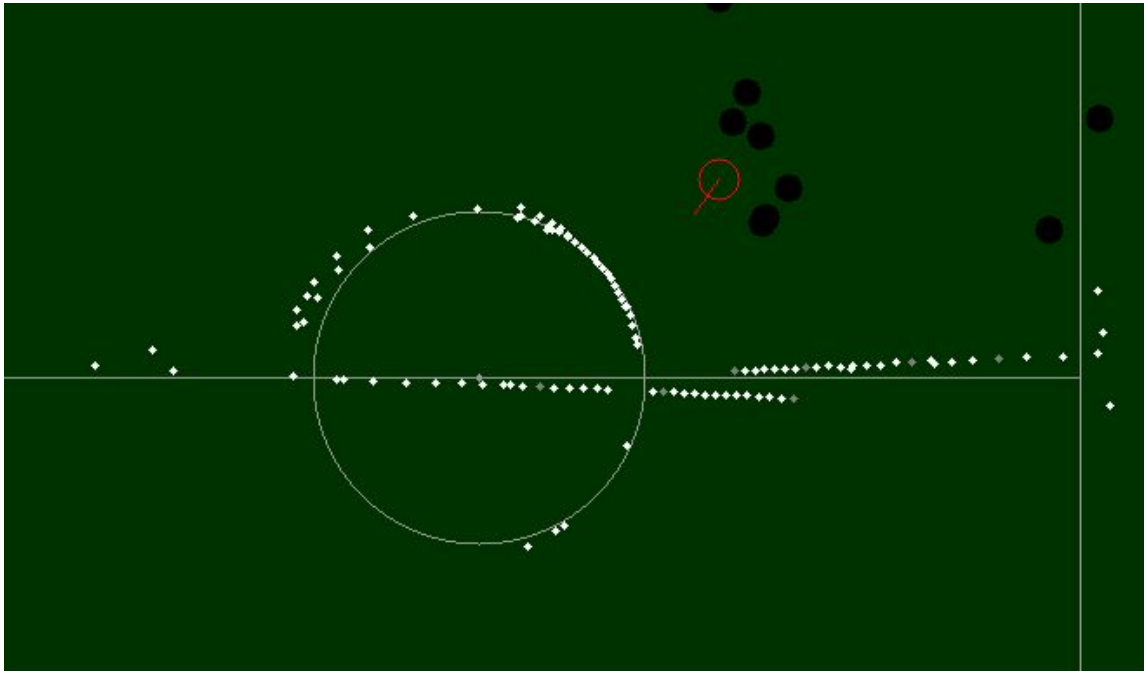
- **localization**
- ball detection
- obstacle detection

Localization requires information from **all** 4 cameras



Vision Synchronization Problem Statement

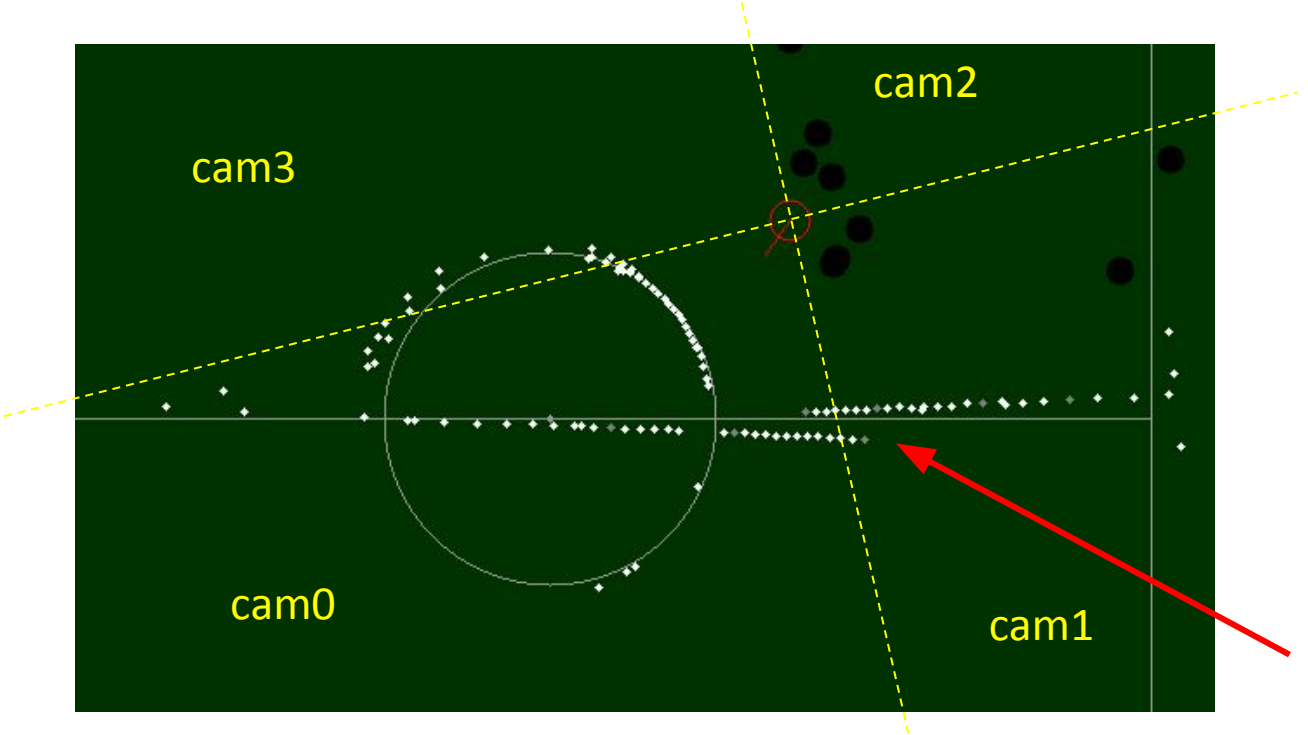
Image displacement when captured on different time when **moving**



top view

Vision Problem statement

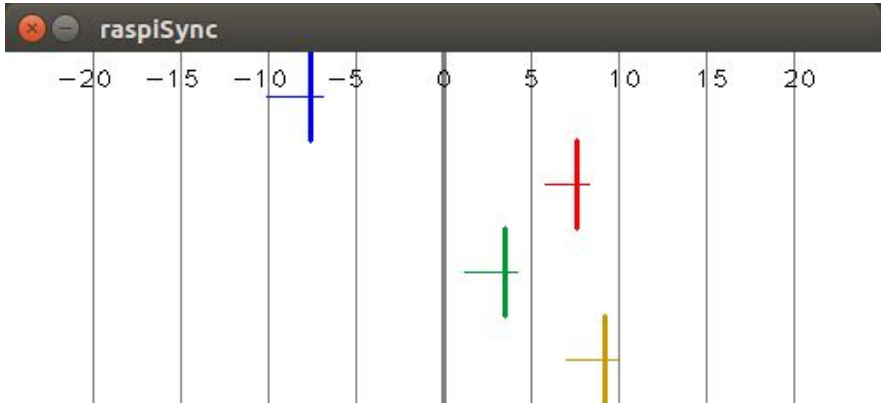
Image displacement when captured on different time when **moving**



Vision Synchronization Numbers

driving 5 m/s (18km/h)
acceptable error 6.5cm (line width 13cm)
acceptable delta time 0.065/5 = 13ms

rotating 1 turn in 2 second (180 deg/sec)
acceptable angle error 2 deg
acceptable delta time 2/180 ~= 11 ms



rolling shutter
- 10 ms (for region of interest)

camera captures at 40 frames per second (FPS)
- capture delta time between cameras maximal 25 ms

Vision Synchronization Target

FALCONS
supported by ASML

Synchronize capture delta time from 25 ms to less than 5 ms

Vision Synchronization Scenarios

FALCONS
supported by ASML

higher FPS

NO: limited by Raspberry PI

start/stop cameras in controlled way

NO: video stream interrupted / difficult to perform on correct time

change camera pixel clock

NO: pixel clock control granularity too large / limitations Raspberry Pi

slightly modify amount of pixels per line

YES: software control

slightly modify amount of lines per frame

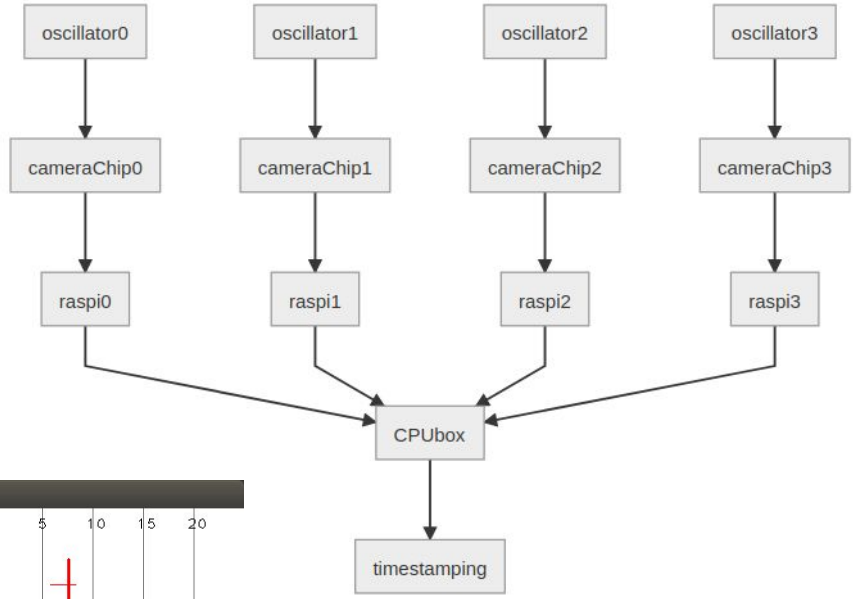
YES: software control (but larger granularity than pixels)

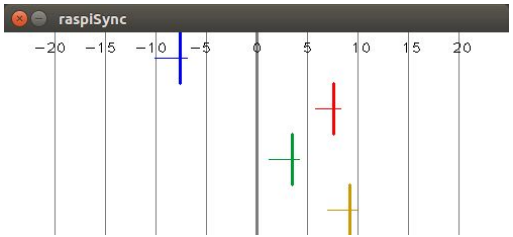
*FPS = constant * oscillator / (lines * pixels)*

Vision Synchronization Implementation

camera 0 = time reference
send start of frame to CPU box

- received after camera 0
  - remove few pixels (for short while)
- received before camera 0
  - add few pixels (for short while)
- received nearly at the same time
  - use default pixel size

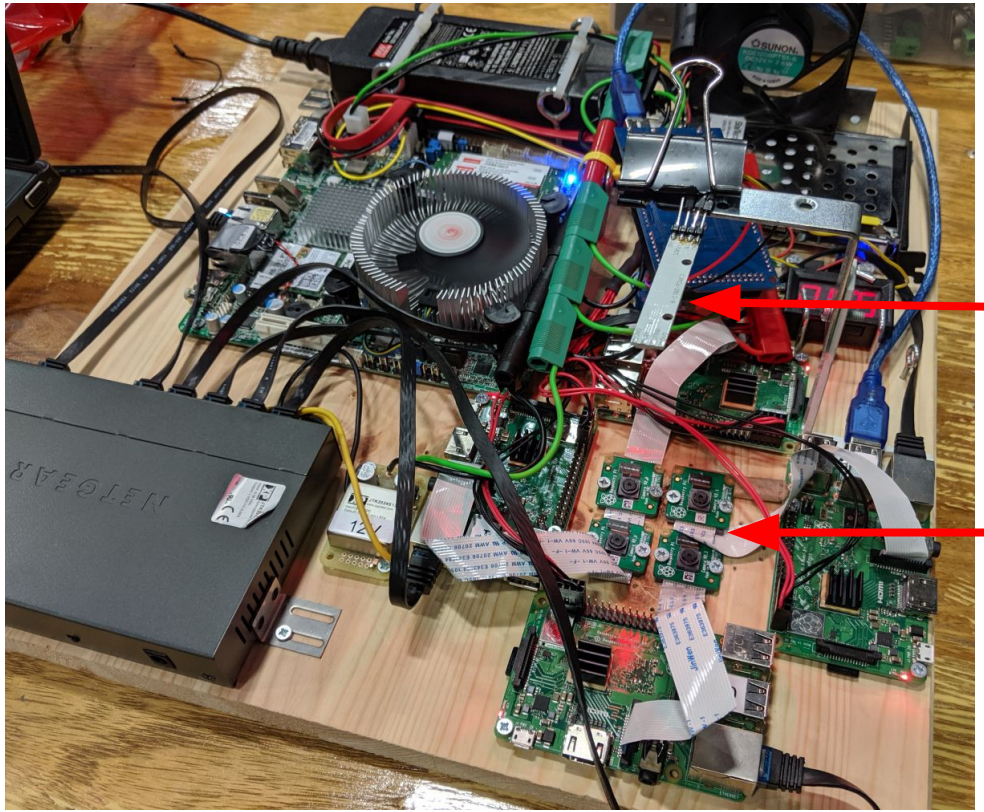




make use of i2c to access camera chip

“The Plank” Synchronization Test Setup

FALCONS
supported by ASML

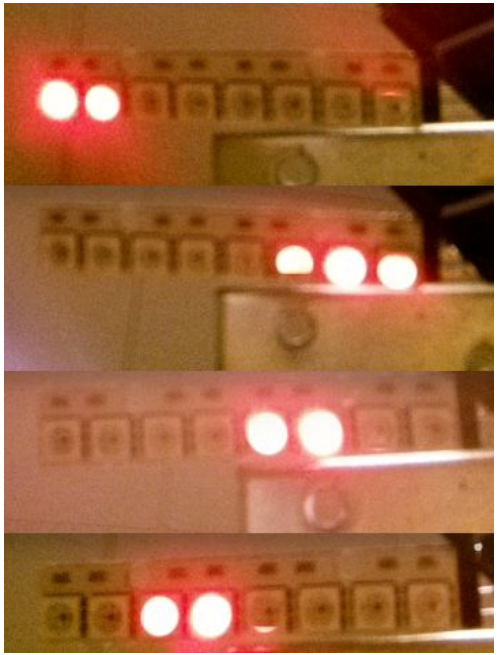


blinking lightstrip
8 LEDs * 40Hz = 320Hz

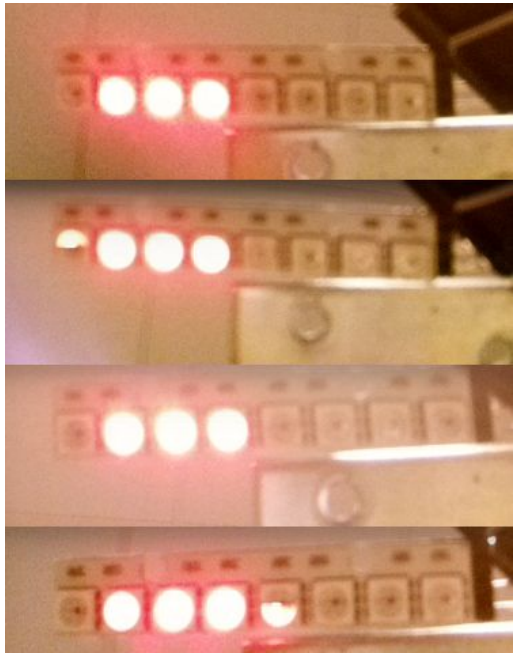
4 x Camera

Vision Result

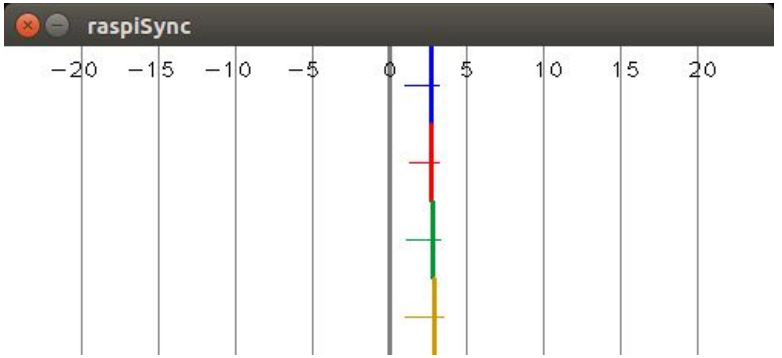
led blinking lightstrip at n * fps seen by all 4 cameras



before



after



Vision Synchronization Roadmap

FALCONS
supported by ASML

Increase frame rate
- lower impact rolling shutter

Synchronize cameras over all robots
- moving ball triangulation

Keeper extension actuators