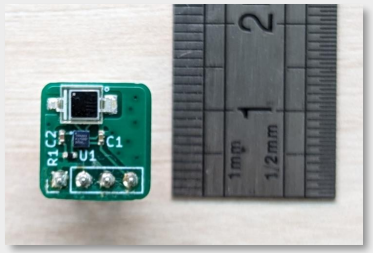
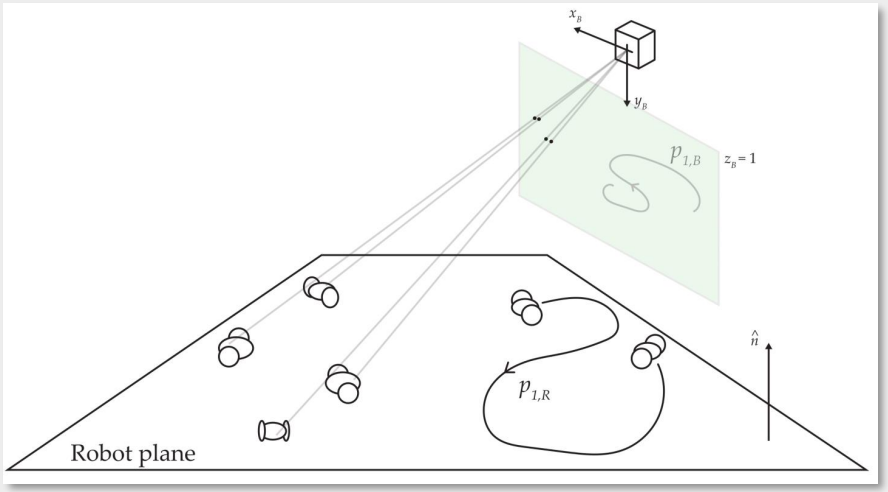
Repurposing Valve's SteamVR 2.0 for an Open-Source, Low-Cost Motion Capture System

Presented by: Said Alvarado-Marin
Supervised by: Dr. Filip Maksimovic
Dr. Thomas Watteyne

System Overview



Lighthouse



Robot Localization

Accuracy

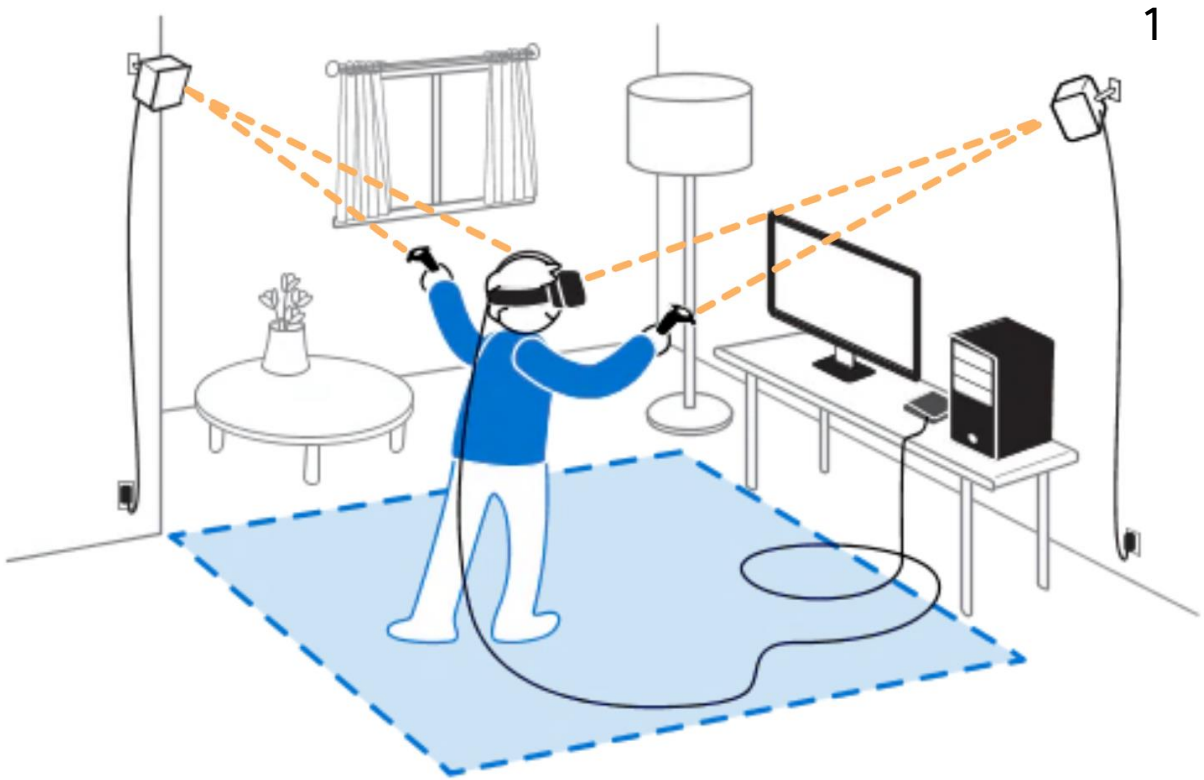
6 mm

Precision

0.15 mm

Results

What is a Lighthouse ?



1 - https://www.vive.com/nz/support/vive-pro-hmd/category_howto/verifying-your-setup.html

Why the Lighthouse ?

	Refresh Rate	Accuracy	Processing	Price
	100Hz	5mm	Decentralized	$350
VICON	500Hz	~0.1mm [1]	Centralized	> $10K

1 - Windolf M, Götzen N, Morlock M. Systematic accuracy and precision analysis of video motion capturing systems--exemplified on the Vicon-460 system. J Biomech. 2008

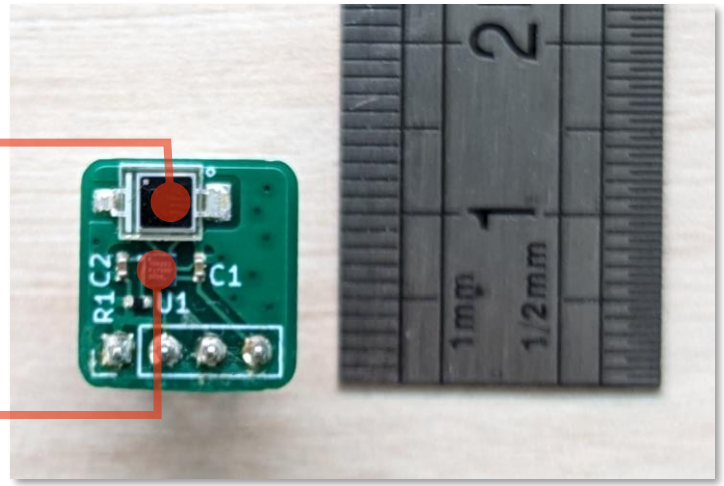
How does the Lighthouse system work ?

Basestations and Sensors

1

2

Basestation: $160

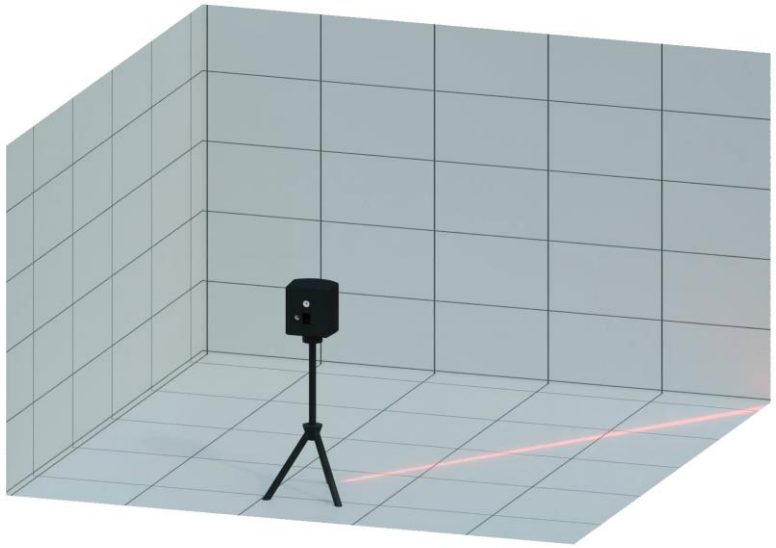


- Photo-diode: $1.3
- Processing IC (TS4231) : $2.2

Total: $3.5

1 - https://www.roadtovr.com/next-gen-lighthouse-base-station-bring-rapid-cost-reductions/
2 - https://www.roadtovr.com/valves-lighthouse-base-station-action-inner-workings-explained/

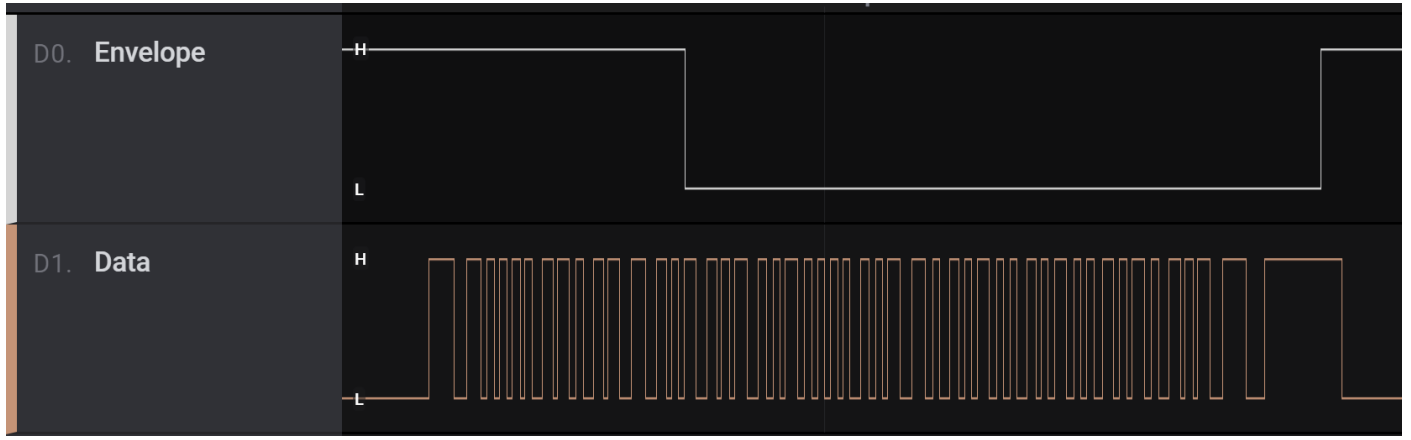
Light-beam Pattern

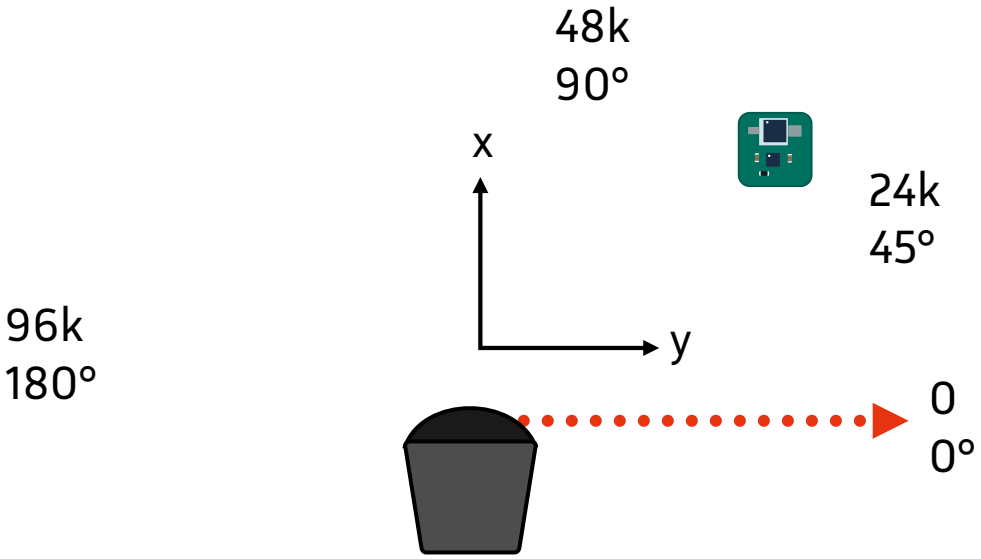


Lighthouse V1

Lighthouse V2

Receiving the Laser

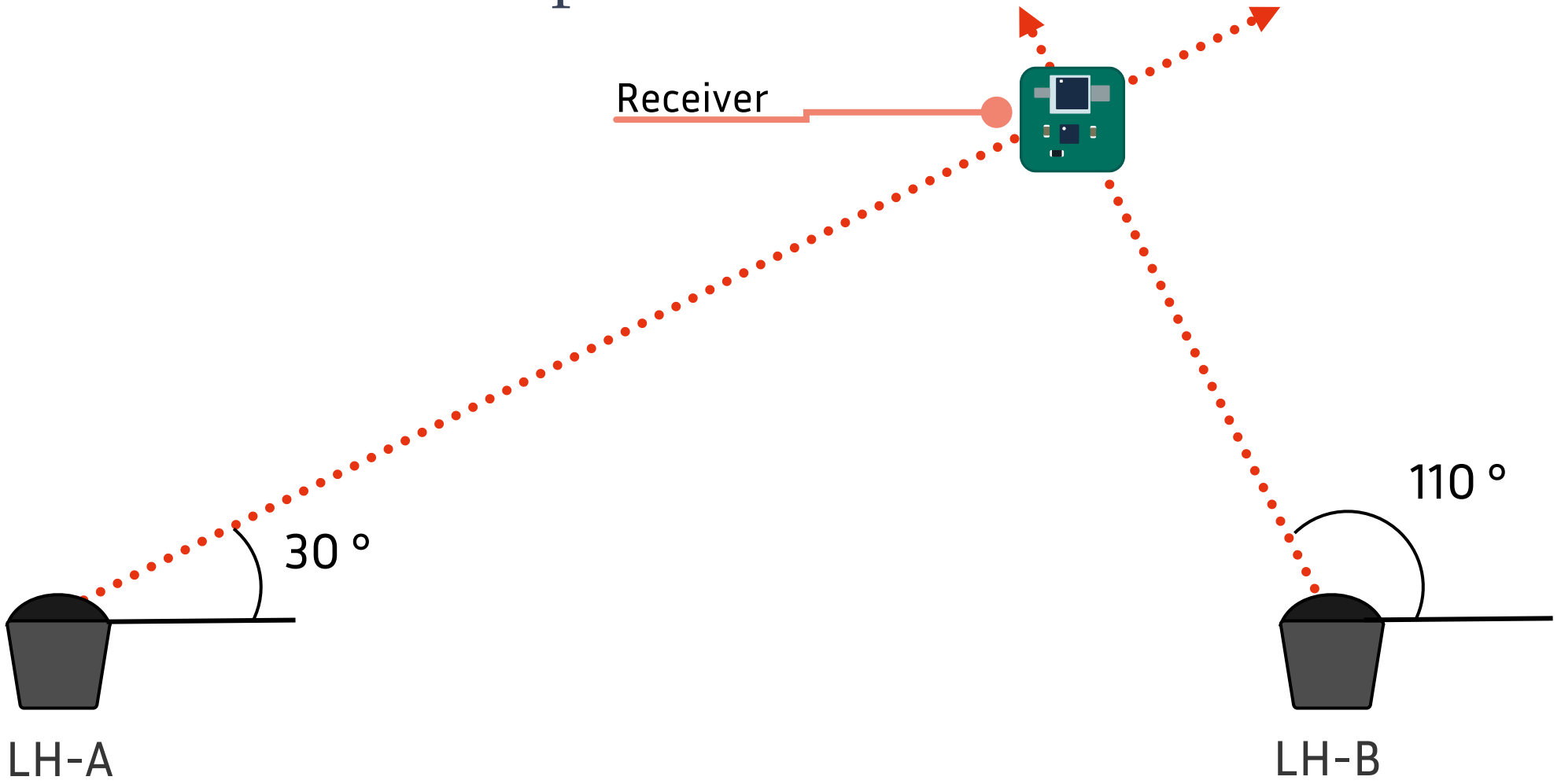




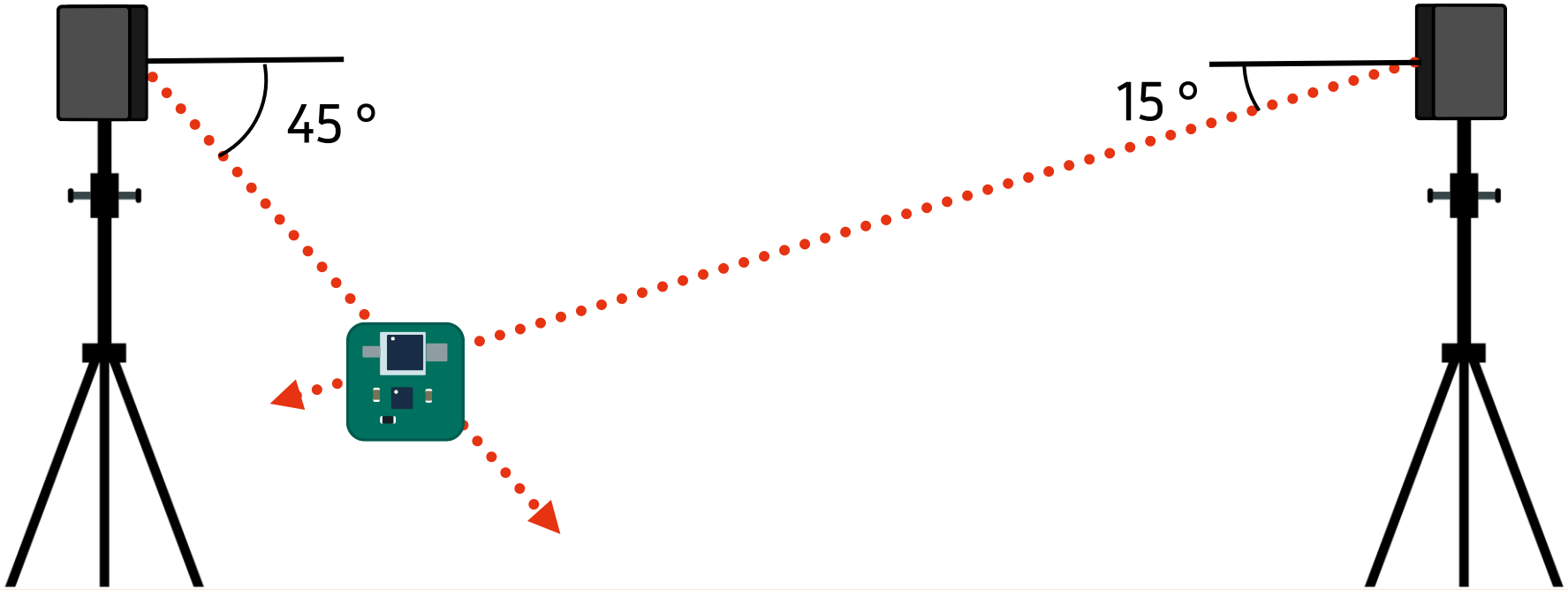
- 12Mhz Differential Manchester Encoding
- **Known** pseudo-random sequence

Sample the **Data** line at 32MHz

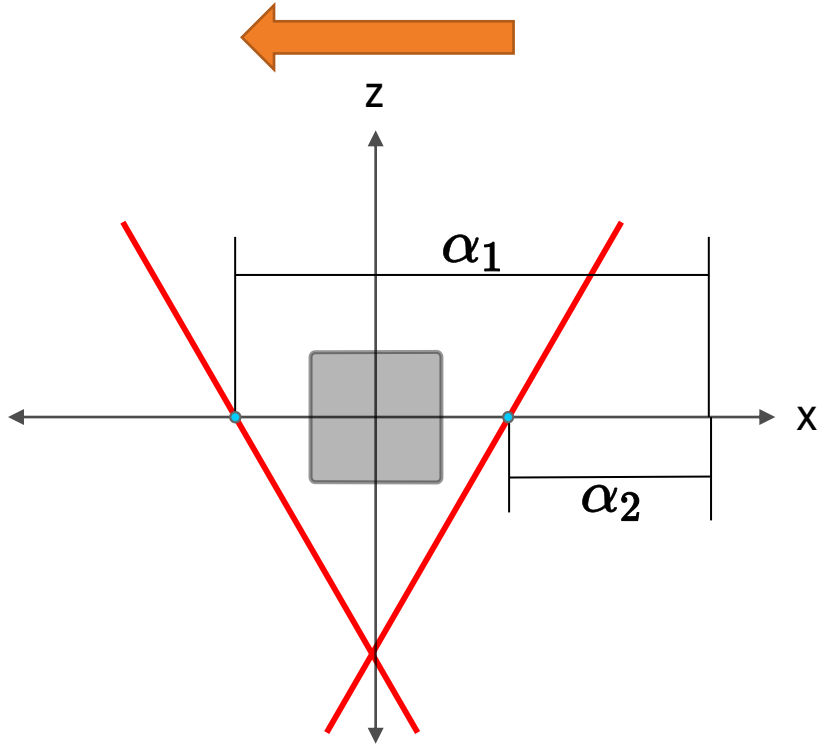
2D Localization Example



Extension to 3D



Angled Planes



Azimuth

$$\theta = \frac{\alpha_1 + \alpha_2}{2}$$

Elevation

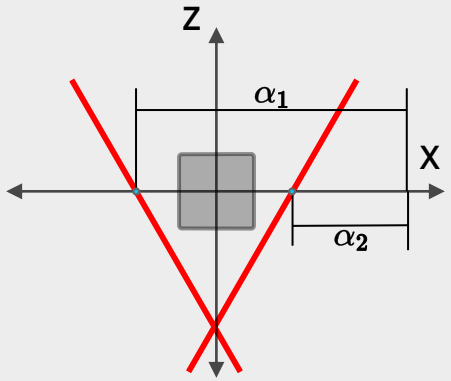
$$\phi = tan^{-1}\left(\frac{sin\left(\left(\alpha_2 - \alpha_1\right)/2 - \pi/3\right)}{tan\left(\pi/6\right) cos\left(\left(\alpha_2 + \alpha_1\right)/2\right)}\right)$$

Lighthouse Summary

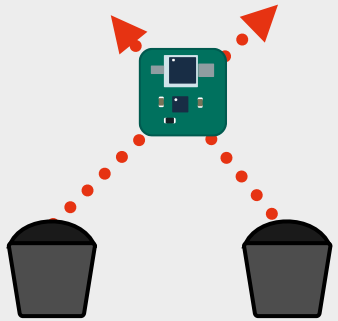
LH2 Sweeps Lasers

Receive the Laser

Compute Azimuth and Elevation

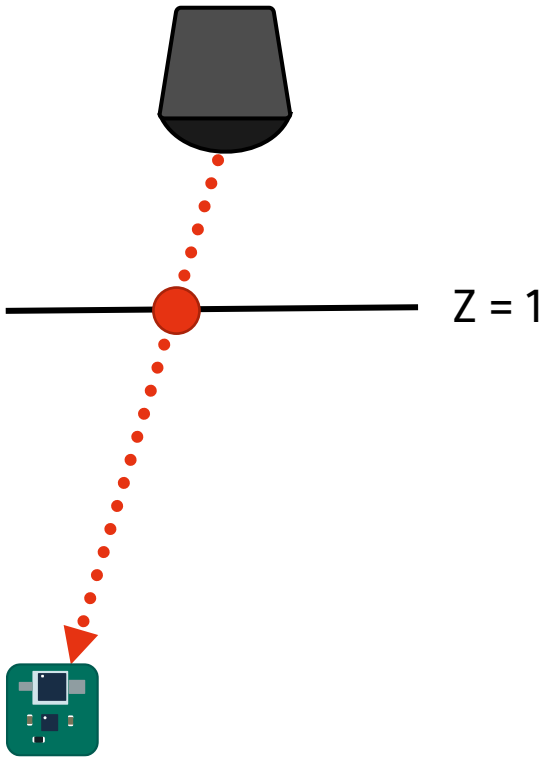


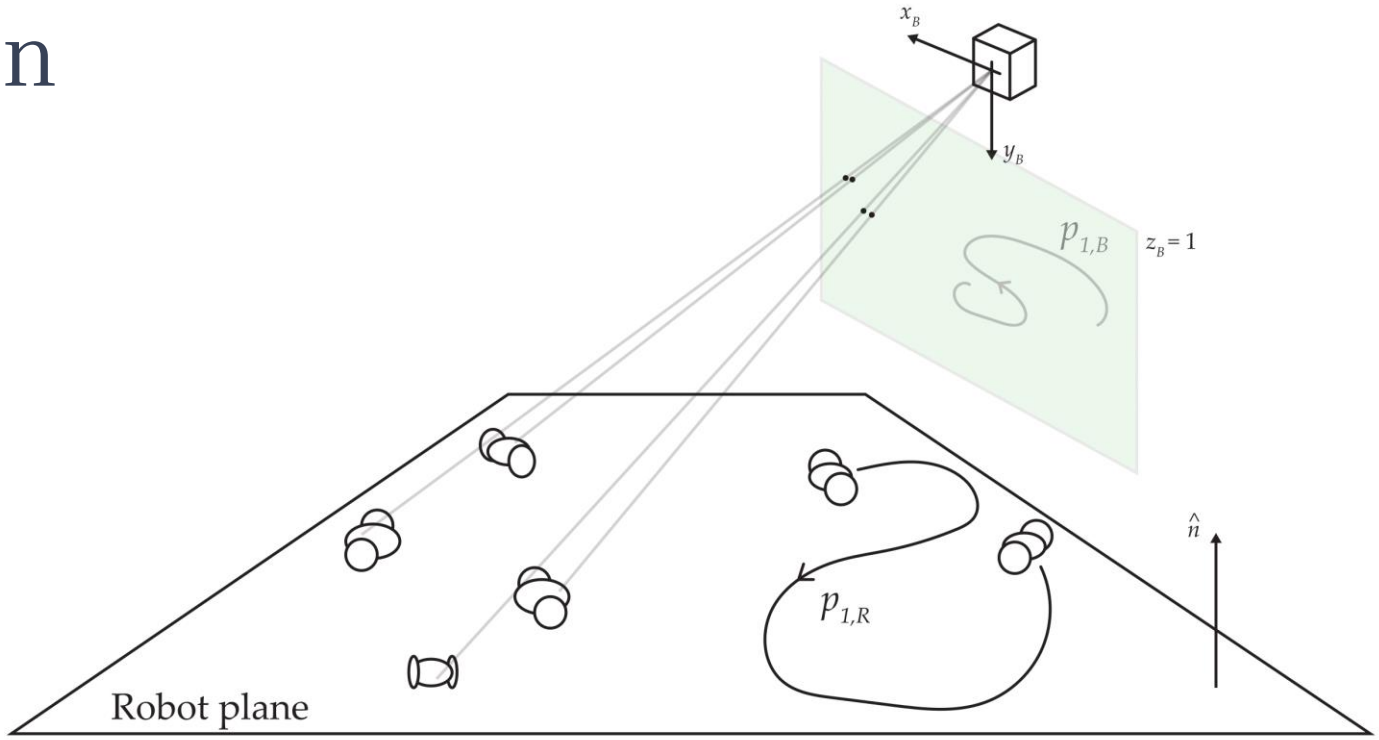
Triangulate



Localization

Camera Assumption





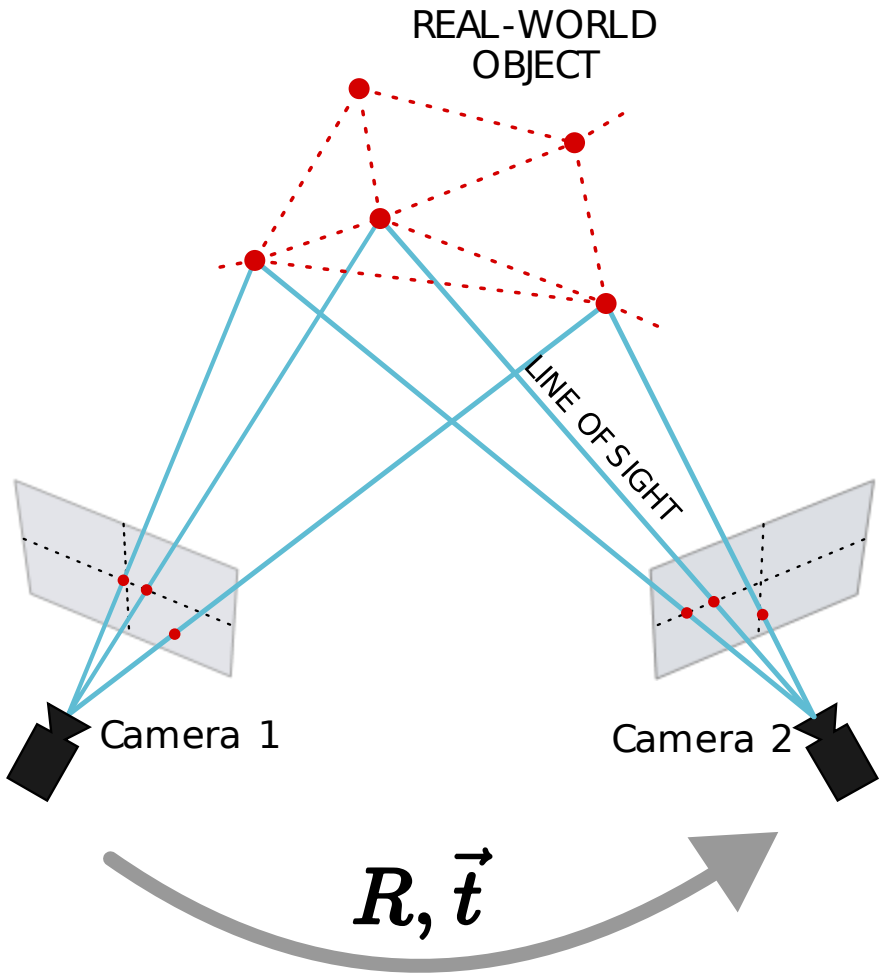
$$x = -tan(\theta)$$

$$y = -tan(\phi)/cos(\theta)$$

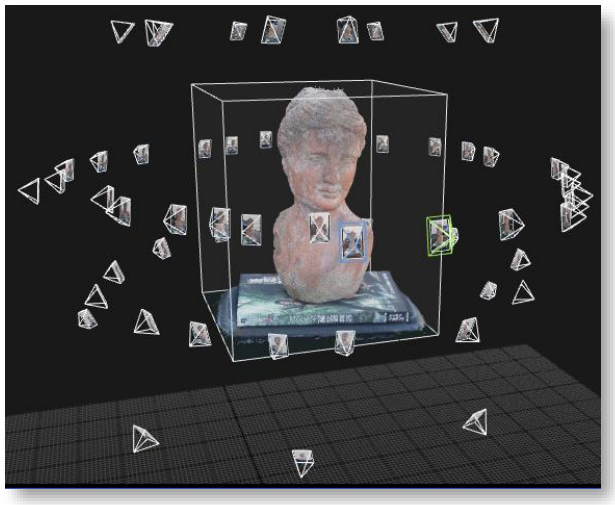
$$z = 1$$

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Stereo Vision

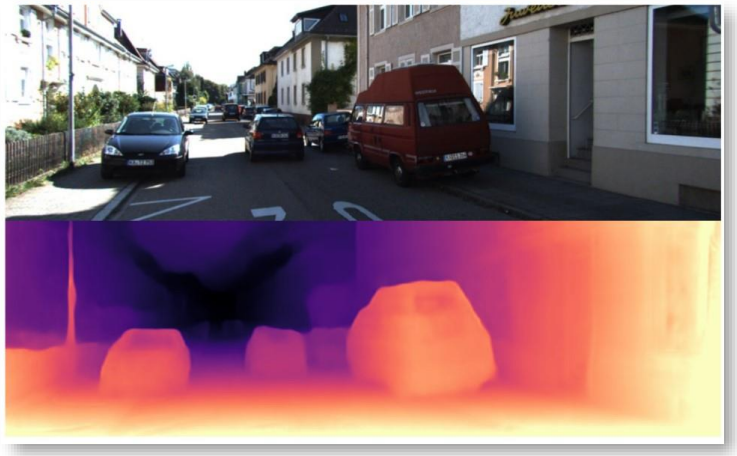


Photogammetry



1

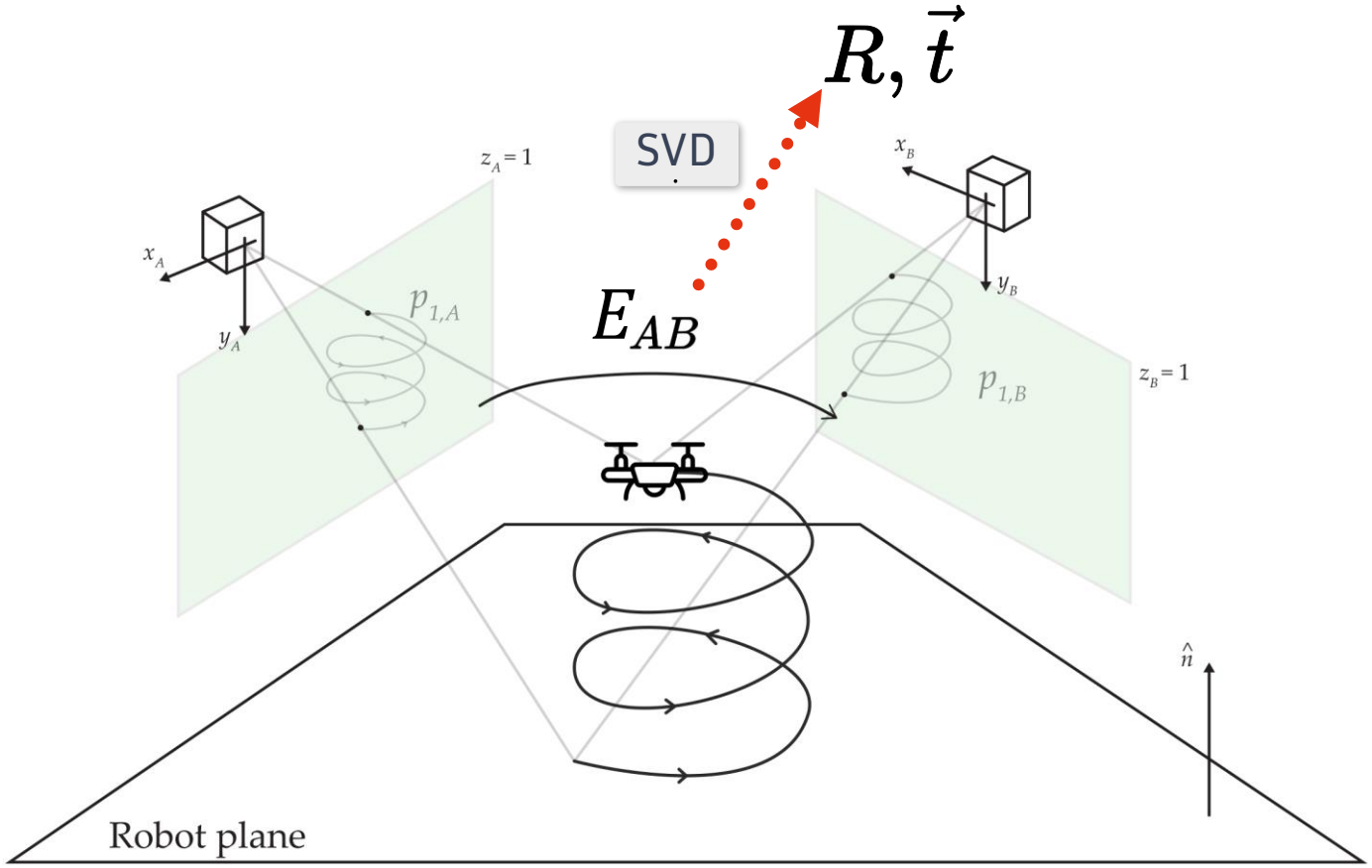
Depth-Estimation



2

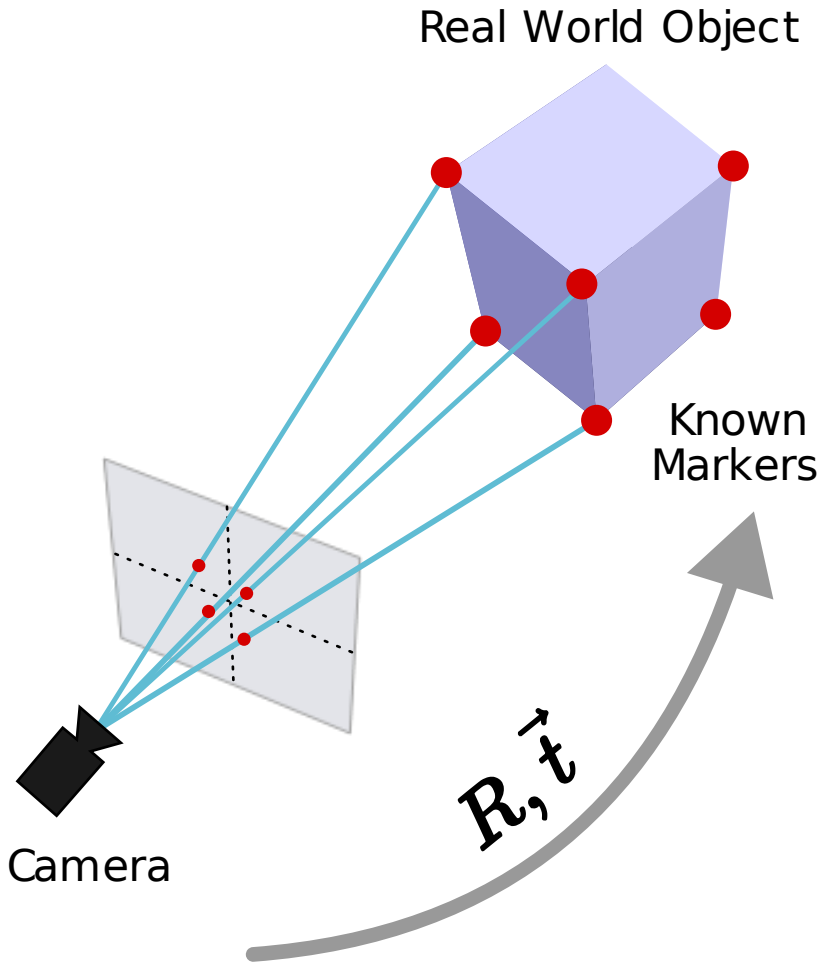
1 - https://3dscanexpert.com/realitycapture-photogrammetry-software-review/
2 – C.Godarg et. al, "Digging into Self-Supervised Monocular Depth Prediction ", 2019

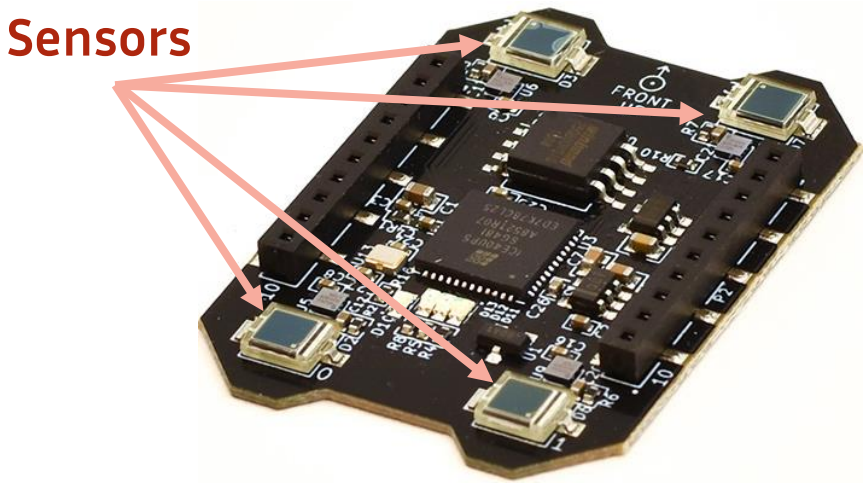
3D Essential Matrix



- Essential Matrix
- Only for **NON-Coplanar** Points
- **7** corresponding Points
- OpenCV function:
  - `cv2.findEssentialMat()`

Perspective N-Points





- 4 Known Correspondence Points
- NO need to be coplanar
- OpenCV function:
  - `cv2.solvePnP()`

How well does this work?

Single Sensor Tracking [1]

Metric	Measurement
Mean Euclidean Error	**6.2 mm**
Std. Dev.	9.3 mm

4 Sensors Tracking [2]

Metric	Measurement
Mean Euclidean Error	**10.4 mm**
Std. Dev.	9.9 mm

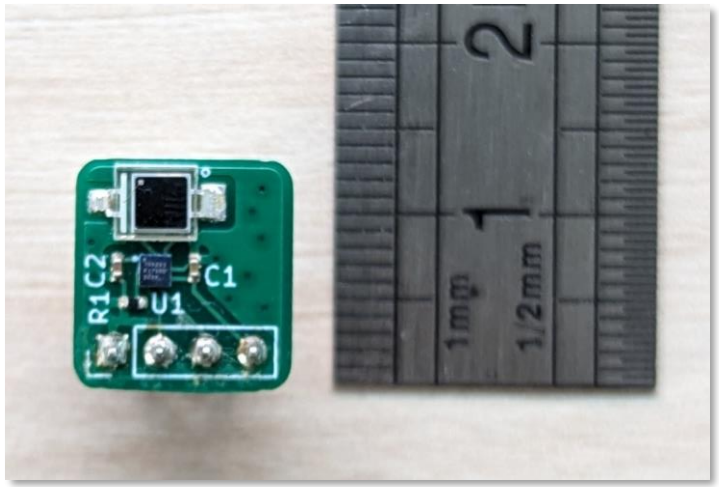
Static Precision [1]

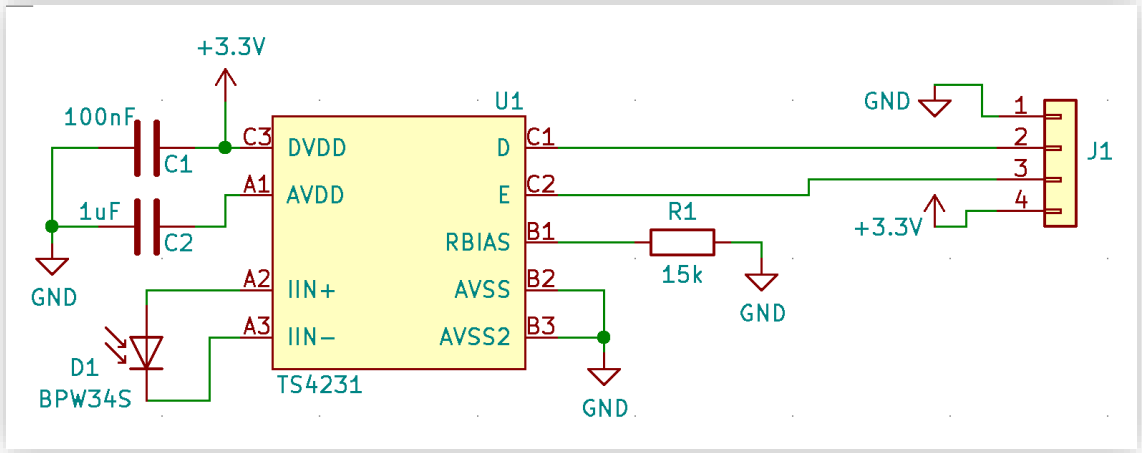
Axis	Std. Dev.
X	0.125 mm
Y	0.243 mm
Z	0.137 mm

1 - Alvarado-Marin, Said, et al. "Lighthouse Localization of Miniature Wireless Robots." *IEEE Robotics and Automation Letters* (2024).
2 - Taffanel, Arnaud, et al. "Lighthouse positioning system: dataset, accuracy, and precision for UAV research." *arXiv preprint arXiv:2104.11523* (2021).

Lighthouse V2 in Practice

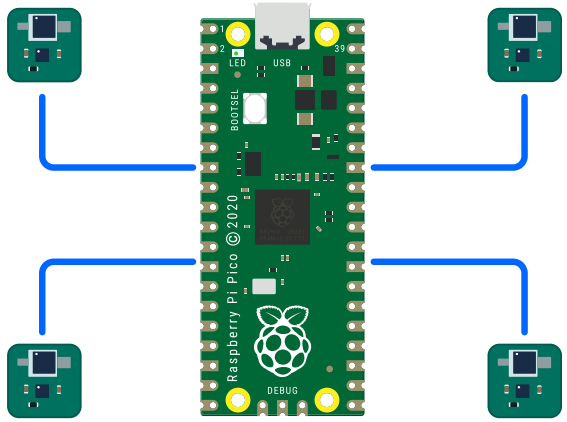
TS4231 Breakout Board

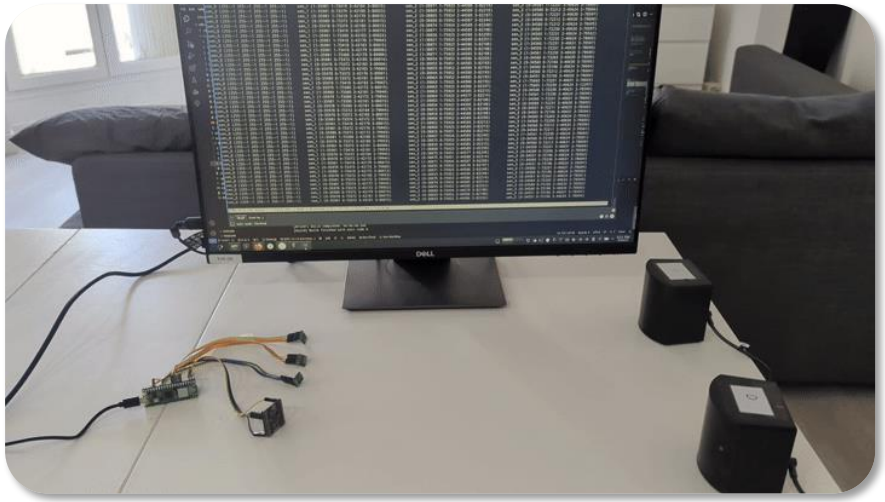




https://github.com/DotBots/TS4231-breakout-board

LH2 Decoder Implementation

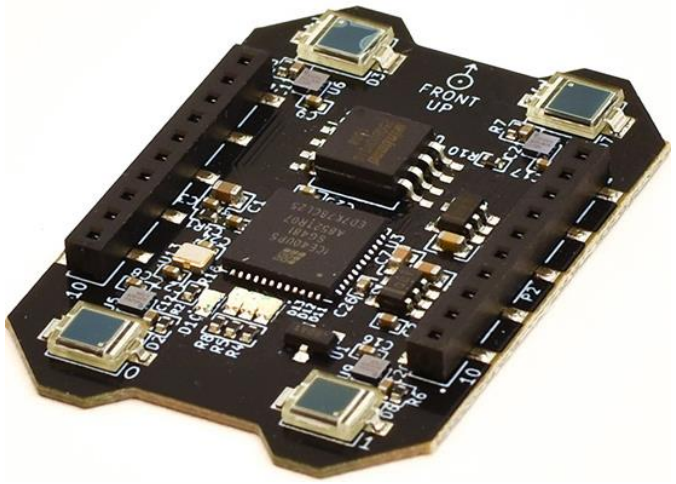




Raspberry Pi Pico 1/2

nRF52 & nRF53

Bitcraze – LH2 deck



https://store.bitcraze.io/collections/decks/products/lighthouse-positioning-deck

Thank you!