BAM days 19 -21 Oct.

bitcraze Awesome Meetup Days

Advanced Lighthouse Usage Workshop

Kimberly (Bitcraze)

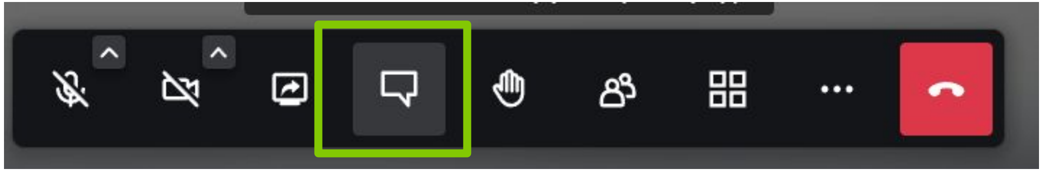
BAM days

19 October 2021

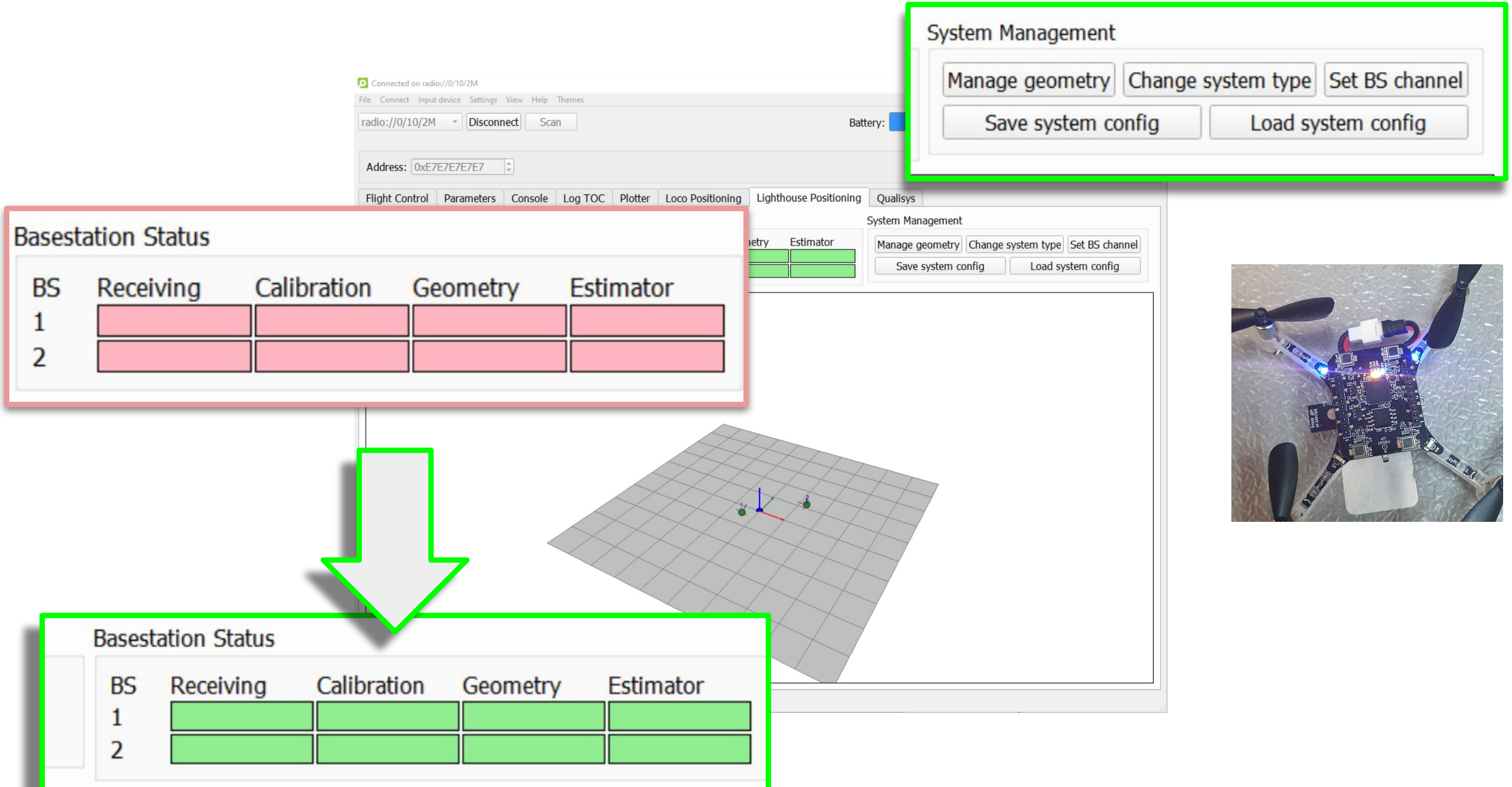
BAM days 19 -21 Oct.
bitcraze Awesome Meetup Days

Advanced Lighthouse Usage Workshop

BAM days
19 October 2021



Lighthouse Tab CFclient



Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟥	🟥	🟥	🟥
2	🟥	🟥	🟥	🟥

Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟩
2	🟩	🟩	🟩	🟩

Content

- Lighthouse Base station mechanics
- Lightsweep decoding
- Calibration Model
- BS Geometry estimation
- On-board Pose Estimation
- Extra
  - Geometry file saving (swarms)
  - More than 2 base stations

Prerequisites:

- Positioning system workshop bamdays
- [Getting started with Lighthouse Positioning Tutorial]()

Crazyflie and positioning

Marcus Eliasson @ Bitcraze

BAM days
19 October 2021

Getting started with the Lighthouse system

Home / Documentation / Tutorials / Getting started with the Lighthouse system

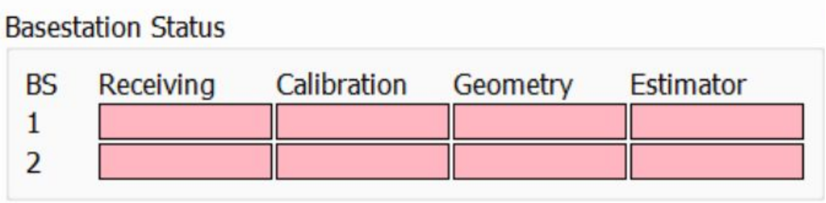
Introduction

The Lighthouse positioning system uses the SteamVR Base stat
this system, the Crazyflie can estimate its own X, Y and Z in a gl

This tutorial is also available as a video.

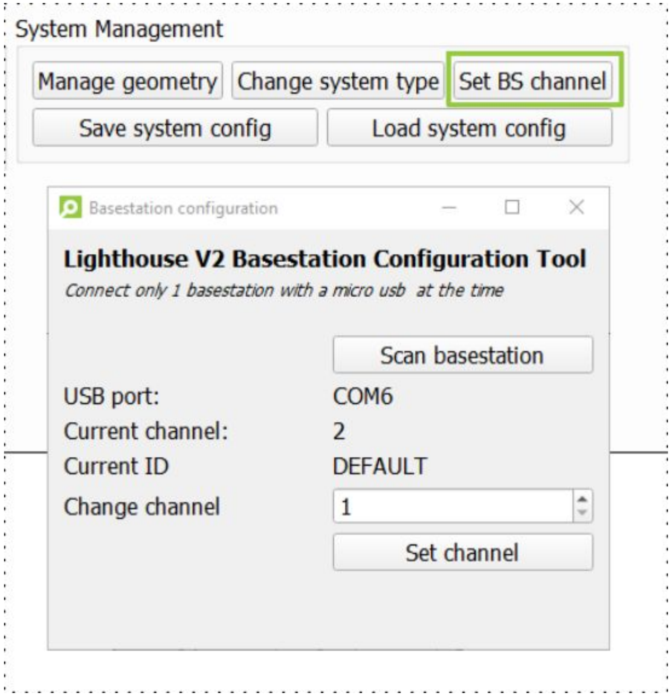
Lighthouse Base Station

- Steam VR base stations 2.0 by Valve
- Different channels



Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟥	🟥	🟥	🟥
2	🟥	🟥	🟥	🟥



Lighthouse V2 Basestation Configuration Tool

Connect only 1 basestation with a micro usb at the time

USB port: COM6

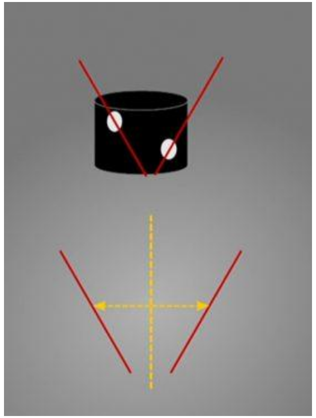
Current channel: 2

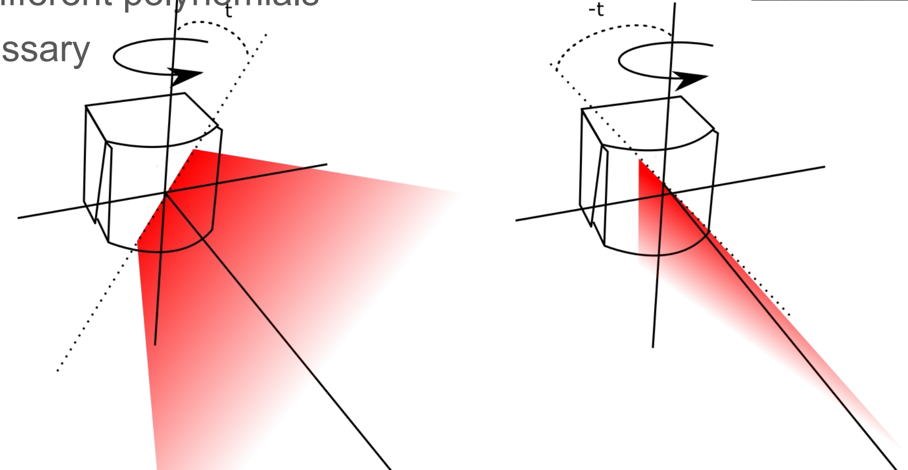
Current ID DEFAULT

Change channel 1

Mechanics Base station

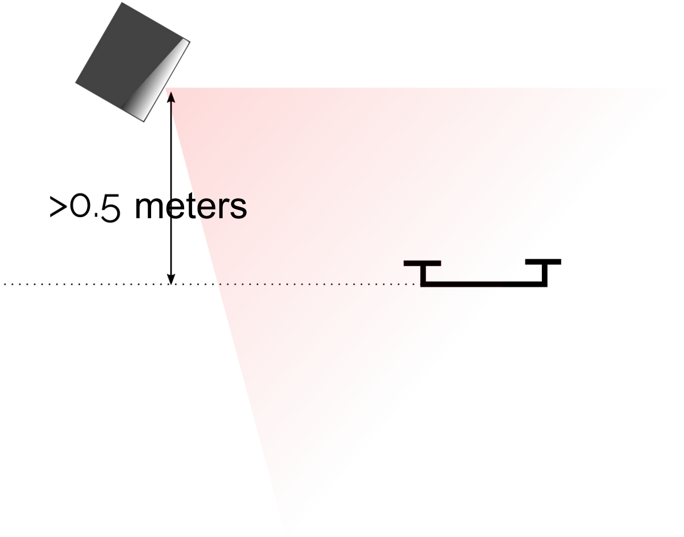
- Infrared light sweep
- FOV
  - H: 150 deg / V 110 deg
- Single drum with 2 tilted beam \ /
  - 30 degrees from the center
- Two pulse / turn @ ~50Hz
- Different channels = Different polynomials
  - No syncing necessary
  - >2 basestations
  - Interference



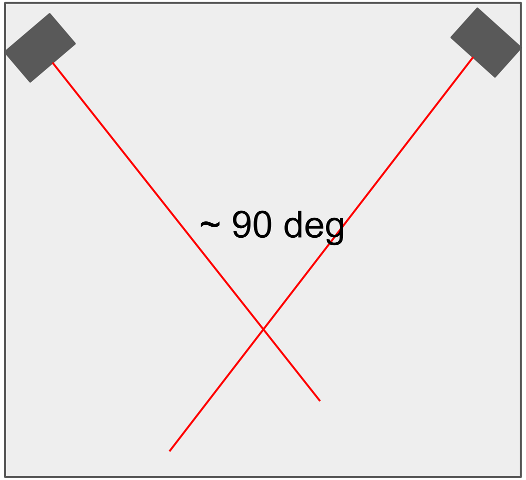


Basestation placement

- Flight area
  - 5 x 5 x 3 meters
  - Very little sunlight
  - No or very little reflections

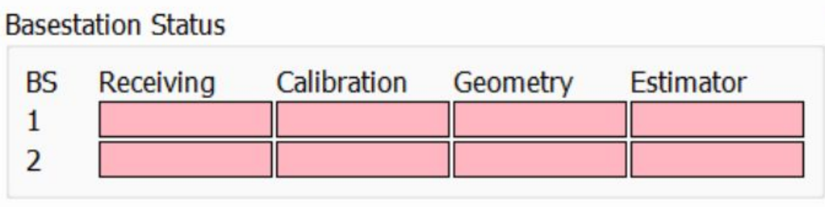


Top view



Receiving status

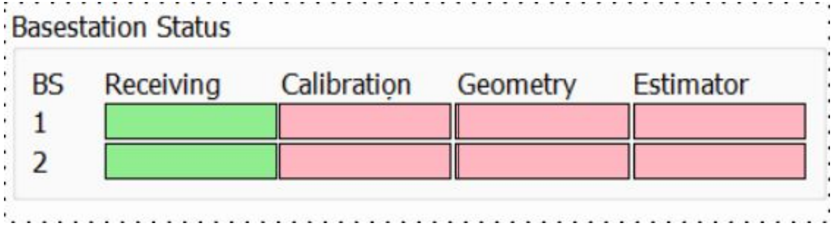
- Not receiving sweeps



Basestation Status

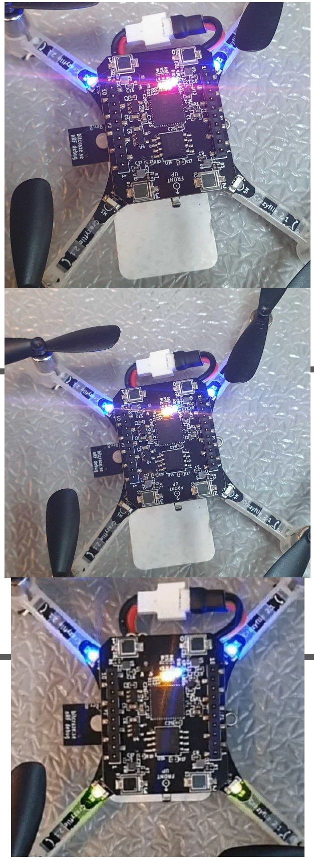
BS	Receiving	Calibration	Geometry	Estimator
1	🟥	🟥	🟥	🟥
2	🟥	🟥	🟥	🟥

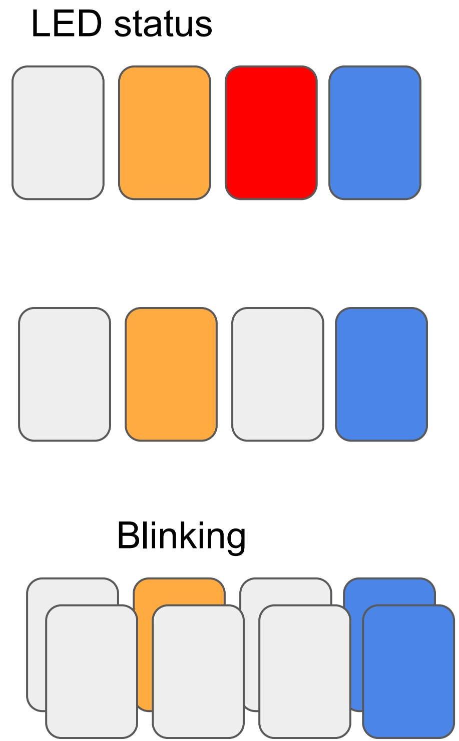
- Receiving sweeps, no geo data



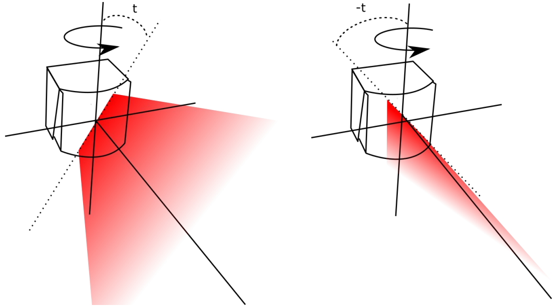
Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟥	🟥	🟥
2	🟩	🟥	🟥	🟥





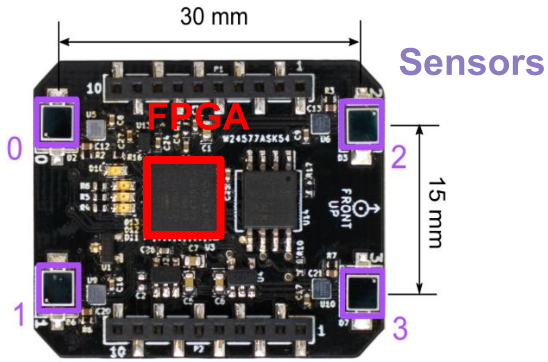
Decoding sweeps



- Sweeps hit the light sensor
  - Time stamp
- Data encoded in Sweep
  - Start of revolution (offset)
  - Basestation Identification
  - Calibration data
- Need 2 sensors for decoding offset, identity and channel

```
Sensor:1  TS:16159895  offset:151736  Chan: 2(0)
Sensor:0  TS:16160332  offset:151736  Chan: 2(0)
Sensor:3  TS:16161232  offset:151736  Chan: 2(0)
Sensor:2  TS:16161686  offset:151736  Chan: 2(0)
Sensor:0  TS:16308402  offset:335460  Chan: 3(0)
Sensor:1  TS:16308740  offset:335460  Chan: 3(0)
Sensor:2  TS:16309669  offset:335460  Chan: 3(0)
Sensor:3  TS:16309995  offset:335460  Chan: 3(0)
```

- Timestamp and offset-> Sweep angle.



Lighthouse positioning deck.

Interference

- Sweeps of 2 basestations hit the sensors at the same time
- Timestamps are then disregarded

```
Sensor:1  TS:16059976  offset:     -  Chan: 3(0)
Sensor:0  TS:16060303  offset:168352  Chan: 3(0)
Sensor:1  TS:16060606  offset:     -  Chan: 2(0)
Sensor:0  TS:16060987  offset:153272  Chan: 2(0)
Sensor:3  TS:16061124  offset:     -  Chan: -(-)
Sensor:2  TS:16061464  offset:     -  Chan: -(-)
Sensor:3  TS:16062268  offset:     -  Chan: 2(0)
Sensor:2  TS:16062653  offset:154936  Chan: 2(0)
```

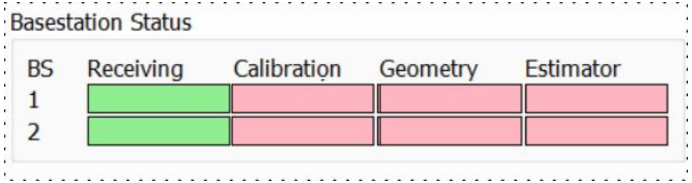
Calibration values

- Slow bits from decoding

```
Sensor:0  TS:16060303  offset:168352  Chan: 3(0)
```

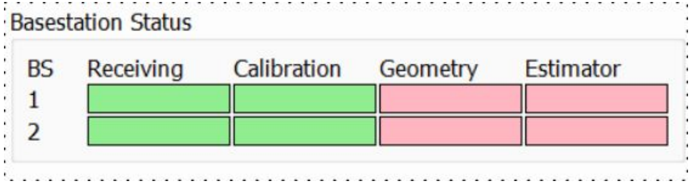
- Takes about 20 seconds
- Console output:

```
LH: Got calibration from F4A1A908 on channel 2
LH: Got calibration from 202A870D on channel 1
```



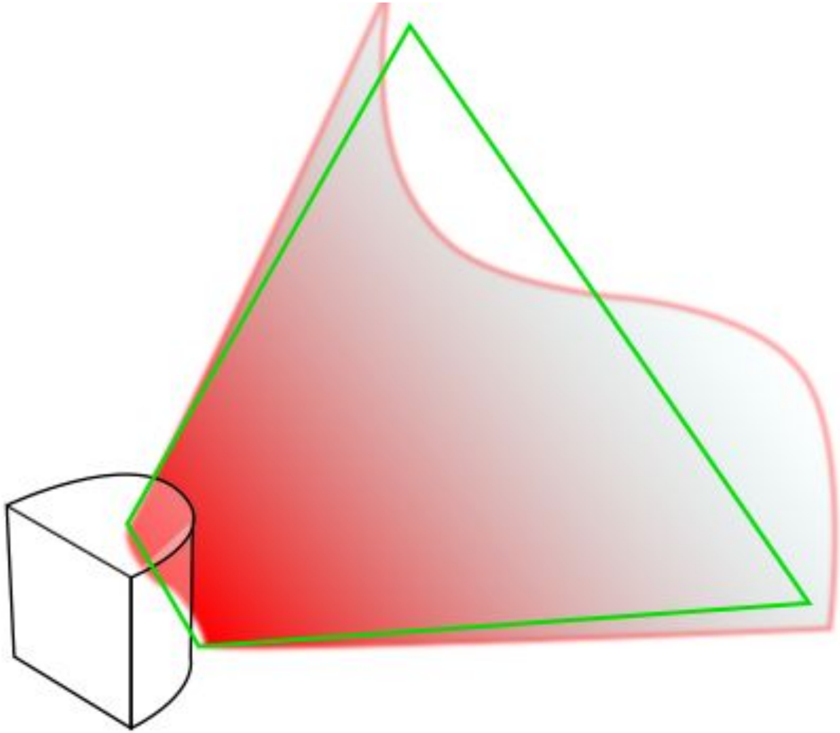
Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟥	🟥	🟥
2	🟩	🟥	🟥	🟥



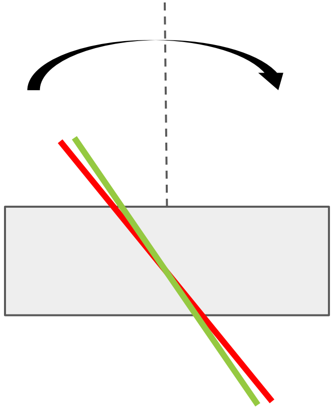
Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟥	🟥
2	🟩	🟩	🟥	🟥

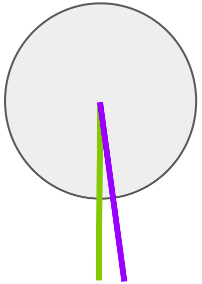


Calibration values meaning

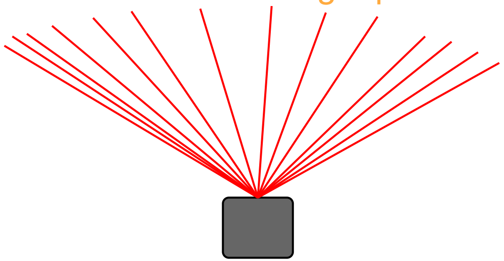
- Reversed engineering
- For each lightplane
  - Motor calibration
    - Tilt (error of lightplane tilt)
    - Phase (error rotating drum)
  - Distortion
    - Gibphase / -mag (Compression along the sweep?)
    - Curve (Curvature of lightplane?)
    - Ogeephase/ -mag (Shape of lightplane?)

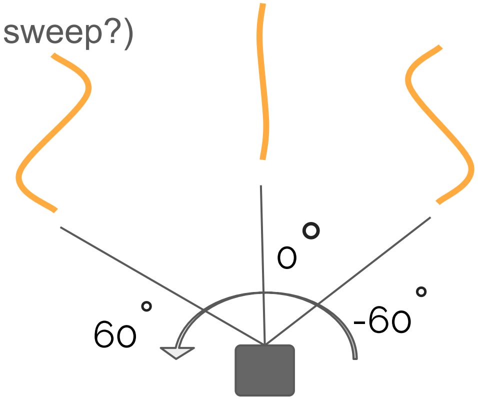


Front view



Top view





Calibration model we implemented

- Input:
  - x, y, z (location of sensor)
  - Calibration data
    - Tilt error: δt_{calib}
    - Phase error: δp_{calib}
    - Gibbous magnitude : $g_{mag,calib}$
    - Gibbous Phase: $g_{phase, calib}$
- Output: distorted angle
- Unused data: Curve and Ogeemag/phase
- Crazyflie-firmware
  - utils/lighthouse_calibration.c

$$a_x = \arctan 2\left(\frac{x}{y}\right)$$

$$b = a_x + \arcsin\left(\frac{z - \tan(t - \delta t_{calib})}{r}\right)$$

$$g_{comp} = -g_{mag,calib} \cdot \cos(a_x + g_{phase,calib})$$

$$\alpha_{dist} = b - \delta p_{calib} + g_{comp}$$

Calibration model we implemented

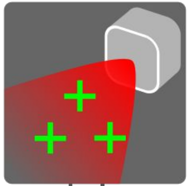
- Input:
  - x, y, z (location of sensor)
  - Calibration data
    - Tilt error: δt_{calib}
    - Phase error: δp_{calib}
    - Gibbous magnitude : $g_{mag,calib}$
    - Gibbous Phase: $g_{phase, calib}$
- Output: distorted angle
- Unused data: Curve and Ogeemag/phase
- Crazyflie-firmware
  - utils/lighthouse_calibration.c

$$a_x = \arctan 2\left(\frac{x}{y}\right)$$

$$b = a_x + \arcsin\left(\frac{z - \tan(t - \delta t_{calib})}{r}\right)$$

$$g_{comp} = -g_{mag,calib} \cdot \cos(a_x + g_{phase,calib})$$

$$\alpha_{dist} = b - \delta p_{calib} + g_{comp}$$

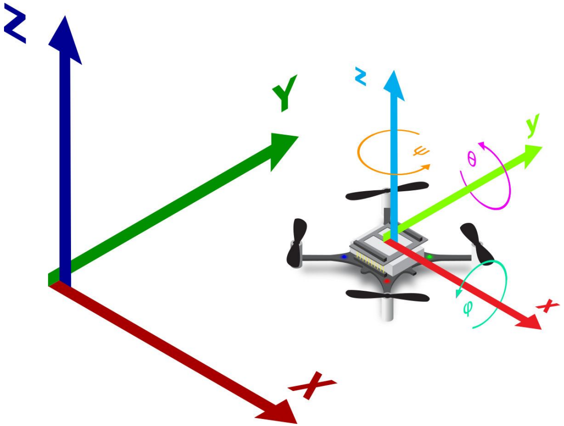


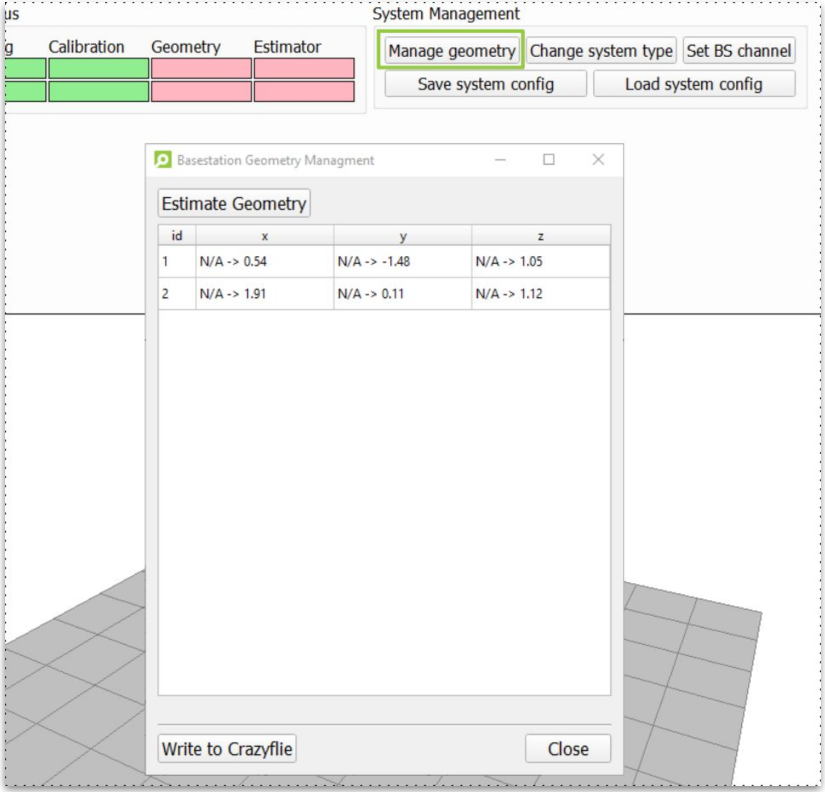
- Libsurvive uses a different calibration model
  - survive_reproject_gen2.c

Limitations Calibration

- Takes long time to receive calibration values
- Calibration model
  - Still up to 10 cm inaccuracy at the edges FOV Base stations
  - We don’t use curve or ogee values…

Geometry Estimation





g	Calibration	Geometry	Estimator
🟩	🟩	🟥	🟥
🟩	🟩	🟥	🟥

id	x	y	z
1	N/A -> 0.54	N/A -> -1.48	N/A -> 1.05
2	N/A -> 1.91	N/A -> 0.11	N/A -> 1.12

Geometry Estimation

- Translation and orientation
- Finds an object pose from 3D-2D point correspondences
  - Basestation is a 'camera'
- Crazyflie-lib-python/cflib/localization
  - **Documentation?**
- OpenCV
  - solvePnP
  - Give an initial estimate

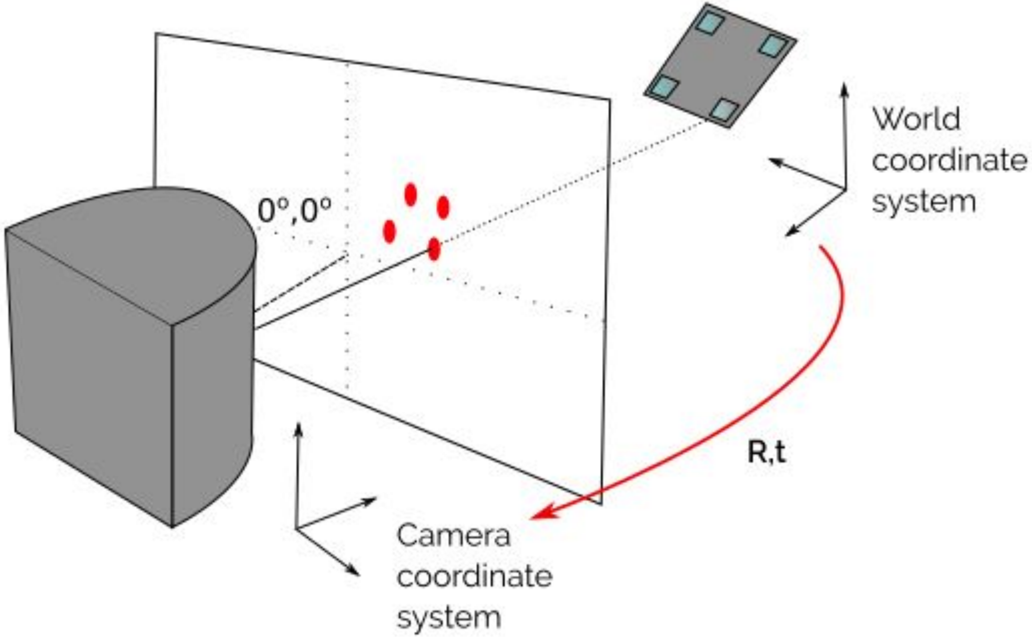
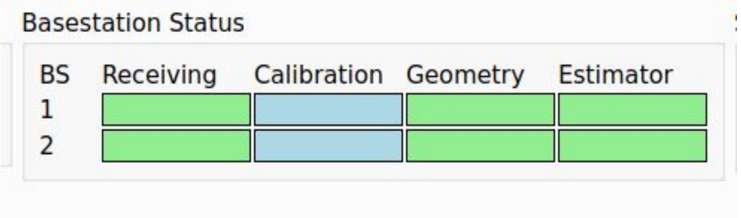


Image inspired by solvepnp description (https://docs.opencv.org/)

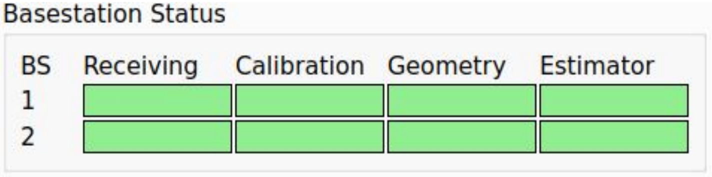
Persistent Memory

- Saving data between reboots
- Geometry and Calibration data
- Double checking Calibration data
  - Using calibration data directly from persistent memory
  - Calibration matches current system
  - Calibration of persistent memory does not match the current system
- Switching between different Lighthouse systems -> Orange warning.
  - Press estimate geometry again to resolve this.



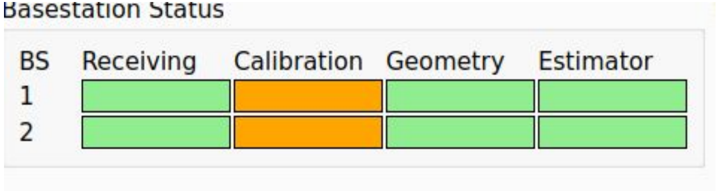
Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟦	🟩	🟩
2	🟩	🟦	🟩	🟩



Basestation Status

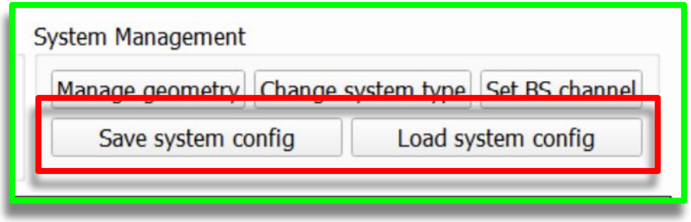
BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟩
2	🟩	🟩	🟩	🟩



Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟧	🟩	🟩
2	🟩	🟧	🟩	🟩

Saving config data in a File



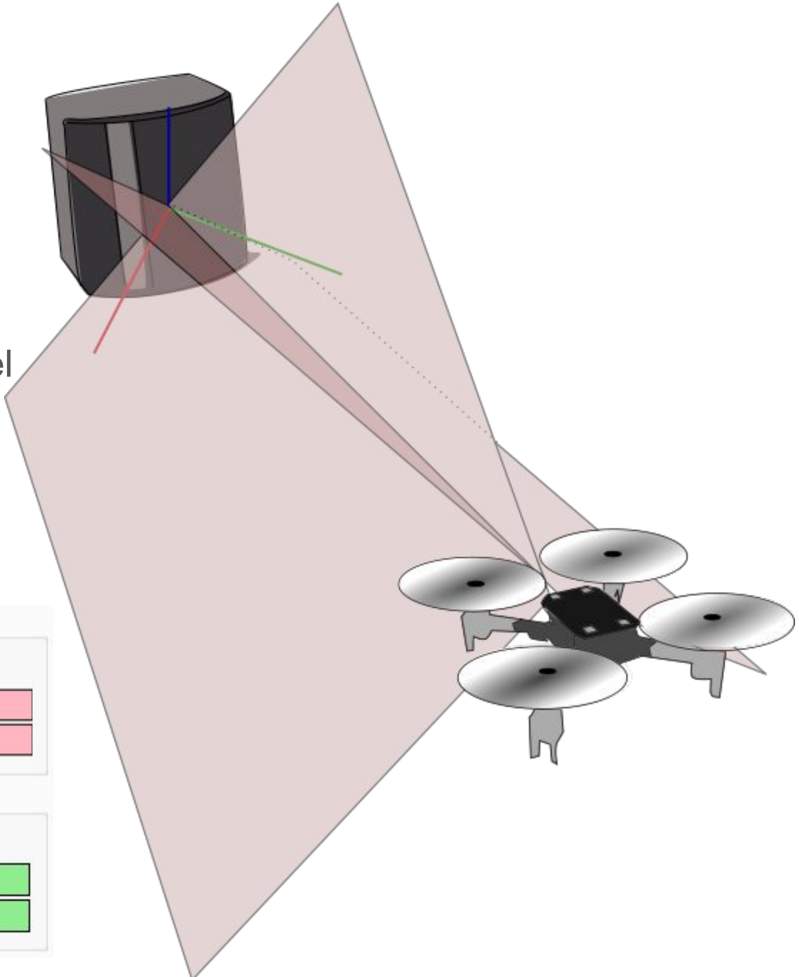
- Config saved for each basestation in .yaml
  - Geometry
  - Calibration
- Handy for:
  - Multiple lighthouse systems in different rooms
  - Unified coordinate system in a swarm
- Swarms
  - Use on one crazyflie for geometry estimation,
  - Save the *.yaml with ‘Save System Config’
  - Connect to the other crazyflies one by one
  - Upload data in yaml in persistent memory by ‘Load system config’

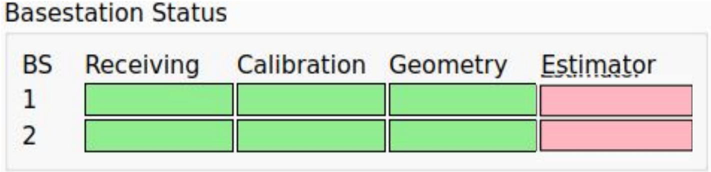
Limitations geometry estimation

- OpenCV python library (gives problems to cfclient)
- Initial position guess needed for a correct estimation
- Using only one position to estimate:
  - Origin coordinate system
  - Basestation pose
  - maybe better to use multiple different positions, like in MoCap ?

Position estimation

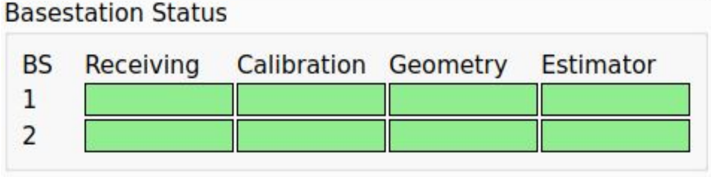
- State estimation
- Different methods
  - Crossing beam
  - EKF measurement model





Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟥
2	🟩	🟩	🟩	🟥

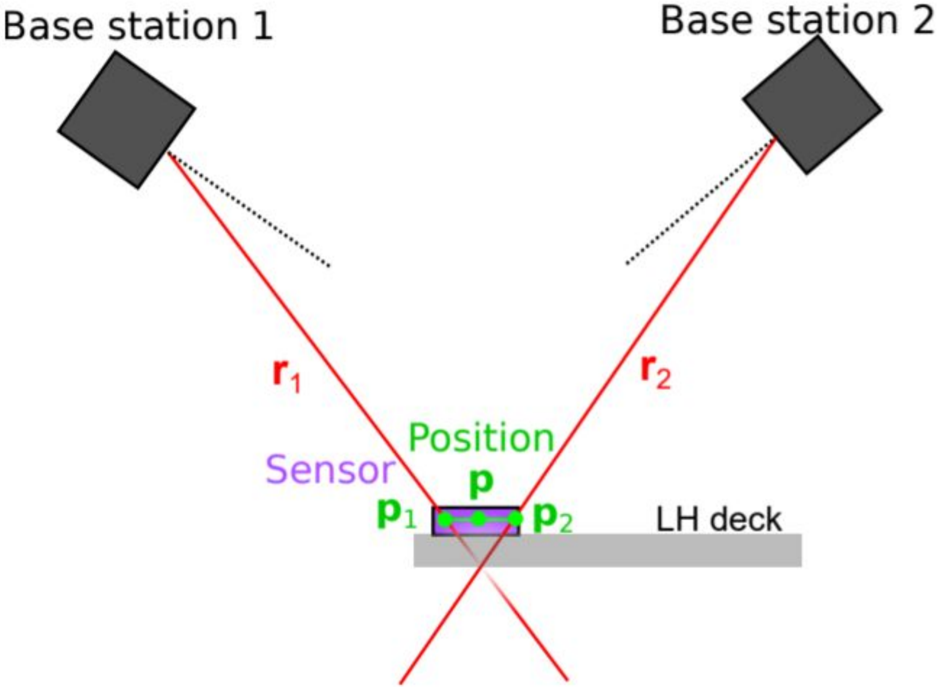


Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟩
2	🟩	🟩	🟩	🟩

Crossing Beam

- Two LH Base Stations
- Intersection of both sweep planes (rays)
- Closest distance between lines
- Lighthouse_position_est.c



$$\boldsymbol{p}_{1,s}, \boldsymbol{p}_{2,s} = \underset{\boldsymbol{p}_1 \in \boldsymbol{r}_{1,s},\ \boldsymbol{p}_2 \in \boldsymbol{r}_{1,s}}{\arg\min} \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|_2$$

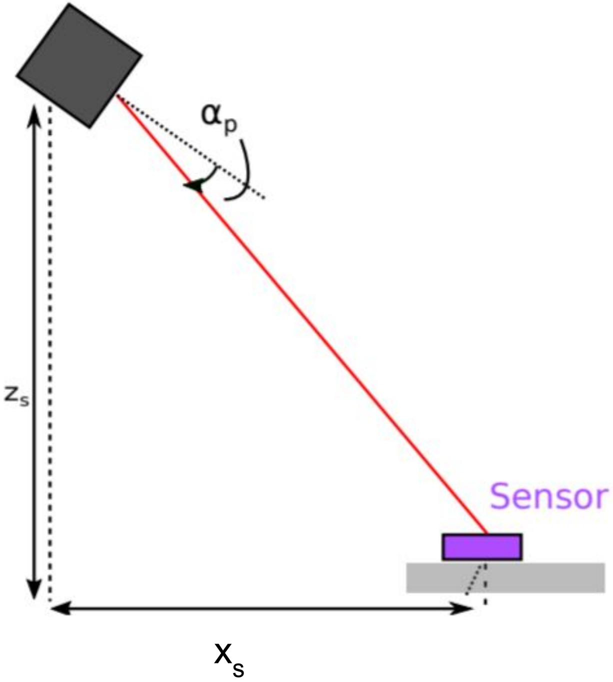
http://geomalgorithms.com/a07-_distance.html#Distance-between-Lines

Measurement model Light plans

- Extended Kalman Filter (EKF)
- Raw IR light sweep angles
- Measurement model
    - Position estimate
    - To wanted sweep angles

$$\boxed{\begin{aligned}\alpha_p &= \arctan\frac{y_s}{x_s} + \arcsin\frac{z_s \tan t_p}{r_s},\\ \text{where } r_s &= \sqrt{x_s^2 + y_s^2}.\end{aligned}}$$

- Calibration model included in implementation
- kalman_core/mm_sweep_angles.c



Limitations

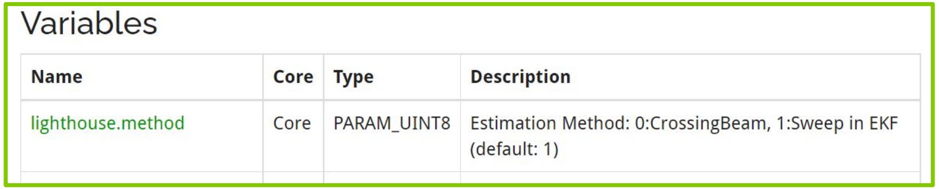
Crossing beam:

- - Only works with 2 base stations
- + Currently more stable
  - If both basestations are visible

EKF:

- + can work with any number of base stations
- - has some flight stability issues sometimes

- You can switch between crossing beam or EKF positioning method with the parameter lighthouse.method (0= crossing beam, 1= EKF)



Variables

Name	Core	Type	Description
lighthouse.method	Core	PARAM_UINT8	Estimation Method: 0:CrossingBeam, 1:Sweep in EKF (default: 1)

Some handy information about position estimation!

- Lighthouse does yaw estimation (not full pose)
    - Lighthouse_positioning_est.c / estimateyaw()

- You can use lighthouse in ground truth mode
    - No lighthouse sweeps will be sent to the estimator
    - Put *CFLAGS += -DLIGHTHOUSE_AS_GROUNDTRUTH* in config.mk, build and reflash the crazyflie-firmware and look at log lighthouse.x/y/z
    - This is only possible with crossing beam

More than 2 base stations

We don’t officially support this … but yes it is possible!

`PULSE_PROCESSOR_N_BASE_STATIONS` in pulse_processor.h

This should work to up to 16 basestations

BUT….

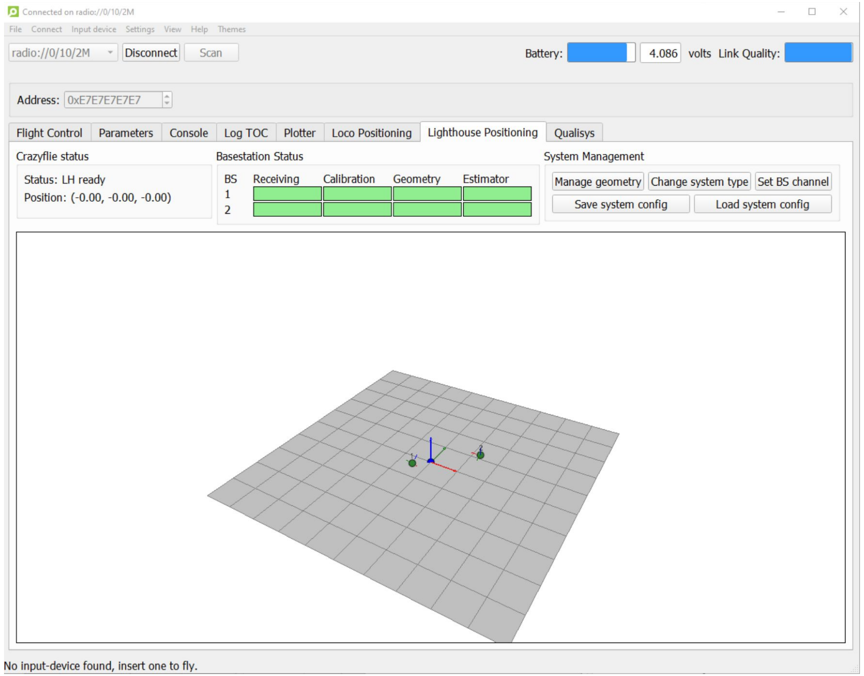
Geometry estimation tools only compatible with 2 basestations at the time

Interference: No more than 4 should be visible at the same time (ideally only 2)

Creativity necessary: App layer or Config file, and the forum :)

Also check out the hyper demo we did a year ago: https://www.bitcraze.io/2020/12/the-hyper-demo/

All is good!



Connected on radio://0/10/2M

radio://0/10/2M

Battery: 4.086 volts Link Quality:

Address: 0xE7E7E7E7E7

Crazyflie status

Status: LH ready

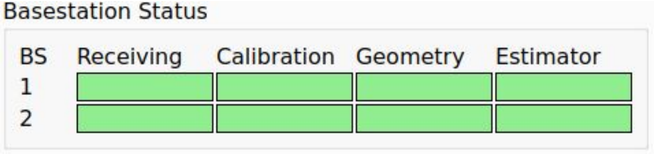
Position: (-0.00, -0.00, -0.00)

Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟩
2	🟩	🟩	🟩	🟩

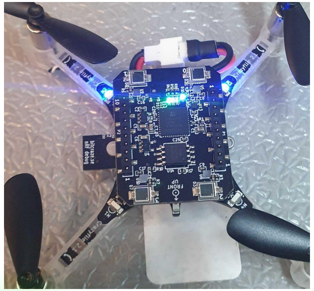
System Management

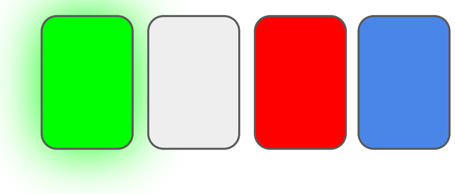
No input-device found, insert one to fly.



Basestation Status

BS	Receiving	Calibration	Geometry	Estimator
1	🟩	🟩	🟩	🟩
2	🟩	🟩	🟩	🟩





Finally

Possible improvements

- Out of early access…. But not perfect!
- Calibration model can be improved
- Better error distribution by the geometry estimation
- No easy setup for more than 2 base-stations

Recourses

www.bitcraze.io/lighthouse

Lighthouse Positioning System: Dataset, Accuracy, and Precision for UAV Research

ICRA 2021 Workshop

[2104.11523] Lighthouse Positioning System: Dataset, Accuracy, and Precision for UAV Research (arxiv.org)