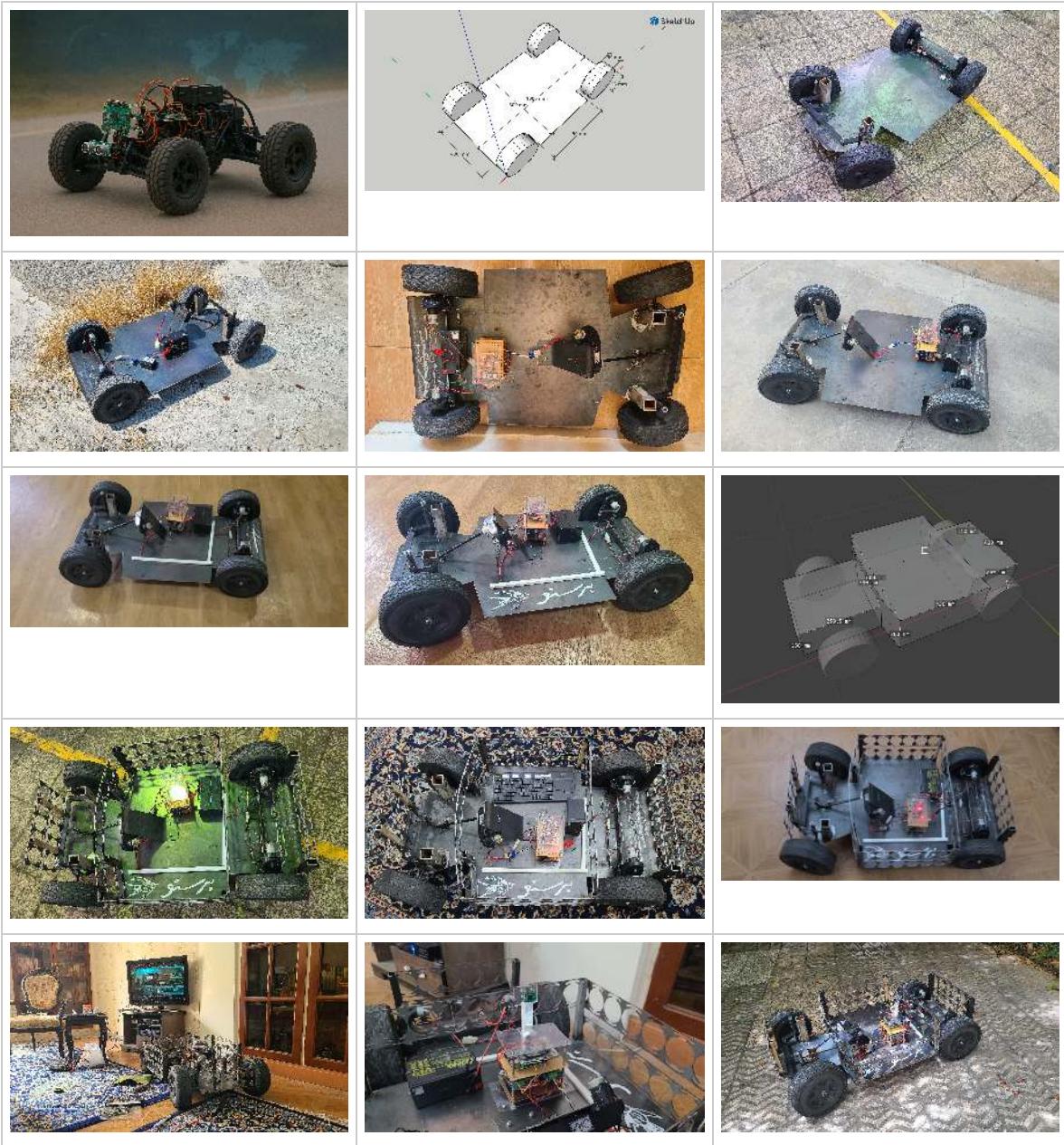


swallow

swallow is a family of UGVs named after the Iranian film “The Swallows Return to Their Nest” (پرستوها به لنه بر می‌گردند) from the 1940s ([wikipedia](#)), ([irmdb](#)), ([pictures](#)), ([pictures](#)). [arzhang](#) and [rangin](#) belong to this family.

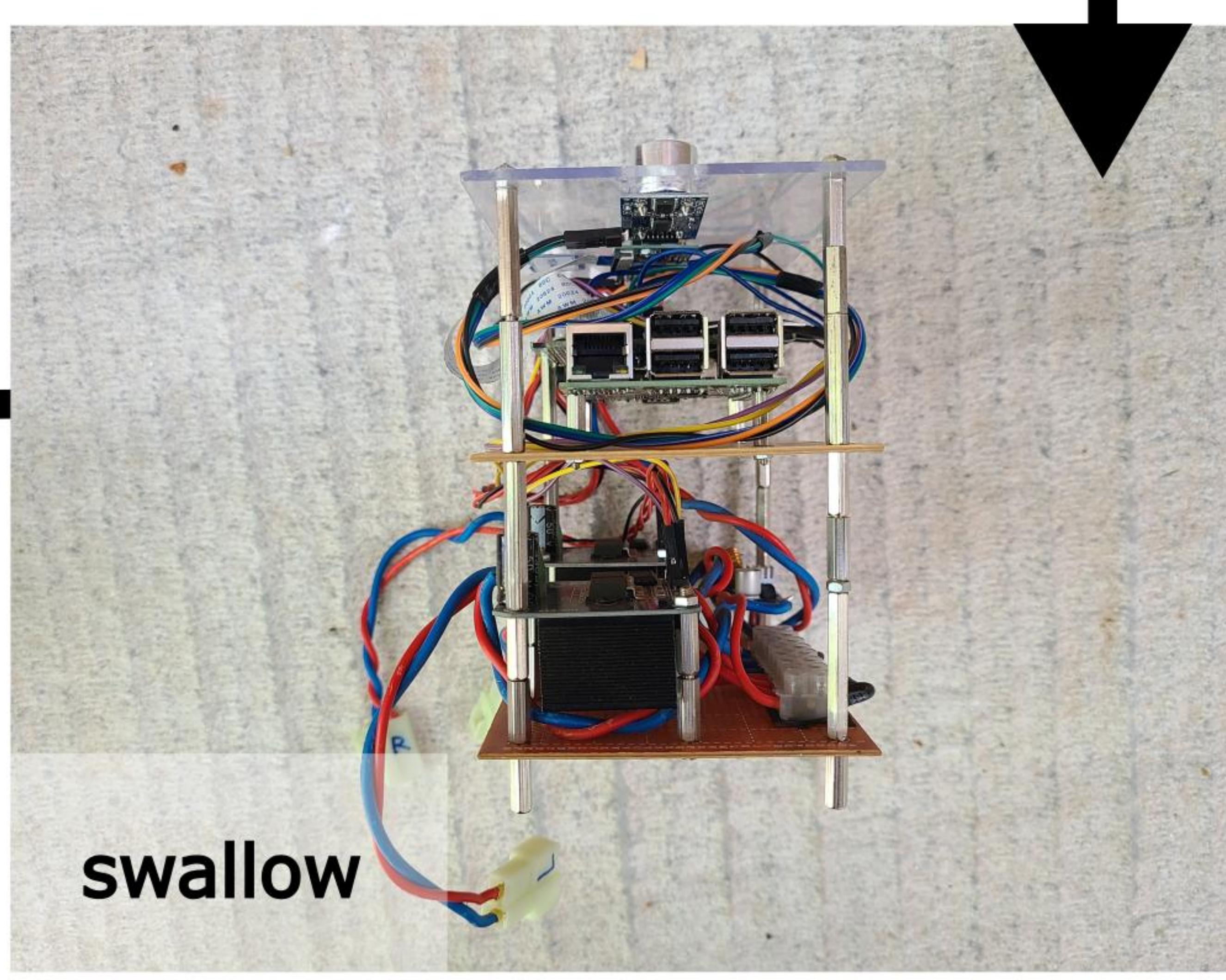
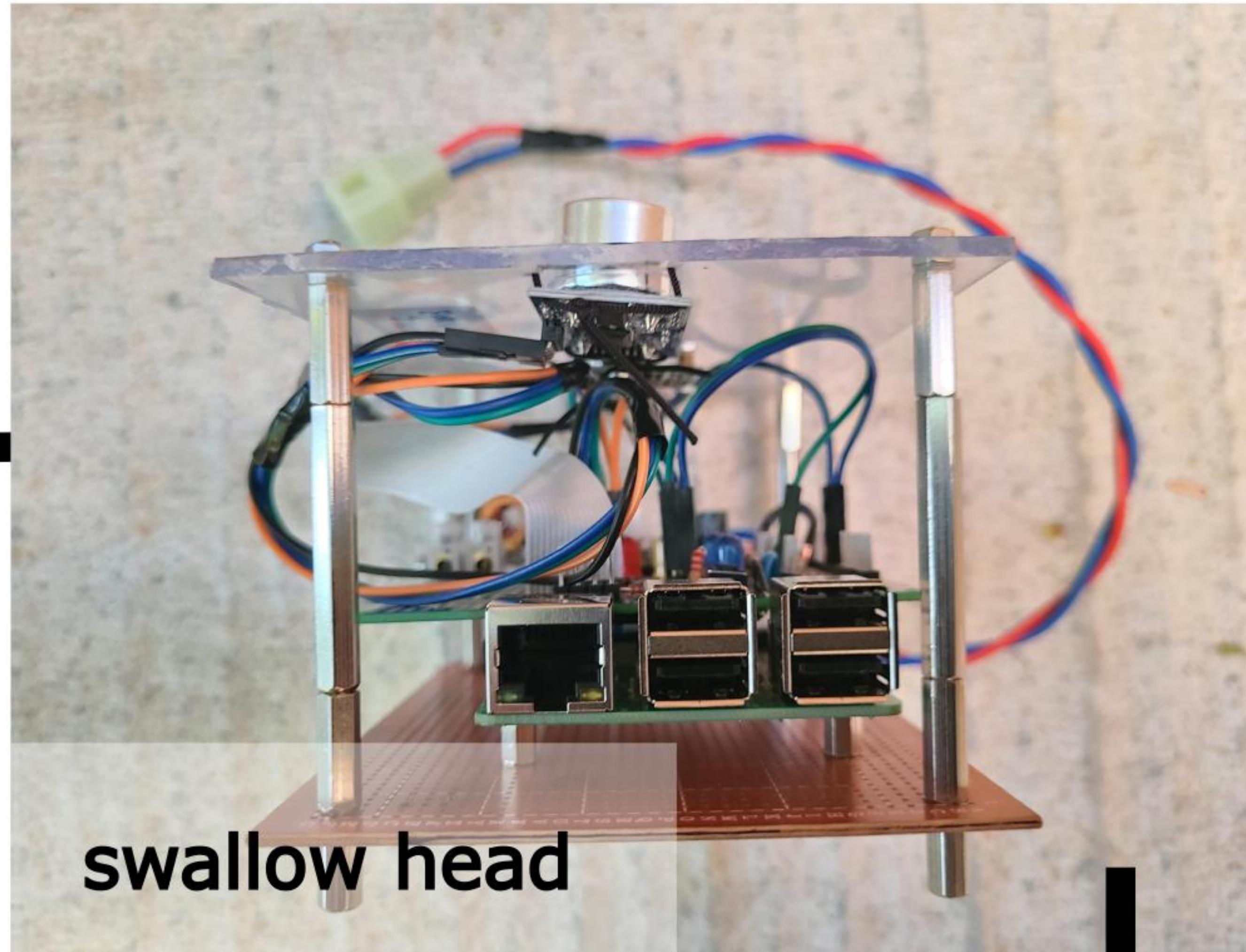
- [analog control](#)
- [digital control](#)



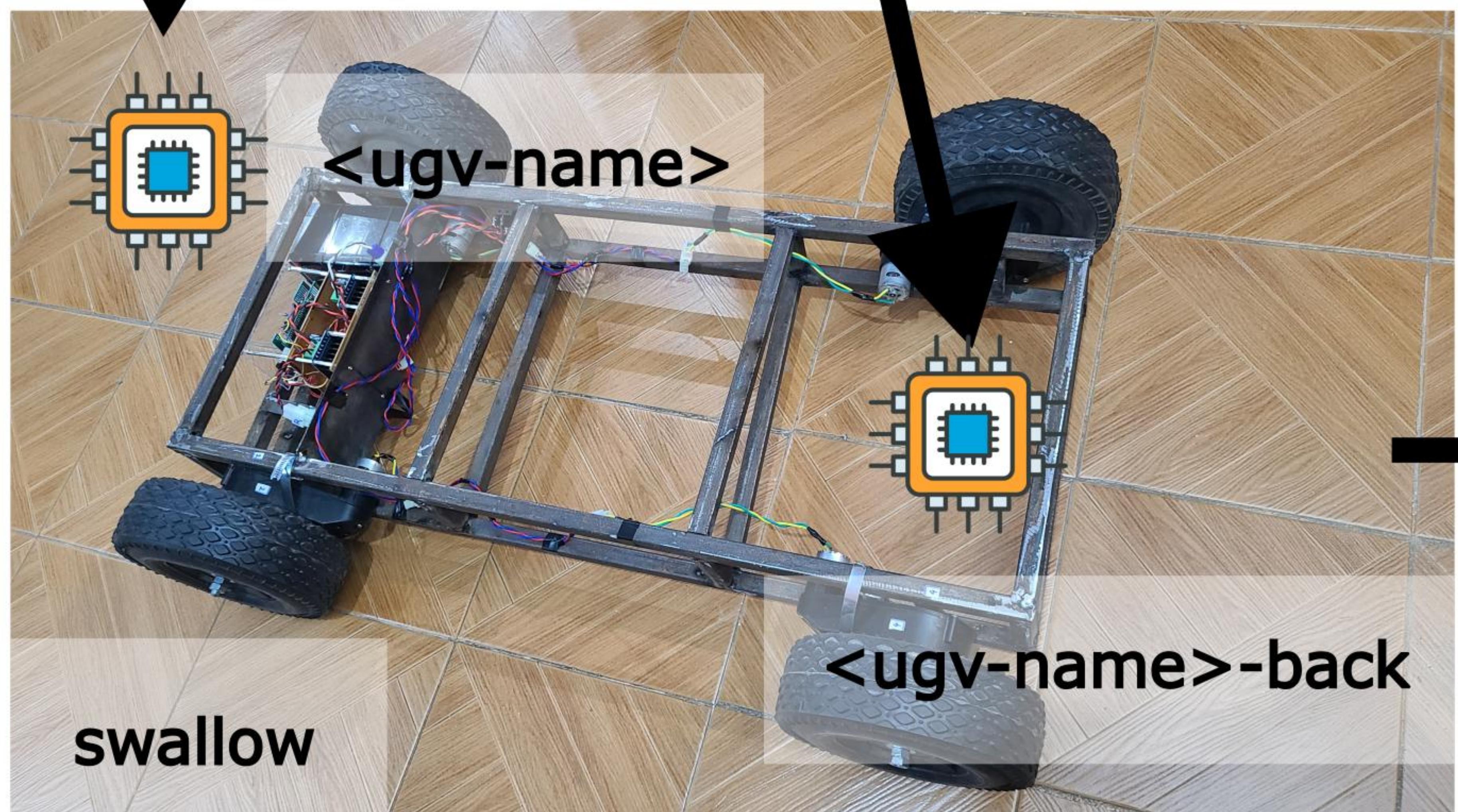


swallow class of UGVs

terminology:
computers,
designs, and
UGVs



designs



swallow

arzhang

arzhang2

arzhang3

rangin

swallow: digital

- [design](#)
- [@swallow](#)
- [algo](#)

swallow: digital: design

- [parts](#)
- [terraform](#)
- [shield](#)
- [rpi-pinout](#)
- [operation](#)
- [mechanical](#)
- [ultrasonic sensor](#)

obsolete

- [over-current detection](#)

swallow: digital: design: parts

a [swallow computer](#) + an optional [swallow-head computer](#) + 

1. [12 VDC motor, 20-45 W](#): type 2, 2 x right + 2 x left.
2. [Rechargeable sealed lead acid battery, 12 V, 7 Ah](#).
3. [auto power connectors](#): 4 pairs.
4. [charging port 5.5 mm](#).
5. [on/off switch, 12V DC 10 A, with indicator led](#).
6. [power wheel wheels](#): 4 x .

12 VDC motor, 20–45 W  type 2, 2 x right + 2 x left	Rechargeable sealed lead acid battery, 12 V, 7 Ah 	auto power connectors  4 pairs
charging_port 5.5 mm 	on/off switch, 12V DC 10 A, with indicator led 	power wheel wheels  4 x

aliases: swallow

dataset

```
@swallow \
    dataset \
    combine \
        [count=<count>,~download,~recent,sequence=<3>,~split,upload] \
        [-|<object-name>] \
        [--datasets <object-name-1>+<object-name-2>] \
        [--test_ratio 0.1] \
        [--train_ratio 0.8]
. combine swallow datasets.

@swallow \
    dataset \
    download \
        [~metadata,navigation|yolo]
. download the swallow dataset.

@swallow \
    dataset \
    edit \
        [~download,navigation|yolo]
. edit the swallow dataset.

@swallow \
    dataset \
    list \
        [~download,navigation|yolo]
. list the swallow dataset.

@swallow \
    dataset \
    upload \
        [~metadata,navigation|yolo]
. upload the swallow dataset.
```

env

```
@swallow \
    env \
    cp \
        [<env-name>]
. cp swallow swallow-raspbian-<env-name>.env.

@swallow \
    env \
    list
. list swallow envs.

@swallow \
    env \
    set \
        bps | full_keyboard | steering \
        0 | 1
. set env.
bps: BLUER_SBC_SWALLOW_HAS_BPS (currently: 0)
full_keyboard: BLUER_SBC_SWALLOW_HAS_FULL_KEYBOARD (currently: 0)
steering: BLUER_SBC_SWALLOW_HAS_STEERING (currently: 1)
```

keyboard

```
@swallow \
    keyboard \
    test \
    [dryrun] \
    [--keys 1234567890-+/.]
. test keyboard.
```

ultrasonic-sensor

```
@swallow \
    ultrasonic \
    review \
    [~download,upload] \
    [.|<object-name>] \
    [--frame_count <-1>] \
    [--gif 0] \
    [--rm_blank 0]
. review ultrasonic sensor data.
@swallow \
    ultrasonic \
    test \
    [~upload] \
    [-|<object-name>] \
    [--export 0] \
    [--frame_count <-1>] \
    [--gif 0] \
    [--log 0] \
    [--max_m 0.80] \
    [--rm_blank 0]
. test ultrasonic sensors.
```

swallow: digital: design: terraform

Raspbian 64-bit

⭐ preferred.

⚠️ on 32-bit, opencv, torch, and other modules install with challenges, and likely at lower versions.

1. follow [RPi](#) (use 64-bit + headless).
2. run in another terminal and paste the seed 🌱 into the ssh window.

```
@seed swallow_raspbian clipboard
```

3. run,

```
@bps install
```

4. run,

```
@swallow env cp navigation  
@swallow env set full_keyboard 1  
@init; @select; @session start
```

now press t, then w, and wait for ~20 seconds (or press a, d), then press i. an dataset should be uploaded that contains a few frames from the camera.

```
2025-08-23-22-04-20-qyq24g/grid.png | 23 August 2025 | 22:06:29 (+0330) |
100x100x3:uint8 | count: 3 | 3 subset(s): train: 3 [%100.0], test: 0 [%0.0],
eval: 0 [%0.0] | 3 class(es): no_action: 3 [%100.0], left: 0 [%0.0], right: 0
[%0.0]
```



```
bluer_algo-4.364.1 | bluer_ai-12.255.1-main | bluer_objects-6.252.1 |
bluer_options-5.164.1 | torch-2.8.0+cpu | Python 3.11.2 | Linux 6.12.25+rpt-
rpi-v8 | sparrow | 00000000953c665 | Sion
```

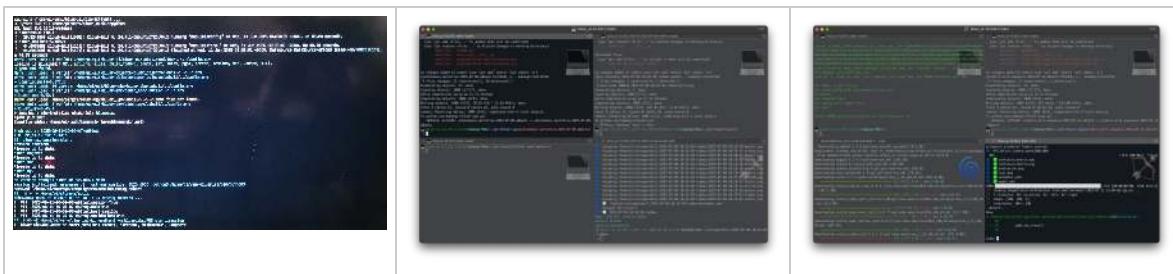
image

5. run,

```
@swallow env cp yolo
@swallow env set full_keyboard 1
@swallow env set bps 1
@init; @select; @session start
```

6. the terraform is complete, shut down the machine,

@host shutdown

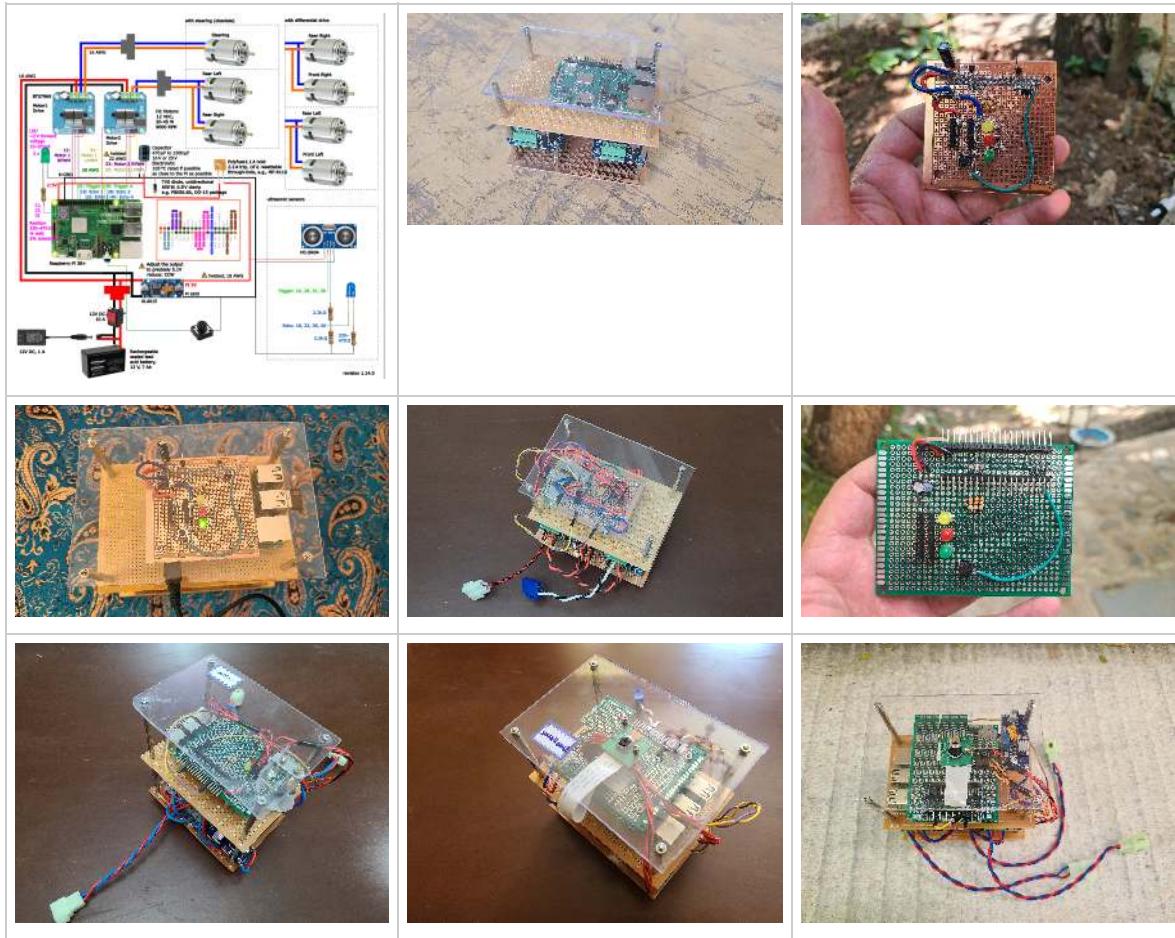


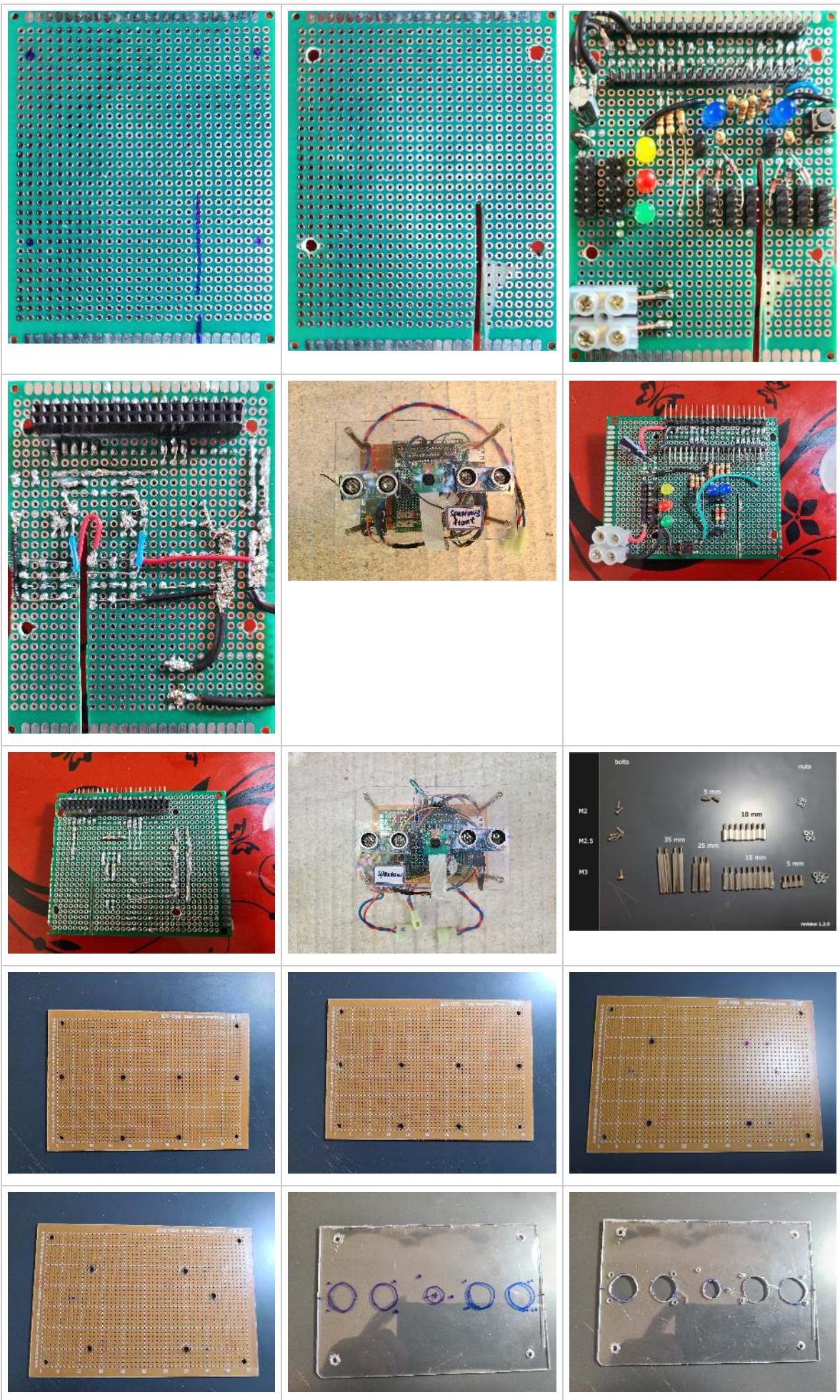
► Ubuntu 64-bit

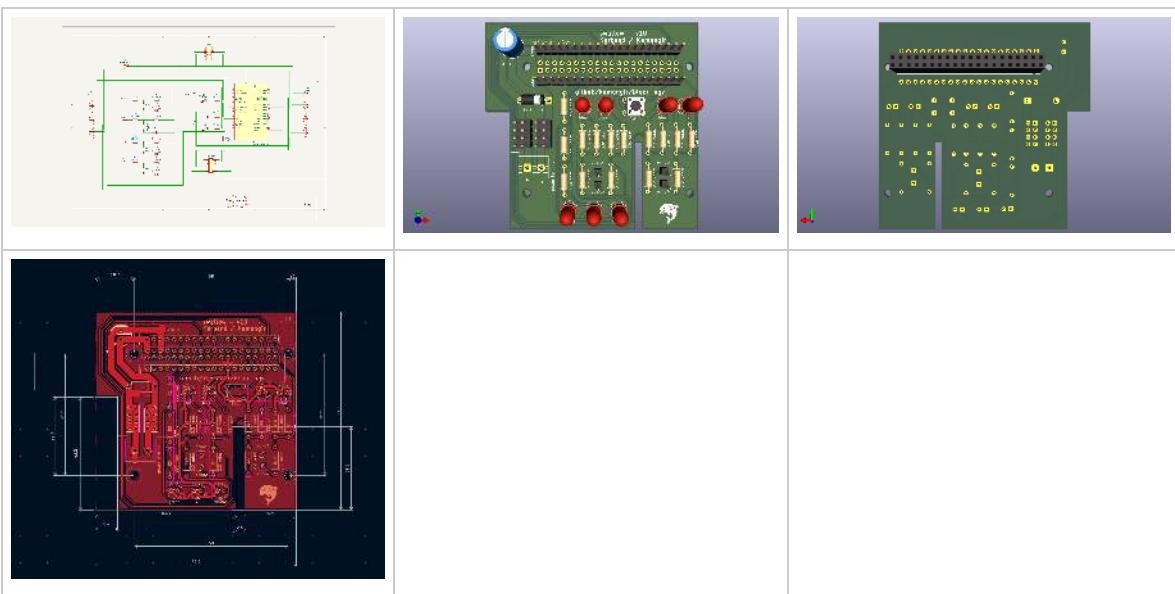
swallow: digital: design: shield


- [electrical diagram](#)
- [schematics and PCB](#)
- [parts and more views](#)
- [Importing a Design from KiCad - Altium Designer Technical Documentation](#)
- connectors: all socket.

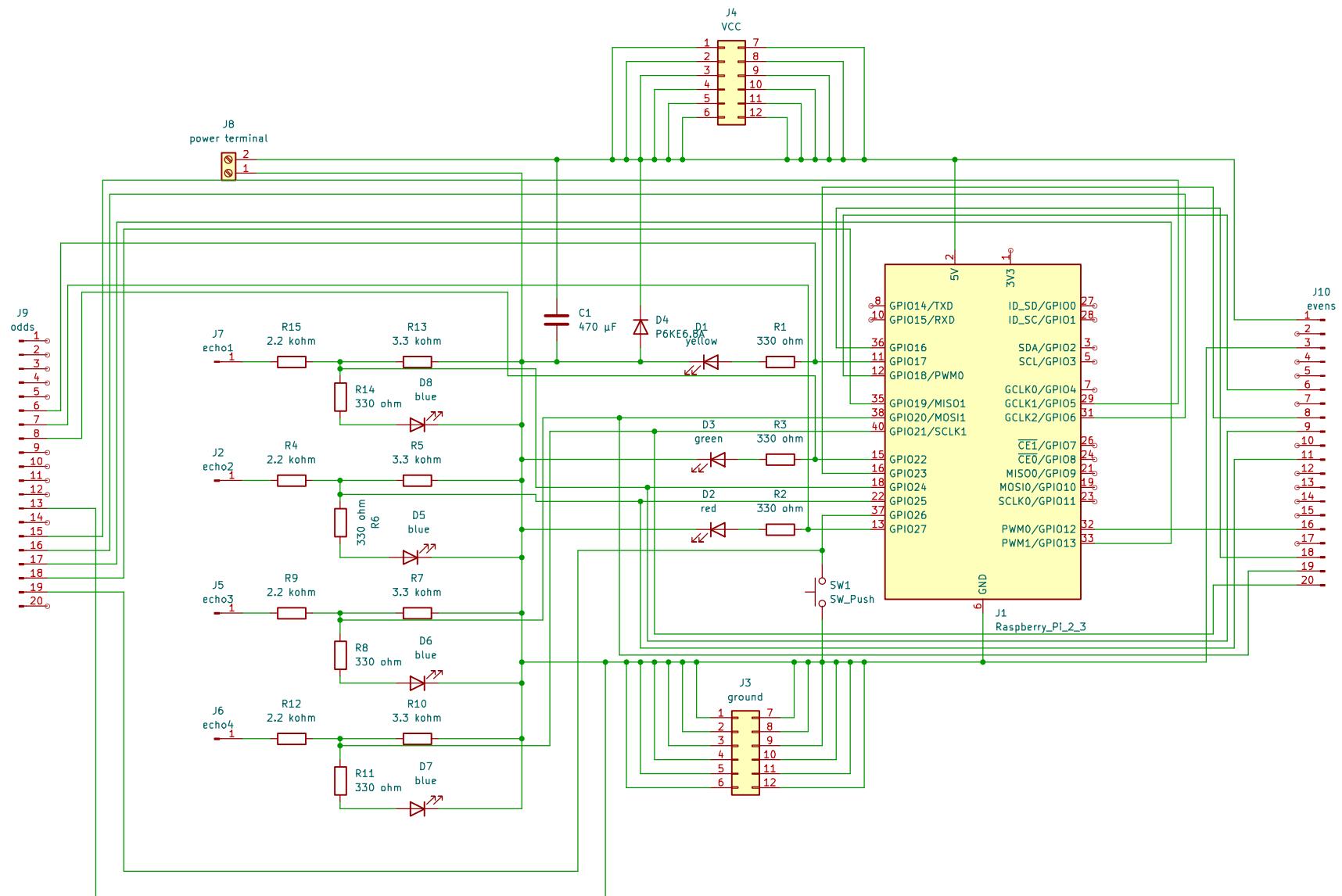
► v1





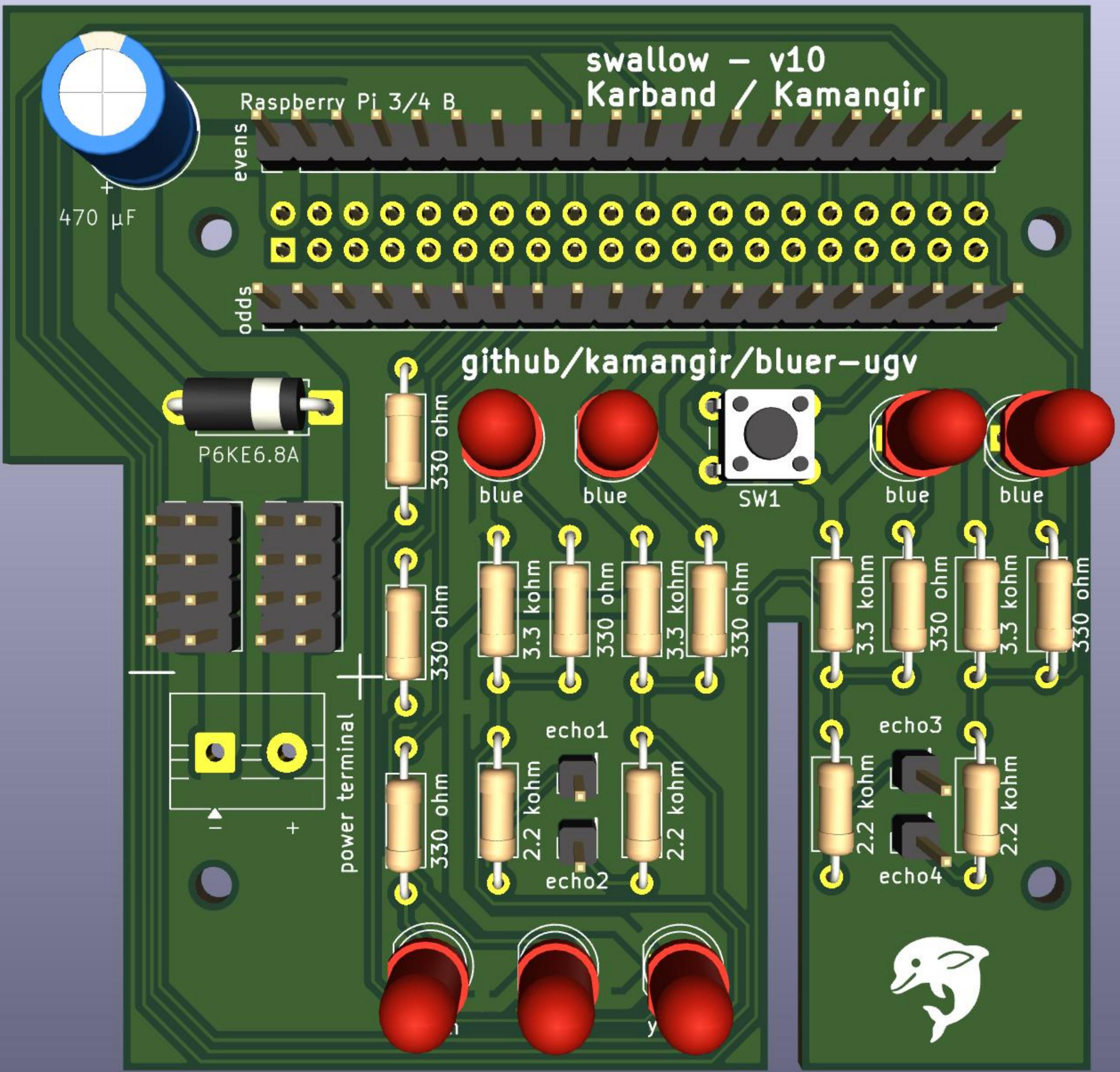


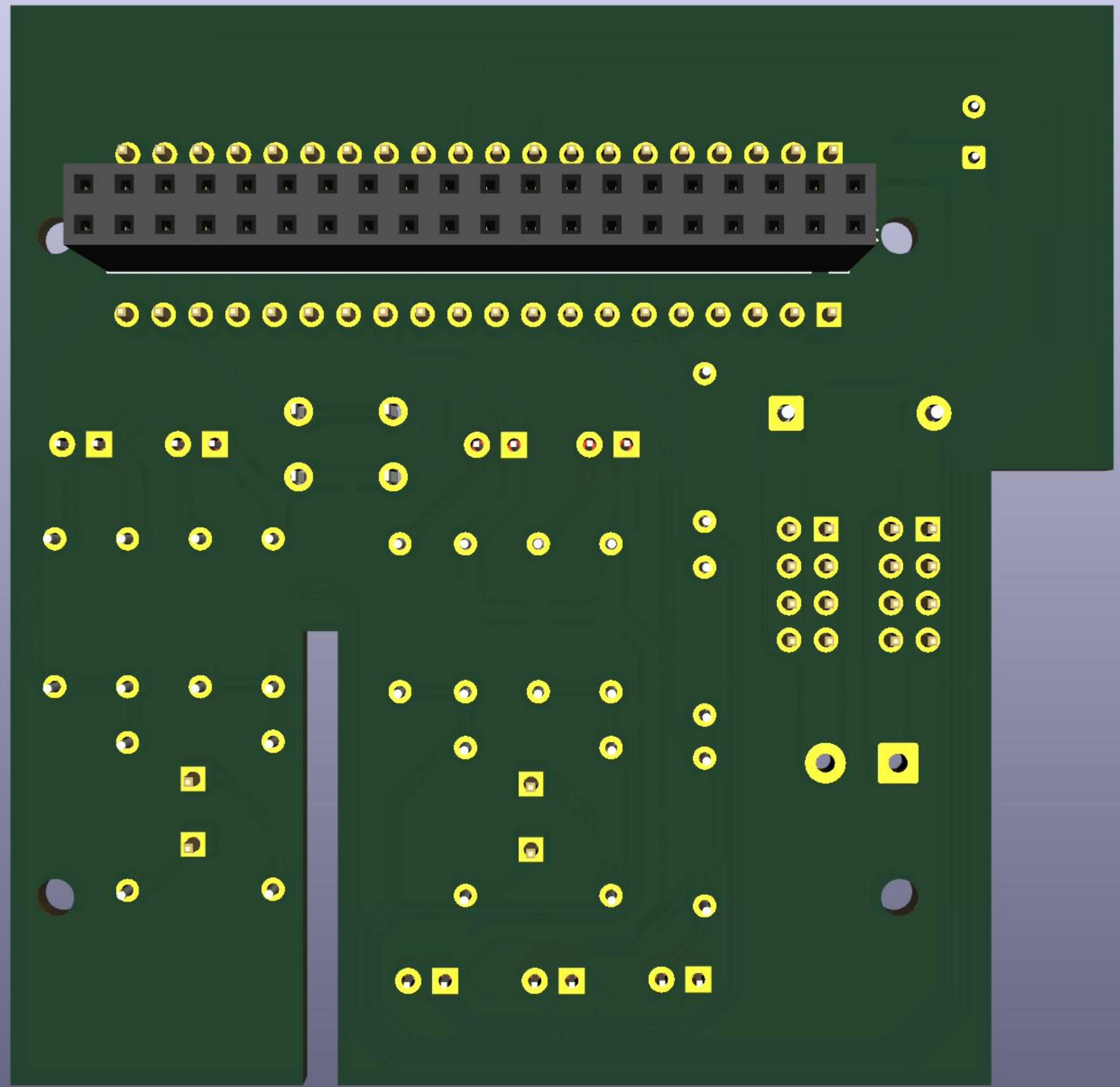


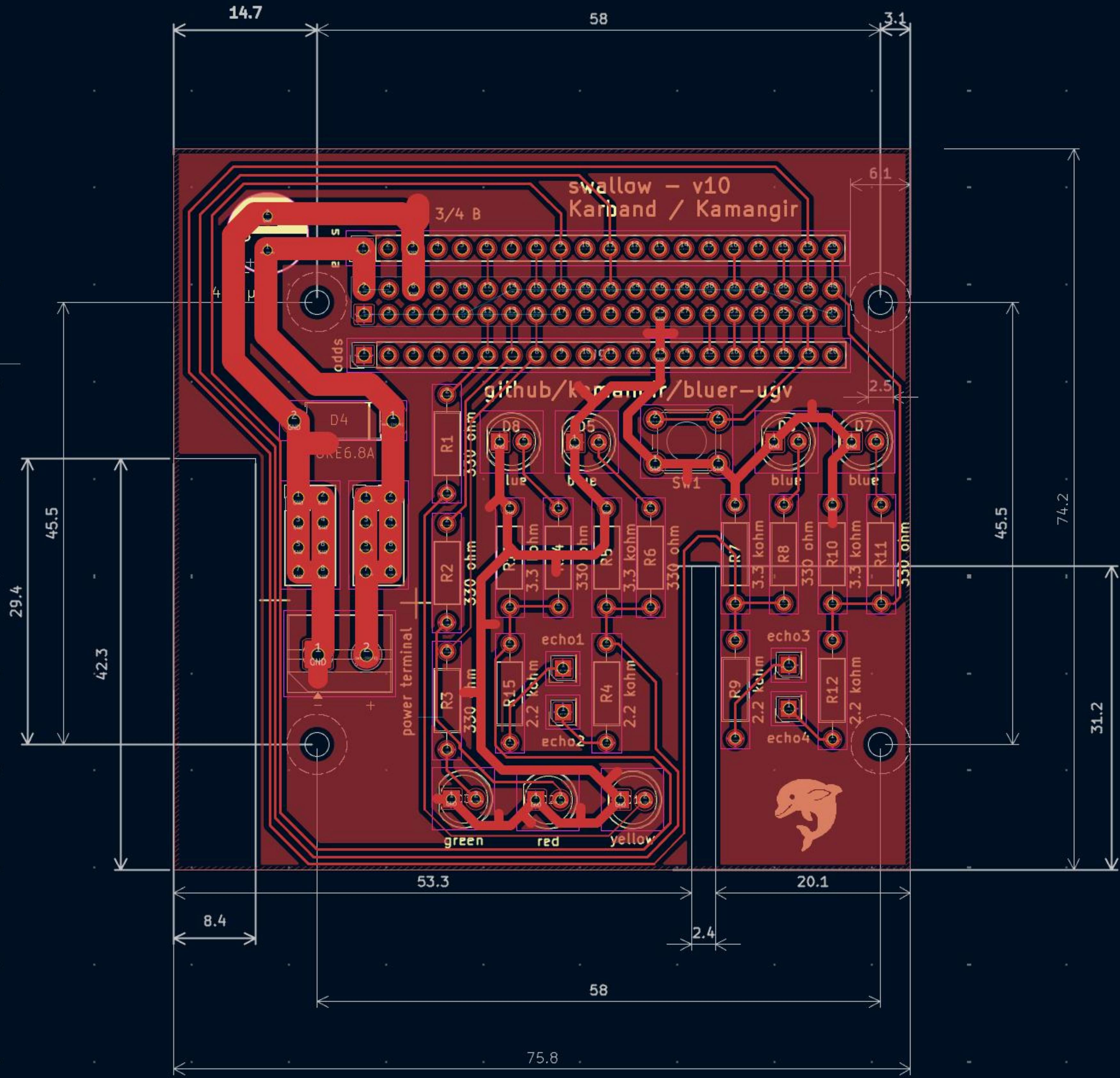


swallow shield

v4







swallow: digital: design: rpi-pinout

Responsibility	Function	Physical Pin	GPIO	Notes
Motor 1, FW	PWM	12	18	PWM0, Steering / Right
Motor 1, BW	PWM	32	12	PWM0 (alternate), Shares PWM0 with GPIO 18
Motor 2, FW	PWM	33	13	PWM1, Rear / Left
Motor 2, BW	PWM	35	19	PWM1 (alternate), Shares PWM1 with GPIO 13
Green LED	Digital Output	11	17	
RED LED	Digital Output	13	27	
Blue LED	Digital Output	15	22	
Push Button	Digital Input	37	26	
Sensor 1, Trigger	Digital Output	16	23	
Sensor 2, Trigger	Digital Output	29	5	
Sensor 3, Trigger	Digital Output	31	6	
Sensor 4, Trigger	Digital Output	36	16	
Sensor 1, Echo	Digital Input	18	24	
Sensor 2, Echo	Digital Input	22	25	
Sensor 3, Echo	Digital Input	38	20	
Sensor 4, Echo	Digital Input	40	21	

swallow: digital: design: operation

keyboard

event	full keyboard	numpad
debug off	v	9
debug on	b	7
exit	*i	*7
mode = action	g	1
mode = none	y	5
mode = training	t	3
reboot	*p	*5
shutdown	*o	*9
special key	z	.
speed backward	s	2
speed forward	w	8
steer left	a	4
steer right	d	6
stop		0
ultrasonic off	n	-
ultrasonic on	m	+
update	*u	*1

*: special key.

to enable full keyboard:

```
@swallow env set full_keyboard 1
```

the range of numpad is ~10-20 m range, noticeably lower than that of the full keyboard, which is ~50 m, see [village-6](#) for details.



leds

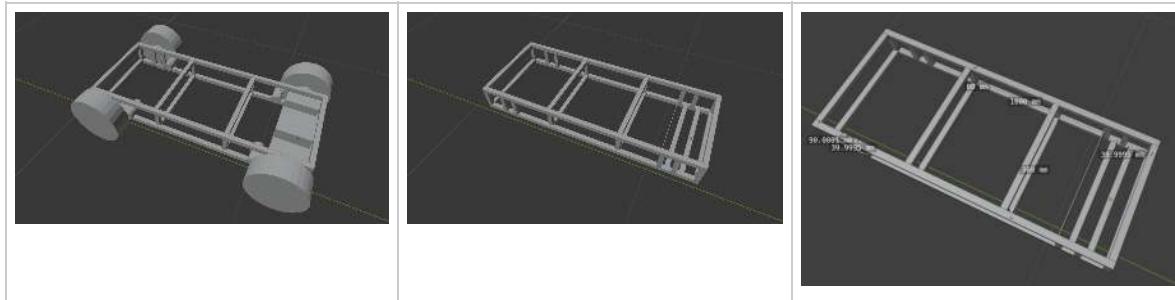
- green: control loop.
- red:
 - flashing:
 - motor update.
 - action / training.
 - release the push button to update.
 - setpoint update.
 - solid: release the push button to shutdown.
- yellow:
 - command received.
 - mousepad activity.
- blue: ultrasonic sensor echo.
- mouse pad (obsolete)

push button

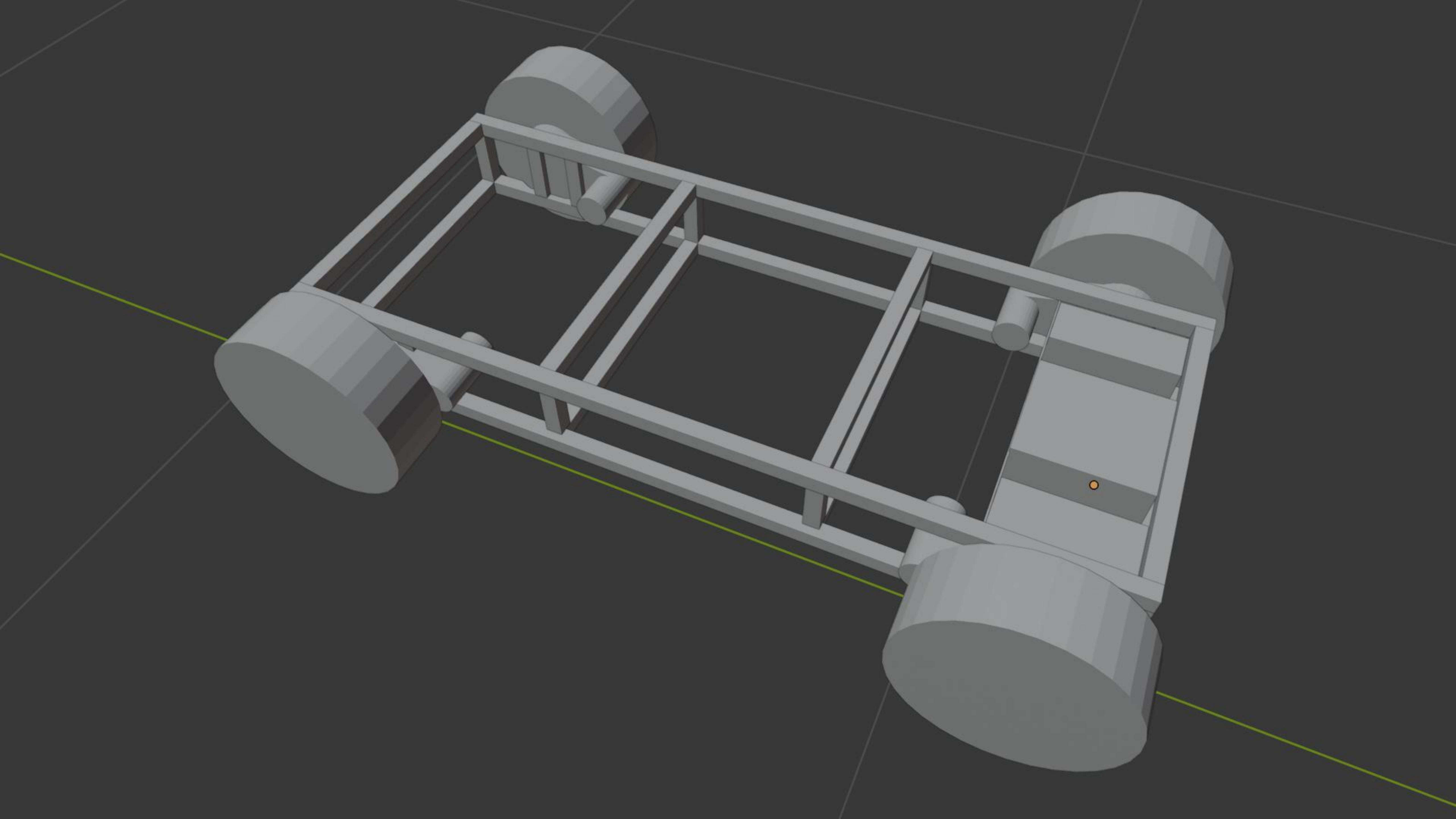
- hold for 5 seconds: update.
- hold for 10 seconds: shutdown.
- hold for > 15 seconds: skip.

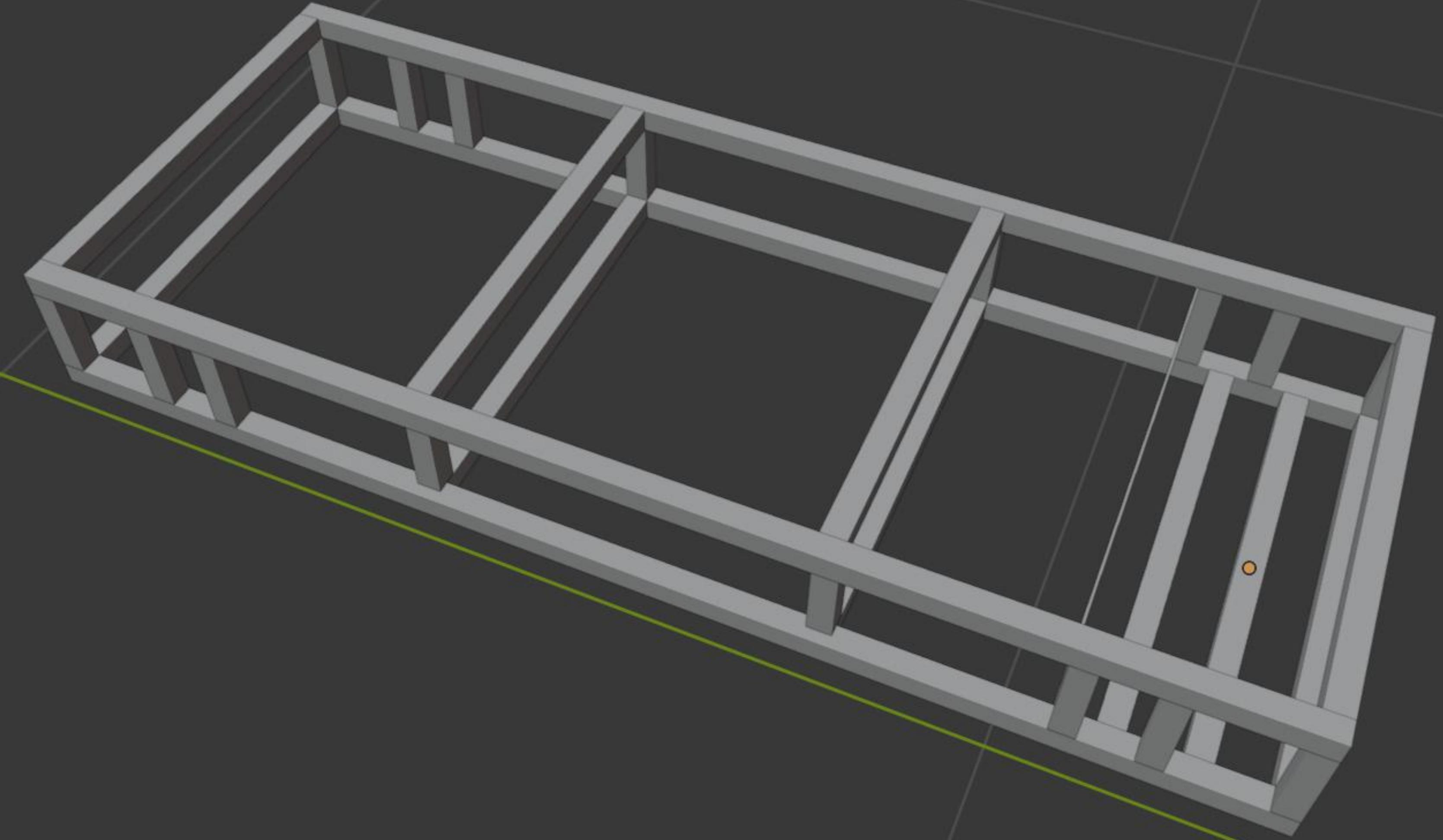
swallow: digital: design: mechanical

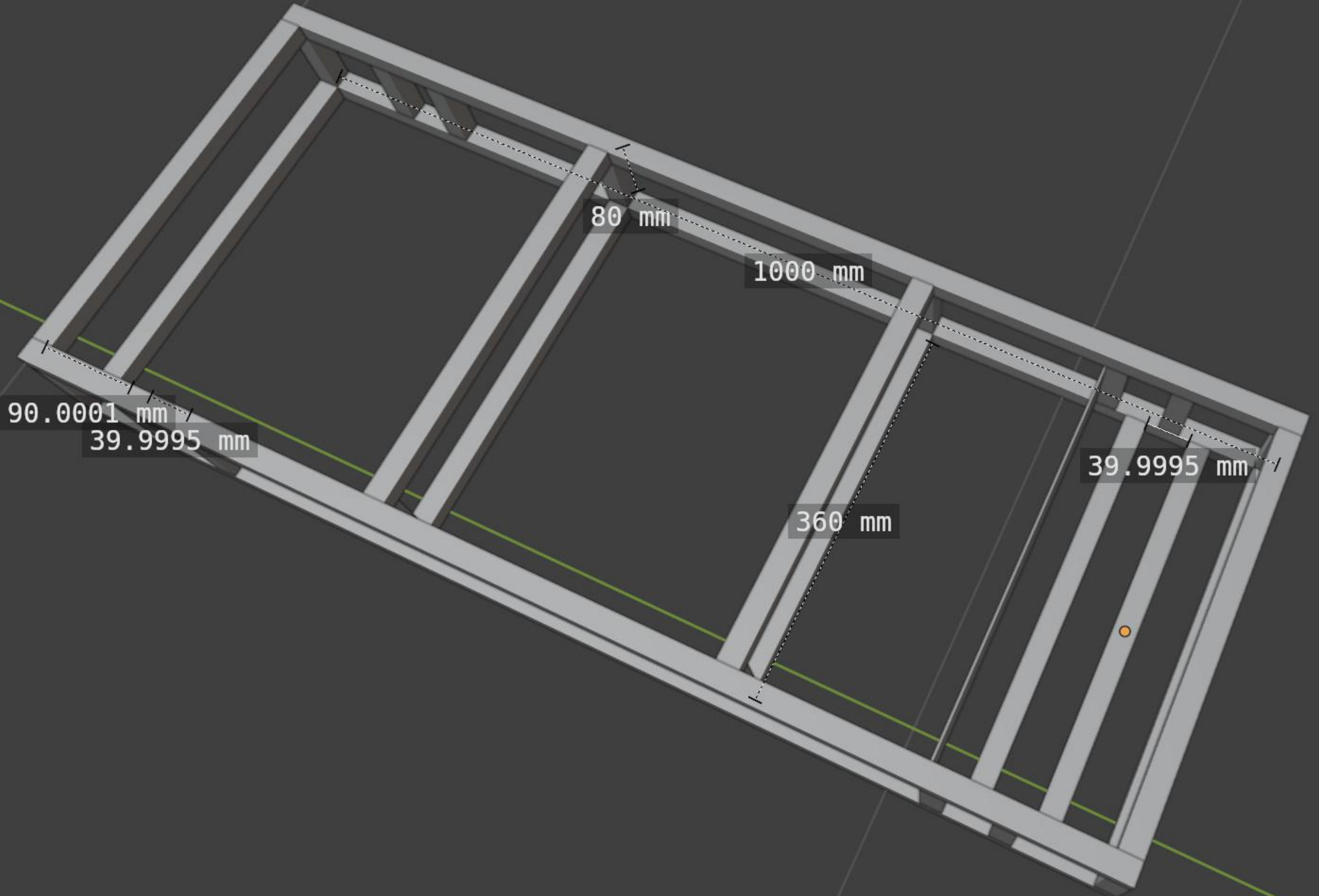
- [Blender files](#)
- material: 2 x 20 mm x 20 mm x 1.5 mm x 6000 mm
- parts:
 - 4 x 1000 mm
 - 10 x 360 mm
 - 16 x 80 mm
 - total: 8520 mm (8.5 m)
 - cut into 4 x (1000 mm + 360 mm = 1360 mm) + 3 x (360 mm + 4 x 80 mm = 680 mm) + (3 x 360 mm + 4 x 80 mm = 1400 mm) \approx 1500 mm



- [v1](#)



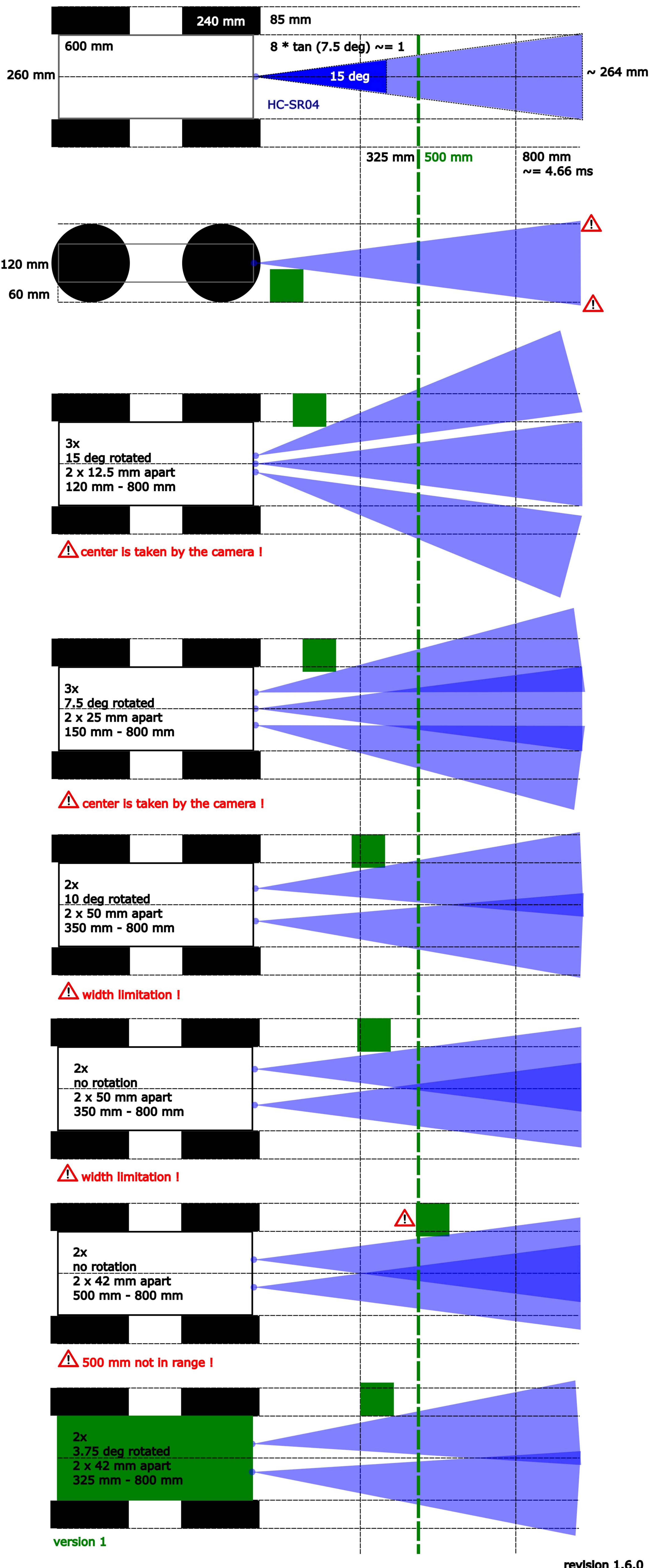




100 mm

time_ms = 2 * distance_mm / 343

1000 mm



swallow: digital: algo

- [driving](#)
- [navigation](#)
- [tracking](#)
- [yolo](#)

swallow: digital: algo: driving

driving [with a keyboard.](#)

@swallow env cp driving

swallow: digital: algo: navigation

navigation using an [@algo/image_classifier](#).

```
@swallow env cp navigation
```

- [dataset](#)
- [model](#)

swallow: digital: algo: navigation: dataset

- [collection](#)
- [review](#)
- [combination](#)



swallow: digital: algo: navigation: dataset: collection

- [validation](#)
- [one](#)



swallow: digital: algo: navigation: dataset: collection: validation

Start swallow, press t (to start training), drive for 5 minutes, press i (to exit).

```
@select
@session start

@select 2025-07-09-10-26-30-itpbmu

@download -
@upload public,zip .
@assets publish \
    extensions=png,push . \
--prefix grid
```

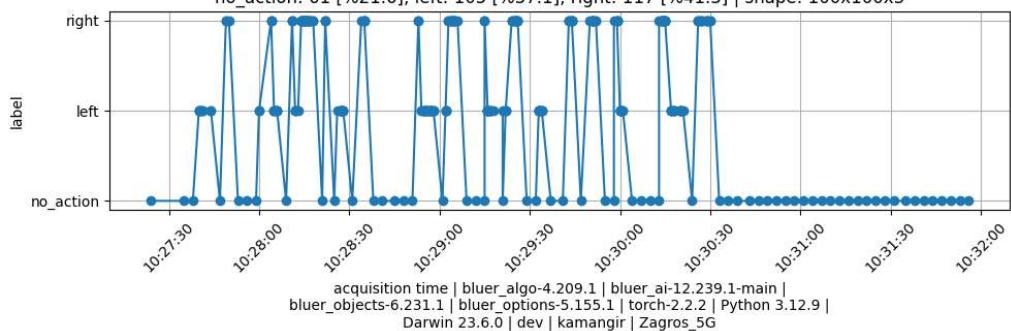
```
2025-07-09-10-26-30-itpbmu/grid.png | 12 July 2025 | 19:36:16 (+0330) |
100x100x3:uint8 | count: 283 | 3 subset(s): train: 283 [%100.0], test: 0 [%0.0],
eval: 0 [%0.0] | 3 class(es): no_action: 61 [%21.6], left: 105 [%37.1], right:
117 [%41.3]
```



```
bluer_algo-4.209.1 | bluer_ai-12.239.1-main | bluer_objects-6.231.1 |
bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
kamangir | Zagros_5G
```

image

2025-07-09-10-26-30-itpbmu | 12 July 2025 | 19:36:16 (+0330) | 283 record(s) | 3
subset(s): train: 283 [%100.0], test: 0 [%0.0], eval: 0 [%0.0] | 3 class(es):
no_action: 61 [%21.6], left: 105 [%37.1], right: 117 [%41.3] | shape: 100x100x3



image

[2025-07-09-10-26-30-itpbmu](#)

```
dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  count: 283
  shape:
    - 100
    - 100
    - 3
  source: 00000000c74cf7d2
  subsets:
    eval: 0
    test: 0
    train: 283
```

**swallow: digital: algo: navigation:
dataset: collection: one**

```
@list log \
    $($@list filter \
        $($@swallow dataset list) \
        --contains $($@today))
```

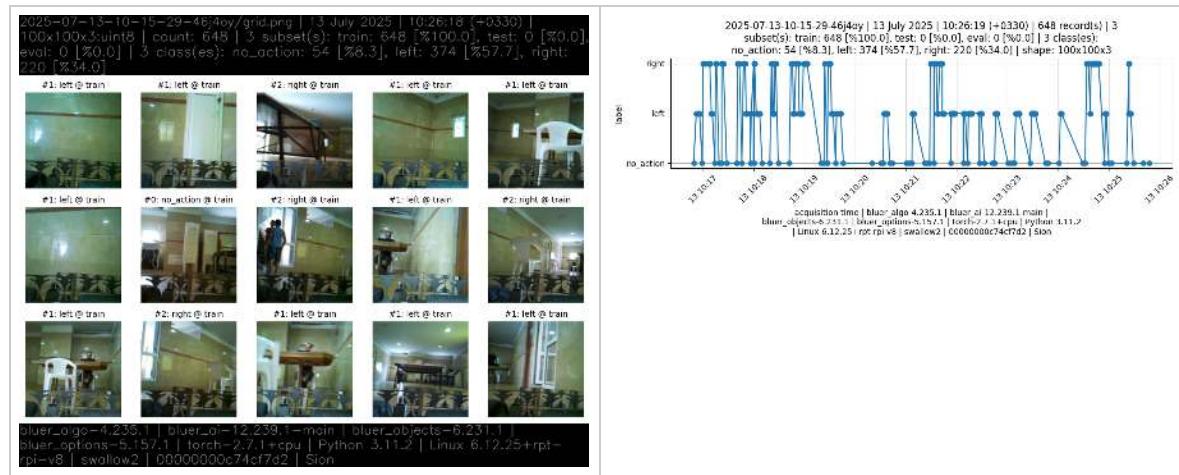
list of 3 item(s): 2025-07-13-10-15-29-46j4oy, 2025-07-13-10-37-12-d4iwpm, 2025-07-13-12-55-54-cx5mhk.

```
runme() {
    local object_name
    for object_name in $(@list filter \
        $($@swallow dataset list) \
        --contains $($@today) | tr , " "); do
        @select $object_name

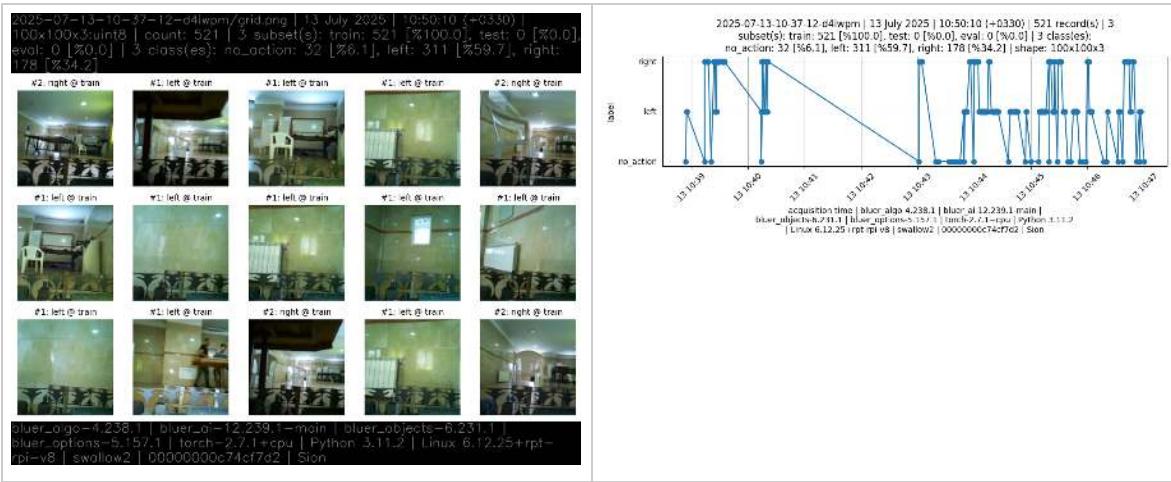
        @download policy=doesnt_exist .
        @upload public,zip .
        @assets publish \
            extensions=png,push .
            --prefix grid
    done
}

runme
```

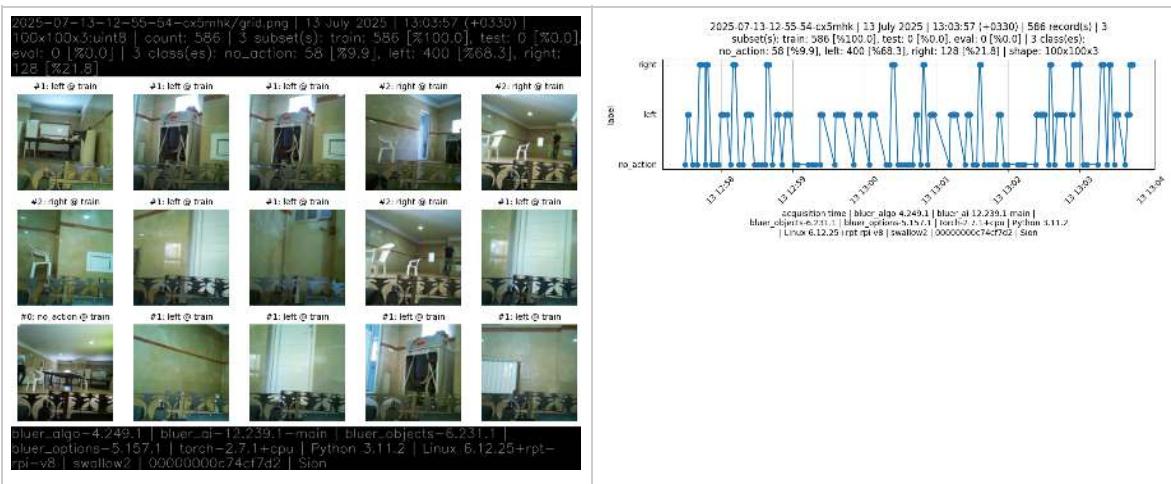
2025-07-13-10-15-29-46j4oy



2025-07-13-10-37-12-d4iwpm



[2025-07-13-12-55-54-cx5mhk](#)



swallow: digital: algo: navigation: dataset: review

```
@select 2025-07-09-10-26-30-itpbmu
@algo image_classifier dataset review - .
@upload public,zip .
@assets publish \
  extensions=png,push . \
  --prefix grid
```

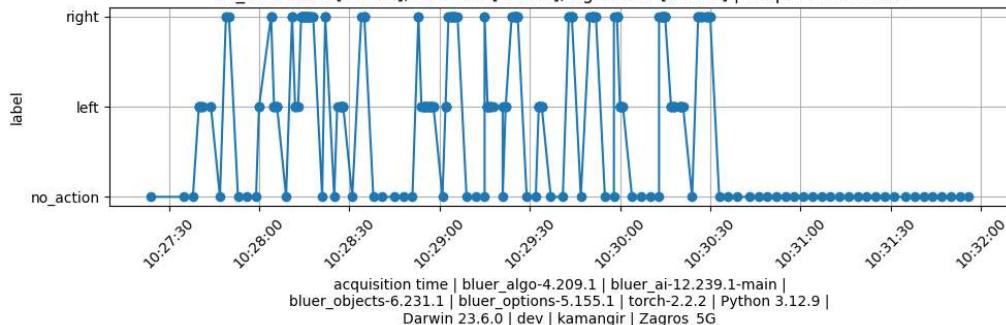
2025-07-09-10-26-30-itpbmu/grid.png | 12 July 2025 | 19:36:16 (+0330) |
 100x100x3:uint8 | count: 283 | 3 subset(s): train: 283 [%100.0], test: 0 [%0.0],
 eval: 0 [%0.0] | 3 class(es): no_action: 61 [%21.6], left: 105 [%37.1], right:
 117 [%41.3]



bluer_algo-4.209.1 | bluer_ai-12.239.1-main | bluer_objects-6.231.1 |
 bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
 kamangir | Zagros_5G

image

2025-07-09-10-26-30-itpbmu | 12 July 2025 | 19:36:16 (+0330) | 283 record(s) | 3
 subset(s): train: 283 [%100.0], test: 0 [%0.0], eval: 0 [%0.0] | 3 class(es):
 no_action: 61 [%21.6], left: 105 [%37.1], right: 117 [%41.3] | shape: 100x100x3



image

2025-07-09-10-26-30-itpbmu

```
dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  count: 283
  shape:
    - 100
    - 100
    - 3
  source: 0000000c74cf7d2
  subsets:
    eval: 0
    test: 0
    train: 283
```

swallow: digital: algo: navigation: dataset: combination

- [validation](#)
- [one](#)

swallow: digital: algo: navigation: dataset: combination: validation

uses [collection/validation](#).

```
@select swallow-dataset-$(@timestamp)
```

```
@swallow dataset combine \
  count=2 .
```

```
@upload public,zip .
@assets publish \
  extensions=png,push . \
  --prefix grid
```



swallow-dataset-2025-07-11-13-03-58-aoadib/grid.png | 11 July 2025 | 13:04:06
 (+0330) | 100x100x3:uint8 | count: 1801 | 3 subset(s): train: 1441 [%80.0],
 test: 178 [%9.9], eval: 182 [%10.1] | 3 class(es): no_action: 100 [%5.6], left:
 952 [%52.9], right: 749 [%41.6]



bluer_algo-4.199.1 | bluer_ai-12.239.1-wifi-fix-2025-07-09-oxrieh |
 bluer_objects-6.231.1 | bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 |
 Darwin 23.6.0 | dev.local | kamangir | Zagros_5G

image

[swallow-dataset-2025-07-11-13-03-58-aoadib](#)

```
dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  contains:
  - 2025-07-09-11-16-52-4zo4zc
  - 2025-07-09-11-34-19-bcoh75
  count: 1801
  shape:
  - 100
  - 100
  - 3
  subsets:
    eval: 182
    test: 178
    train: 1441
```

swallow: digital: algo: navigation: dataset: combination: one

uses [collection/one](#).

```
@select swallow-dataset-$(@timestamp)

@swallow dataset combine \
    sequence=3 . \
    --datasets $($list filter \
        $($swallow dataset list) \
        --contains 2025-07-13)

@upload public,zip .
@assets publish \
    extensions=png,push . \
    --prefix grid
```

```
swallow-dataset-2025-07-14-09-39-22-bfm9sx/grid.png | 14 July 2025 | 09:39:34
(+0330) | 100x300x3:uint8 | count: 1749 | 3 subset(s): train: 1411 [%80.7],
test: 153 [%8.7], eval: 185 [%10.6] | 3 class(es): no_action: 141 [%8.1], left:
1082 [%61.9], right: 526 [%30.1]
```



```
bluer_algo-4.250.1 | bluer_ai-12.242.1-main | bluer_objects-6.233.1 |
bluer_options-5.159.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev.local
| kamangir | Zagros
```

image

[swallow-dataset-2025-07-14-09-39-22-bfm9sx](#)

```
dataset:
class_count: 3
classes:
0: no_action
1: left
2: right
contains:
- 2025-07-13-10-15-29-46j4oy
- 2025-07-13-10-37-12-d4iwpm
- 2025-07-13-12-55-54-cx5mhk
count: 1749
shape:
```

```
- 100
- 300
- 3
subsets:
  eval: 185
  test: 153
  train: 1411
```

swallow: digital: algo: navigation: model

- [validation](#)
- [one](#)

swallow: digital: algo: navigation: model: validation

uses [combination/validation](#).

```
@select swallow-dataset-$(@timestamp)

@swallow dataset combine \
  count=2 .

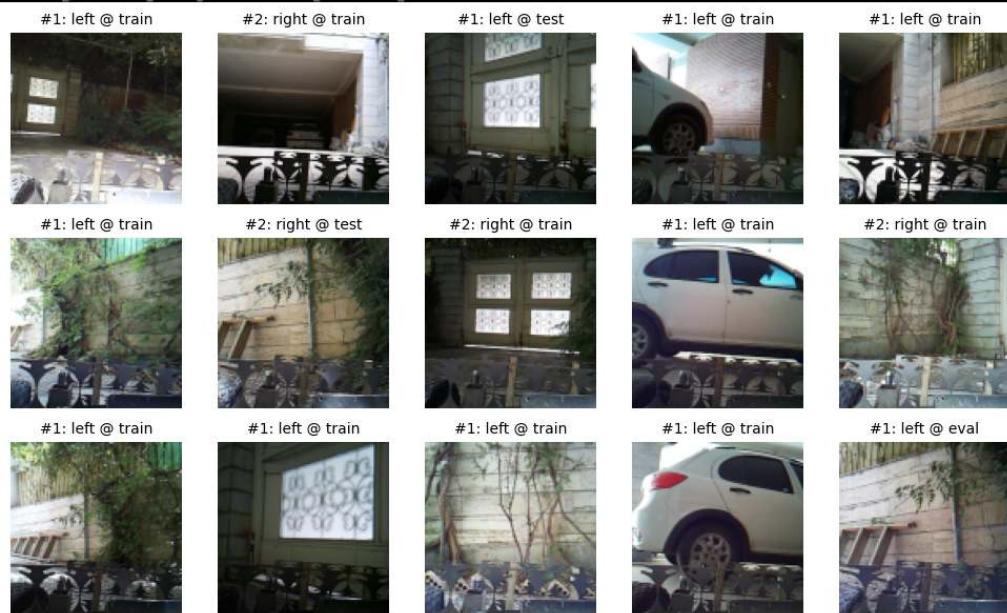
@upload public,zip .
@assets publish \
  extensions=png,push . \
  --prefix grid

@select swallow-model-$(@timestamp)

@image_classifier model train upload . . .

@upload public,zip .
@assets publish \
  extensions=png,push .
```

swallow-dataset-2025-07-11-13-05-02-u4z1ea/grid.png | 11 July 2025 | 13:05:16
(+0330) | 100x100x3:uint8 | count: 1801 | 3 subset(s): train: 1482 [%82.3],
test: 166 [%9.2], eval: 153 [%8.5] | 3 class(es): no_action: 100 [%5.6], left:
952 [%52.9], right: 749 [%41.6]



bluer_algo-4.199.1 | bluer_ai-12.239.1-main | bluer_objects-6.231.1 |
bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
kamangir | Sion

image

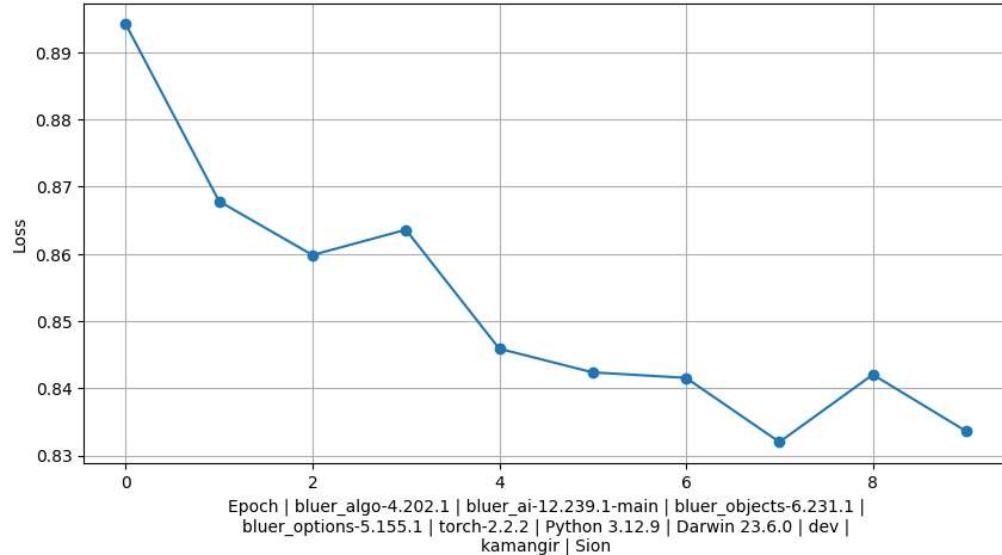
[swallow-dataset-2025-07-11-13-05-02-u4z1ea](#)

```

dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  contains:
  - 2025-07-09-11-16-52-4zo4zc
  - 2025-07-09-11-34-19-bcoh75
  count: 1801
  shape:
  - 100
  - 100
  - 3
  subsets:
    eval: 153
    test: 166
    train: 1482

```

swallow-dataset-2025-07-11-13-05-02-u4z1ea | 11 July 2025 | 15:05:08 (+0330) |
 1801 record(s) | 3 subset(s): train: 1482 [%82.3], test: 166 [%9.2], eval: 153
 [%8.5] | 3 class(es): no_action: 100 [%5.6], left: 952 [%52.9], right: 749
 [%41.6] | shape: 100x100x3 | batch_size: 16 | num_epochs: 10 | eval_accuracy:
 65.36% | model: swallow-model-2025-07-11-15-04-03-2glcch



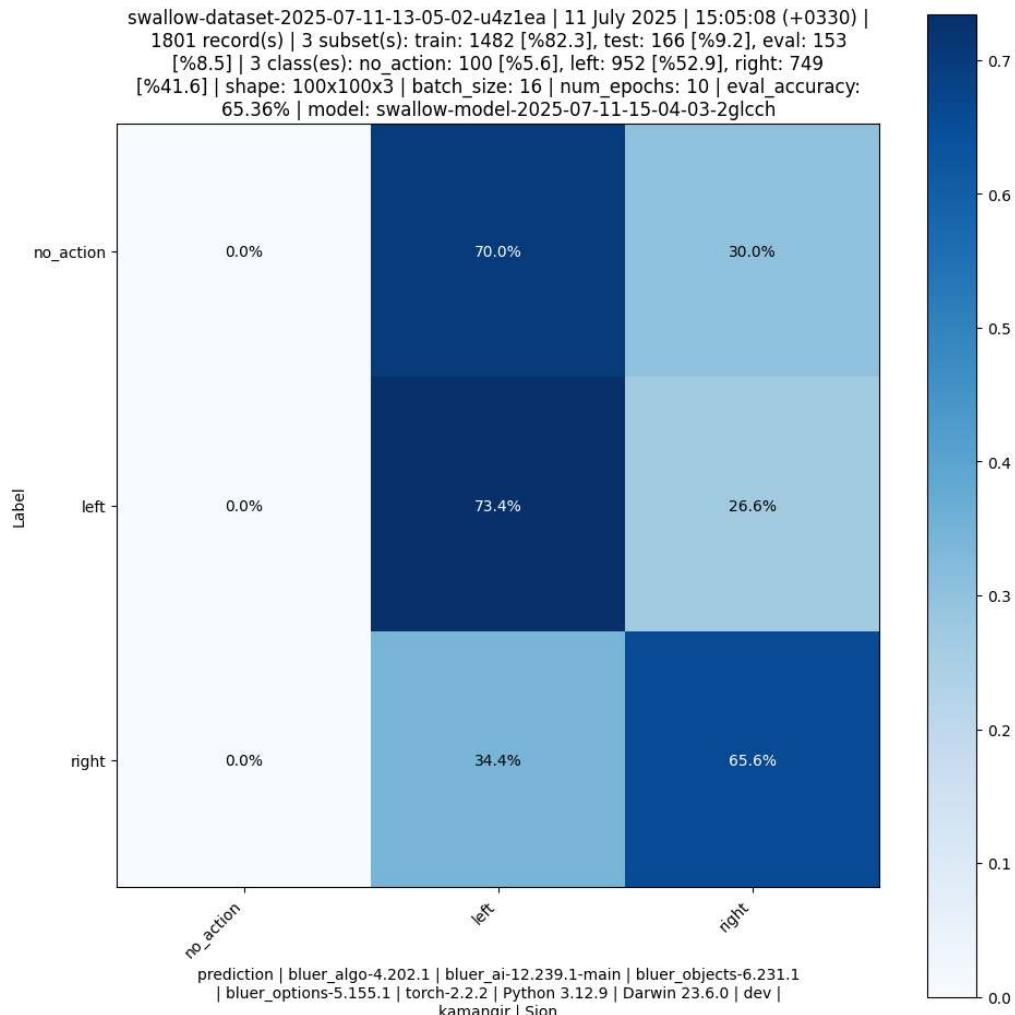
image

```
swallow-model-2025-07-11-15-04-03-2glcch/evaluation.png | 11 July 2025 |
15:05:08 (+0330) | - | swallow-dataset-2025-07-11-13-05-02-u4z1eo | 11 July 2025
| 15:05:08 (+0330) | 1801 record(s) | 3 subset(s): train: 1482 [%82.3], test:
166 [%9.2], eval: 153 [%8.5] | 3 class(es): no_action: 100 [%5.6], left: 952
[%52.9], right: 749 [%41.6] | shape: 100x100x3 | batch_size: 16 | num_epochs: 10
eval_accuracy: 65.36%
```



```
bluer_algo-4.202.1 | bluer_ai-12.239.1-main | bluer_objects-6.231.1 |
bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
kamangir | Sion
```

image



image

[swallow-model-2025-07-11-15-04-03-2glcch](#)

```

model:
dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  count: 1801
  shape:
    - 100
    - 100
    - 3
evaluation:
  class_accuracy:
    0: 0.0
    1: 0.7341772151898734
    2: 0.65625
  eval_accuracy: 0.6535947712418301
inputs:
  batch_size: 16
  num_epochs: 10
training:
  loss:
    - 0.8941771462861343
    - 0.8678298002956045
    - 0.8598417815891838

```

- 0.863602487181845
- 0.8459089610740723
- 0.8423866080208186
- 0.8415681831588951
- 0.8320272445035206
- 0.8420564680286103
- 0.8336275264962643

swallow: digital: algo: navigation: model: one

uses [combination/one](#).

```
@arvan ssh <ip-address>
@arvan seed
# Ctrl+V

@select swallow-dataset-$(@timestamp)

@swallow dataset combine \
    sequence=3 . \
    --datasets $($list filter \
        $($swallow dataset list) \
        --contains 2025-07-13)

@upload filename=metadata.yaml .
@assets publish \
    extensions=png,push . \
    --prefix grid

@select swallow-model-$(@timestamp)

@image_classifier model train upload . . \
    --num_epochs 100

@upload public,zip .
@assets publish \
    extensions=png,push .

@select swallow-prediction-test-$(@timestamp)

@algo image_classifier model prediction_test \
    upload . . .

@assets publish \
    extensions=png,push .
```

swallow-dataset-2025-07-14-13-16-51-ajhuvd/grid.png | 14 July 2025 | 13:17:49
 (Iran) | 100x300x3:uint8 | count: 1749 | 3 subset(s): train: 1380 [%78.9], test: 185 [%10.6], eval: 184 [%10.5] | 3 class(es): no_action: 141 [%8.1], left: 1082 [%61.9], right: 526 [%30.1]

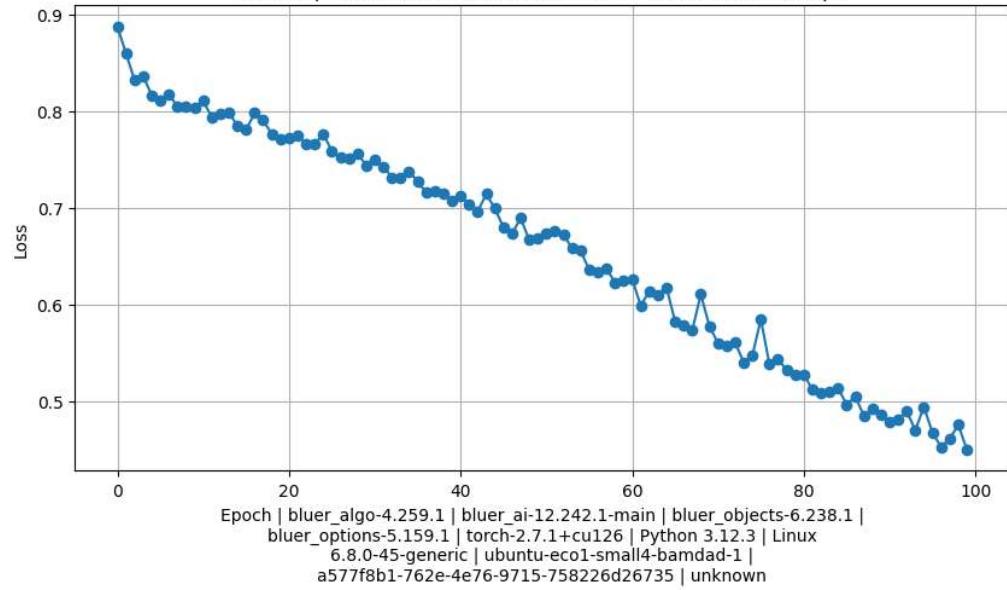


bluer_algo-4.259.1 | bluer_ai-12.242.1-main | bluer_objects-6.238.1 |
 bluer_options-5.159.1 | torch-2.7.1+cu126 | Python 3.12.3 | Linux
 6.8.0-45-generic | ubuntu-eco1-small4-bamdad-1 |
 a577f8b1-762e-4e76-9715-758226d26735 | unknown

image

► metadata

swallow-dataset-2025-07-14-13-16-51-ajhuvd | 14 July 2025 | 14:11:19 (Iran) |
 1749 record(s) | 3 subset(s): train: 1380 [%78.9], test: 185 [%10.6], eval: 184
 [%10.5] | 3 class(es): no_action: 141 [%8.1], left: 1082 [%61.9], right: 526
 [%30.1] | shape: 100x300x3 | batch_size: 16 | num_epochs: 100 | eval_accuracy:
 79.35% | model: swallow-model-2025-07-14-13-18-10-kx0qrw



image

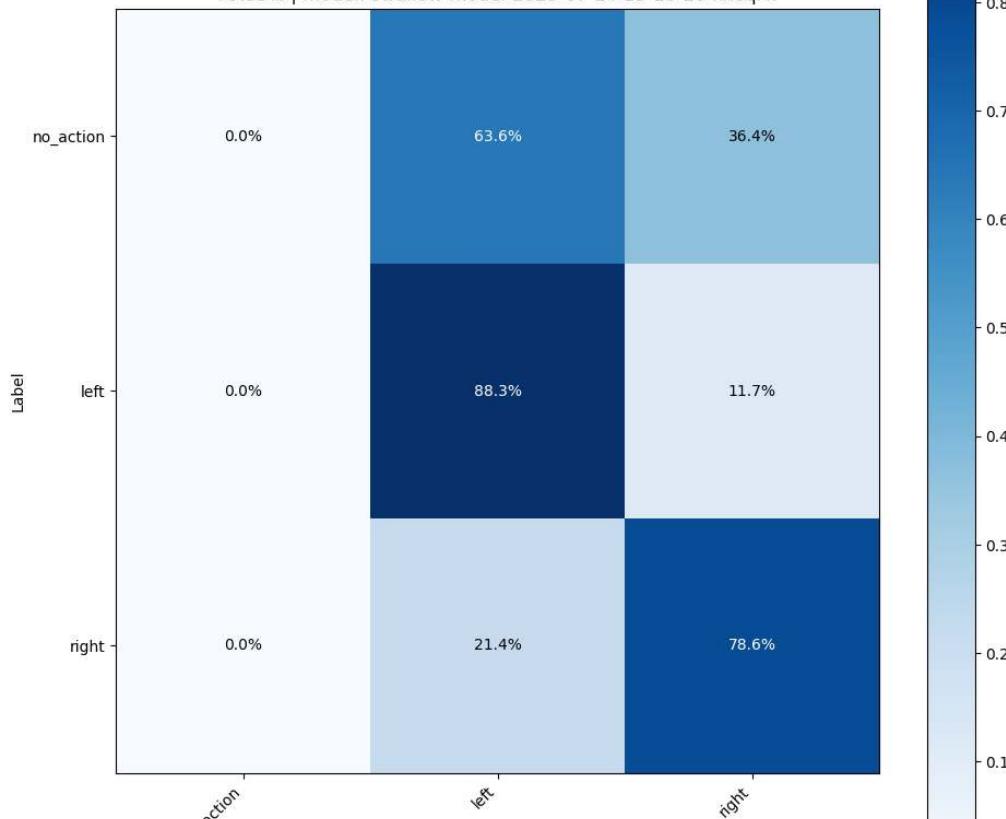
```
swallow-model-2025-07-14-13-18-10-kx0qrw/evaluation.png | 14 July 2025 |
14:11:23 (Iran) | - | swallow-dataset-2025-07-14-13-16-51-ajhuvd | 14 July 2025
| 14:11:19 (Iran) | 1749 record(s) | 3 subset(s): train: 1380 [%78.9], test: 185
[%10.6], eval: 184 [%10.5] | 3 class(es): no_action: 141 [%8.1], left: 1082
[%61.9], right: 526 [%30.1] | shape: 100x300x3 | batch_size: 16 | num_epochs:
100 | eval_accuracy: 79.35%
```



```
bluer_algo-4.259.1 | bluer_ai-12.242.1-main | bluer_objects-6.238.1 |
bluer_options-5.159.1 | torch-2.7.1+cu126 | Python 3.12.3 | Linux
6.8.0-45-generic | ubuntu-eco1-small4-bamdad-1 |
a577f8b1-762e-4e76-9715-758226d26735 | unknown
```

image

```
swallow-dataset-2025-07-14-13-16-51-ajhuvd | 14 July 2025 | 14:11:19 (Iran) |
1749 record(s) | 3 subset(s): train: 1380 [%78.9], test: 185 [%10.6], eval: 184
[%10.5] | 3 class(es): no_action: 141 [%8.1], left: 1082 [%61.9], right: 526
[%30.1] | shape: 100x300x3 | batch_size: 16 | num_epochs: 100 | eval_accuracy:
79.35% | model: swallow-model-2025-07-14-13-18-10-kx0qrw
```



image

[swallow-model-2025-07-14-13-18-10-kx0qrw](#)

► metadata

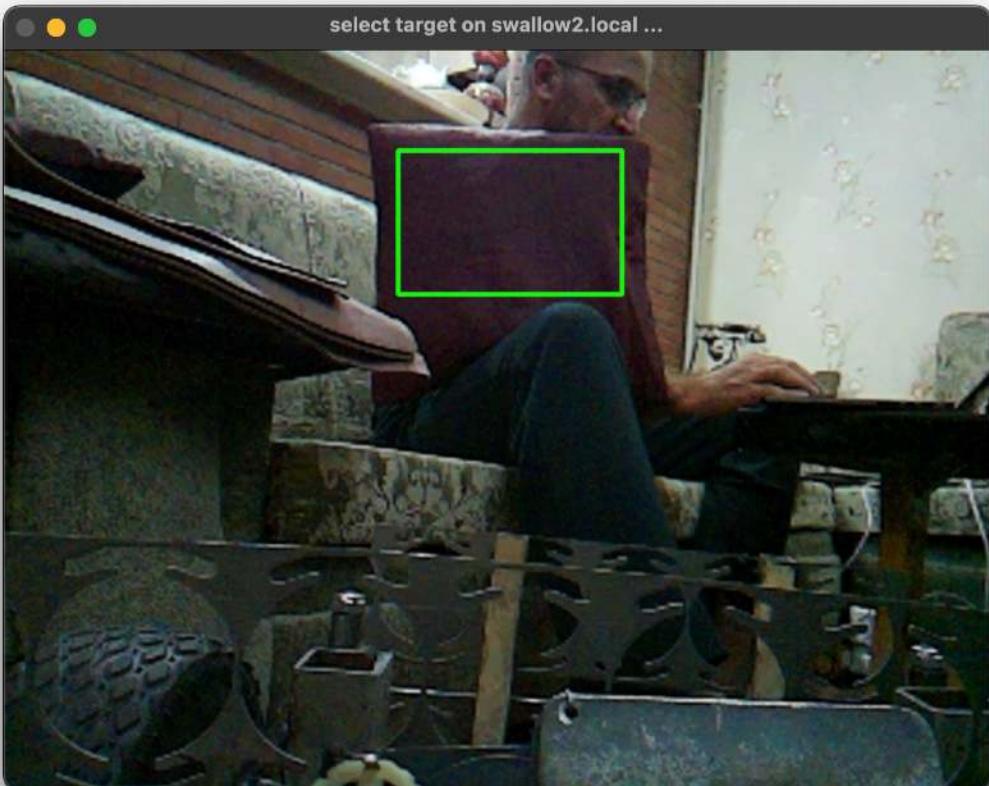
swallow-prediction-test-2025-07-14-14-13-57-ngywj1 | 14 July 2025 | 14:14:30
(Iran) | model: swallow-model-2025-07-14-13-18-10-kx0qrw | 3 class(es): #0:
no_action, #1: left, #2: right | shape: 100x300x3 | prediction: right [#2] |
correct | took 398 ms | bluer_algo-4.259.1 | bluer_ai-12.242.1-main |
bluer_objects-6.238.1 | bluer_options-5.159.1 | torch-2.7.1+cu126 | Python
3.12.3 | Linux 6.8.0-45-generic | ubuntu-ecol-small4-bamdad-1 |
a577f8b1-762e-4e76-9715-758226d26735 | unknown



image

► metadata

swallow: digital: algo: tracking



target tracking using an [@algo/tracker](#).

```
@swallow env cp tracking
```

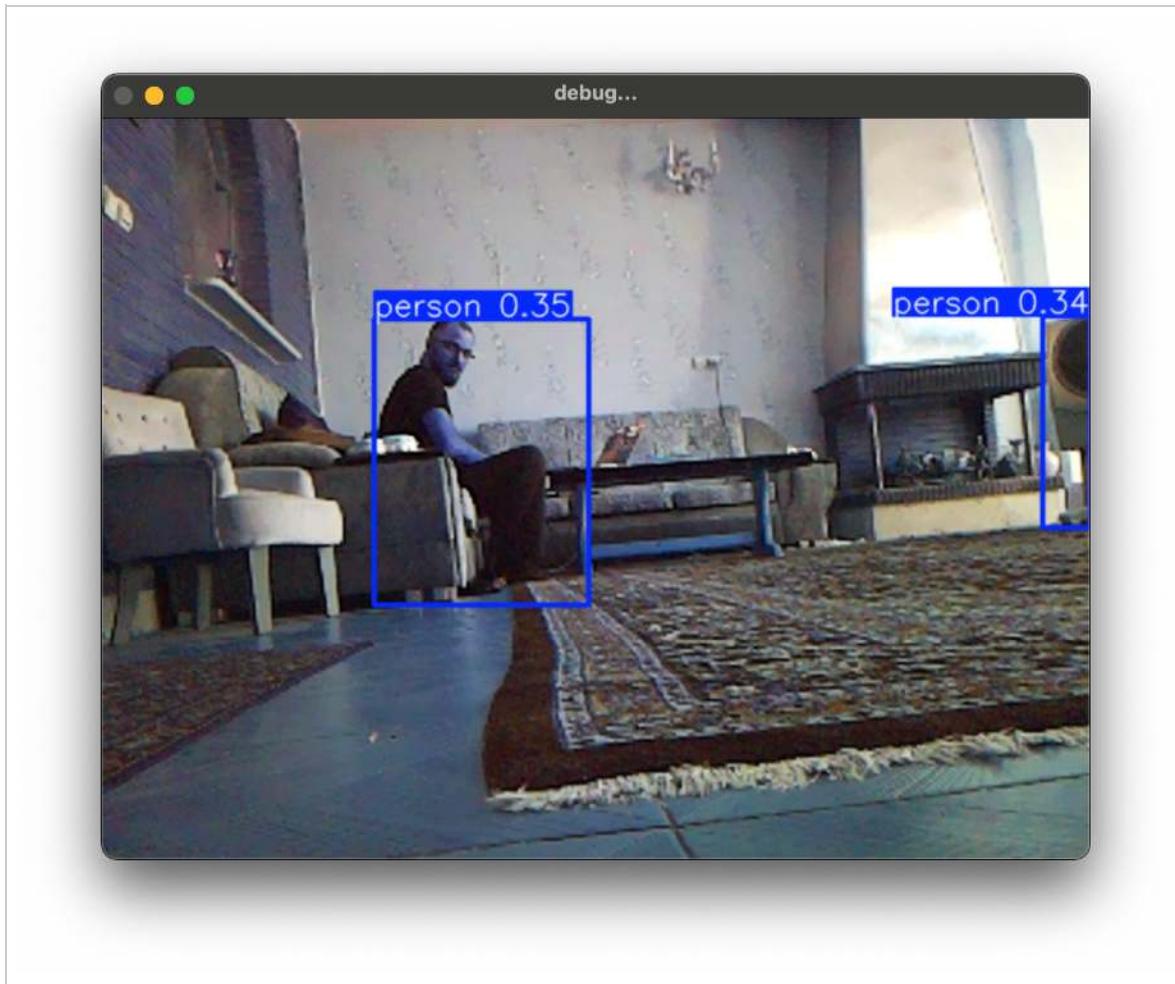
swallow: digital: algo: yolo

target tracking using an [@algo/yolo](#).

```
@swallow env cp yolo
```

- [training a 256x256 model](#)

```
@swallow debug
```



swallow: digital: algo: yolo: train

training a 256 x 256 model.

ingest

```
@select coco128-$(@@timestamp)

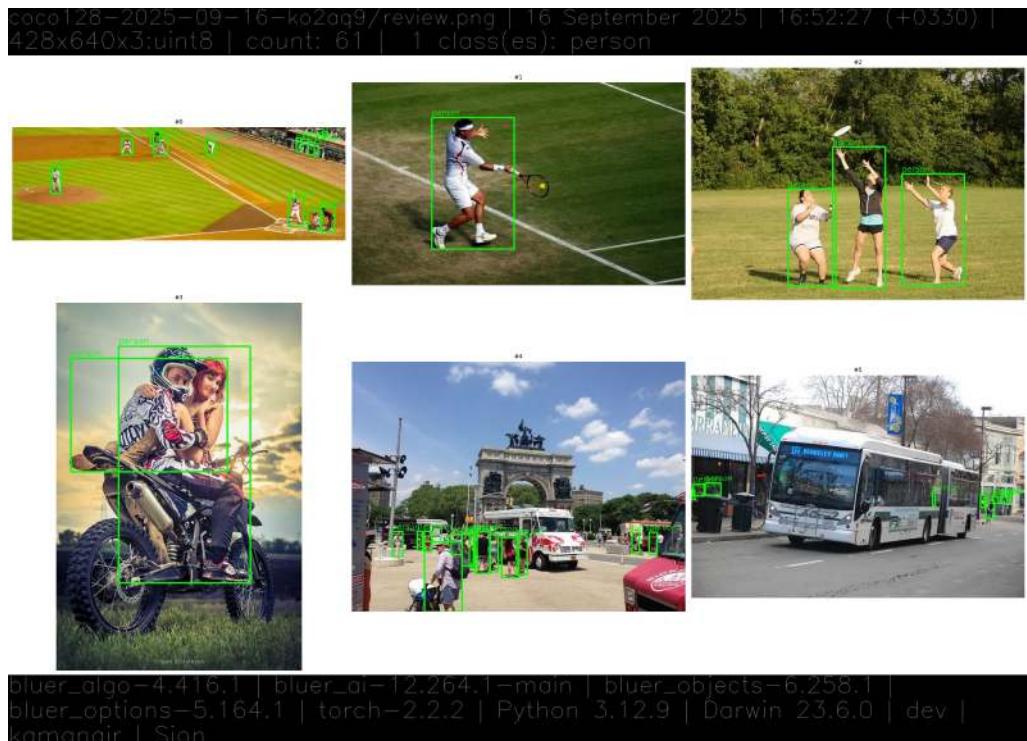
@yolo dataset ingest - . \
--classes person

@yolo dataset review \
~download .

@upload public,zip

@assets publish \
extensions=png,push .
```

[coco128-2025-09-16-ko2aq9](#)



image

```
dataset:
  count: 61
names:
  0: person
source: coco_128
train: coco128/images/train2017
val: coco128/images/train2017
```

train

```
@select coco128-model-$(@@timestamp)
```

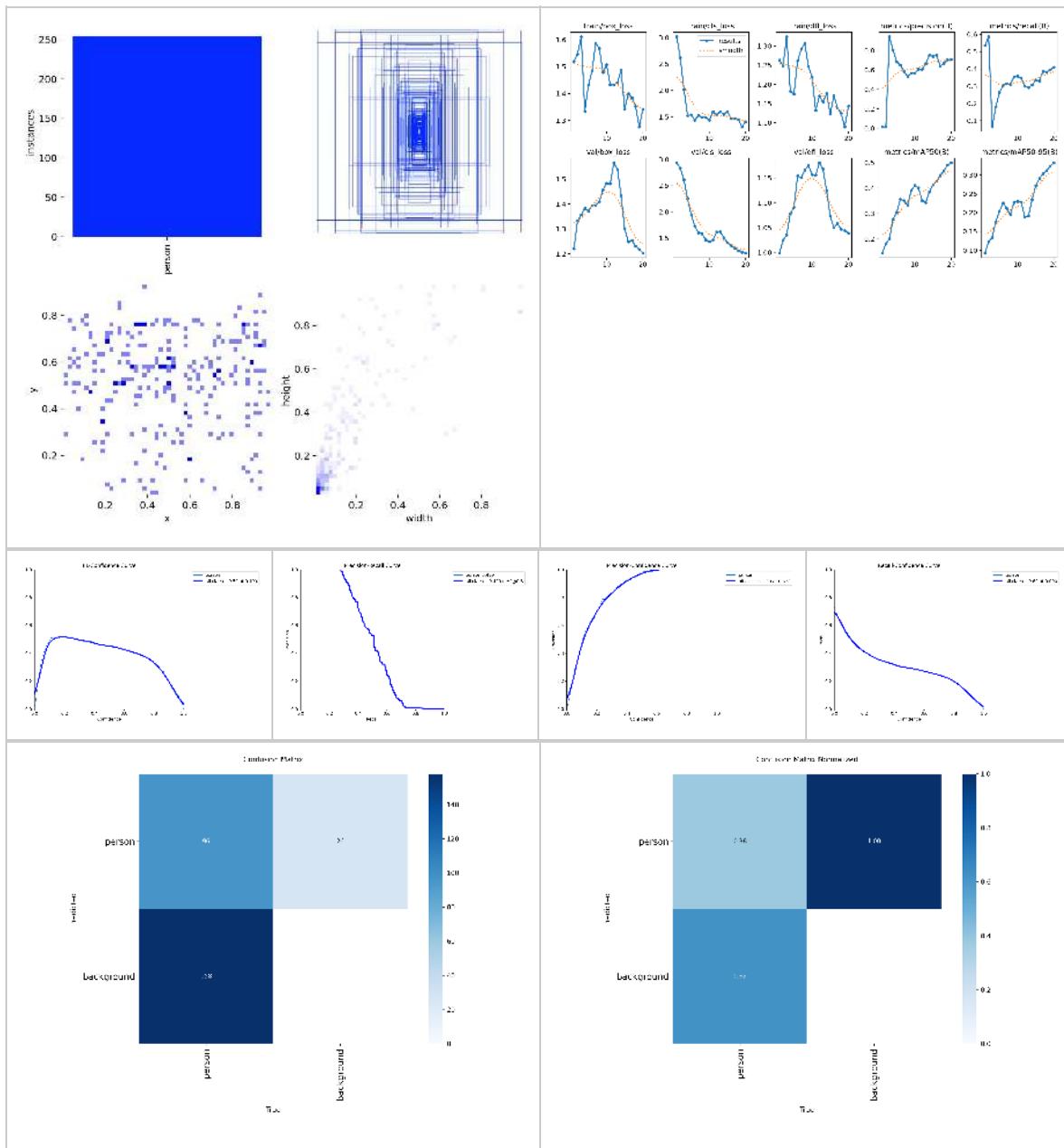
```
@yolo model train \
~download,upload ... \
--epochs 20 \
--image_size 256
```

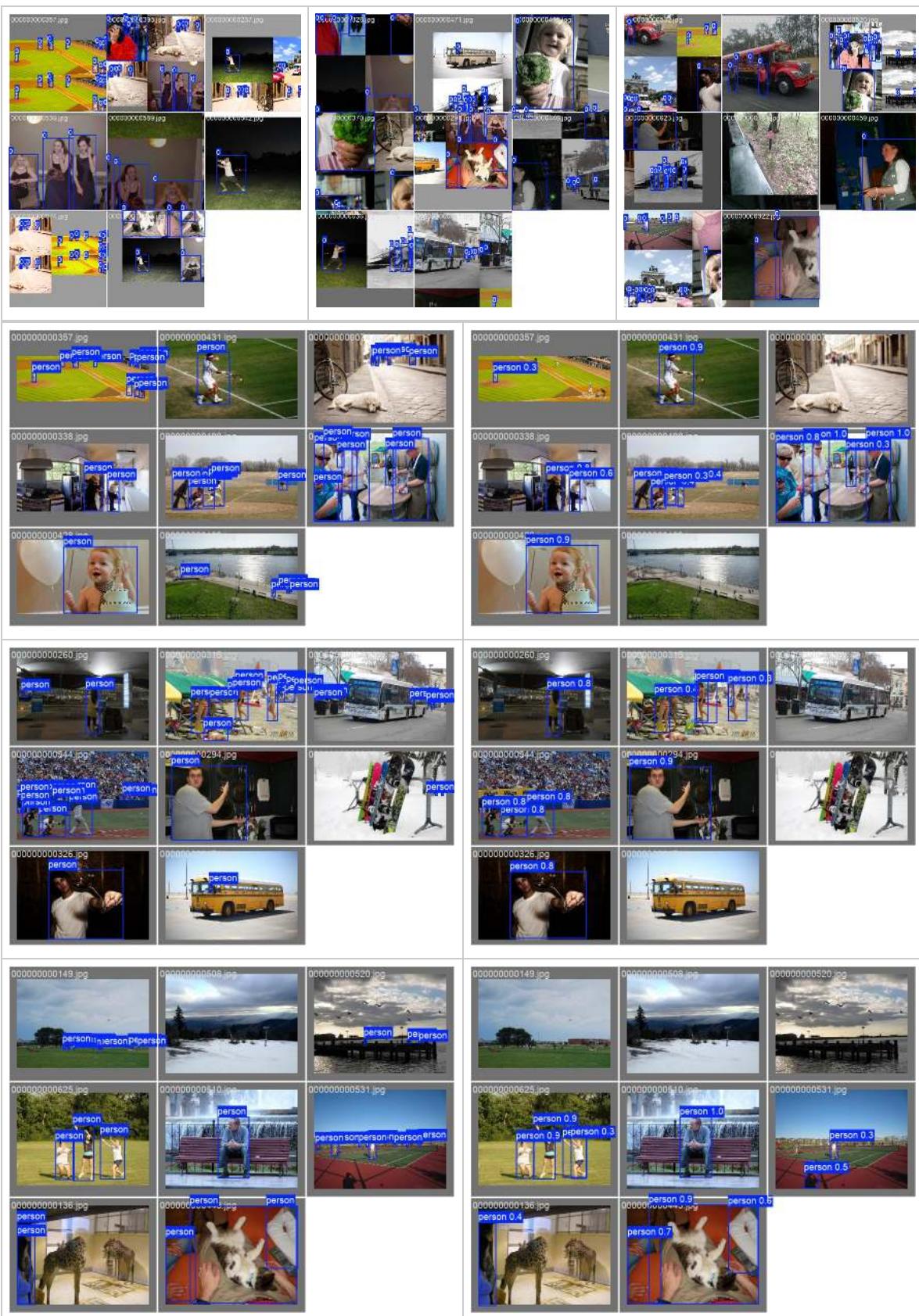
```
@upload public,zip
```

```
@assets publish \
extensions=jpg+png . \
--prefix train/
```

```
@assets publish \
extensions=jpg+png,push . \
--prefix validation/
```

[coco128-model-2025-09-16-meb4if](#)





► metadata

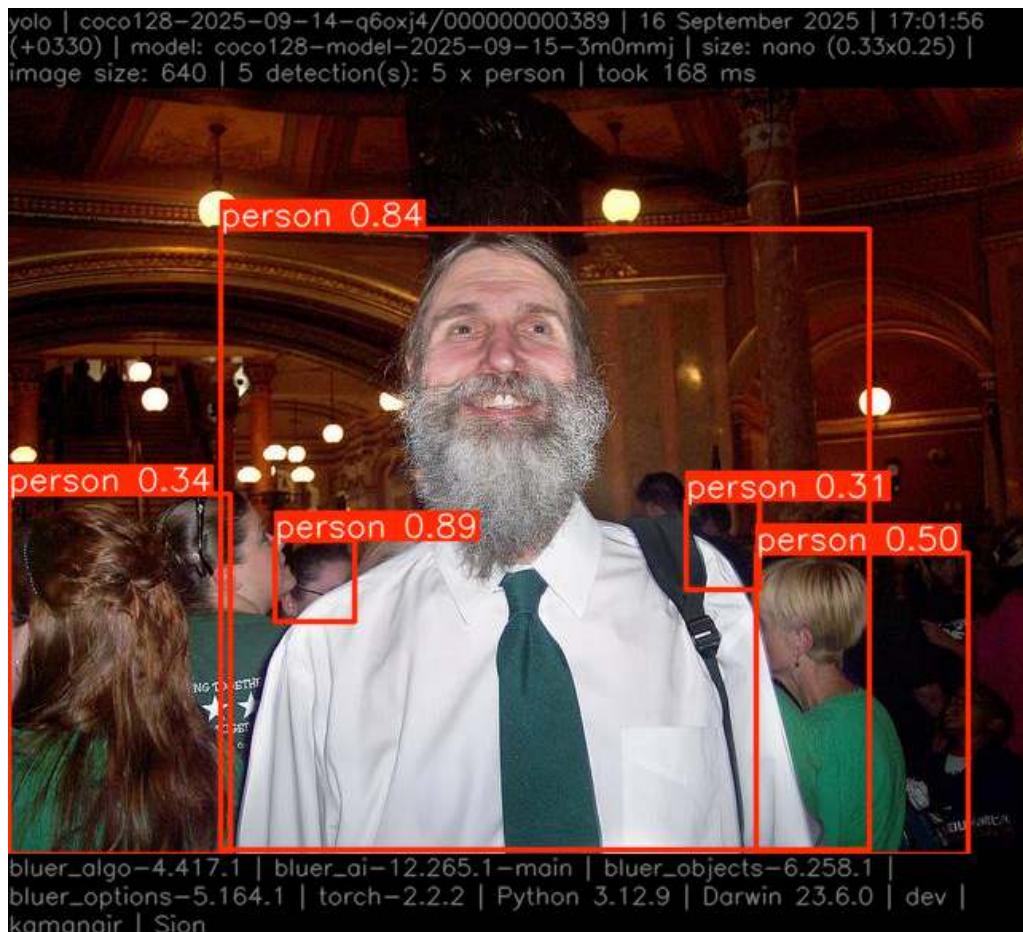
predict

```
@select yolo-prediction-test-$(@timestamp)
```

```
@yolo model prediction_test \
```

```
upload \
$BLUER_ALGO_COCO128_TEST_DATASET \
$BLUER_UGV_SWALLOW_YOLO_MODEL . \
--record_index 3

@assets publish extensions=png,push
```



image

```
000000000389:
detections:
- bbox_xyxy:
  - 167.85377502441406
  - 283.70269775390625
  - 217.58558654785156
  - 334.81072998046875
  class_id: 0
  confidence: 0.8922585248947144
  label: person
- bbox_xyxy:
  - 133.1409912109375
  - 88.43701171875
  - 540.6605834960938
  - 477.65869140625
  class_id: 0
  confidence: 0.842871367931366
  label: person
- bbox_xyxy:
  - 469.0397644042969
  - 291.564697265625
  - 602.921875
  - 480.0
  class_id: 0
```

```

confidence: 0.49844595789909363
label: person
- bbox_xyxy:
  - 0.58984375
  - 254.95266723632812
  - 139.97897338867188
  - 480.0
class_id: 0
confidence: 0.33697929978370667
label: person
- bbox_xyxy:
  - 425.3193359375
  - 258.2589416503906
  - 471.283935546875
  - 314.4782409667969
class_id: 0
confidence: 0.31127116084098816
label: person
elapsed_time: 0.16768479347229004
image_size:
- 480
- 640

```

predict (256)

```

@select yolo-prediction-test-$(@timestamp)

@yolo model prediction_test \
upload \
$BLUER_ALGO_COCO128_TEST_DATASET \
$BLUER_UGV_SWALLOW_YOLO_MODEL . \
--record_index 3 \
--image_size 256

@assets publish extensions=png,push

```

yolo | coco128-2025-09-14-q6oxj4/000000000389 | 16 September 2025 | 17:02:54
 (+0330) | model: coco128-model-2025-09-15-3m0mmj | size: nano (0.33x0.25) |
 image size: 256 | 1 detection(s): 1 x person | took 022 ms



bluer_algo-4.417.1 | bluer_oi-12.265.1-main | bluer_objects-6.258.1 |
 bluer_options-5.164.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
 kamangir | Sion

image

```
000000000389:  

  detections:  

  - bbox_xyxy:  

    - 135.89393615722656  

    - 100.03055572509766  

    - 601.81640625  

    - 480.0  

    class_id: 0  

    confidence: 0.8894820213317871  

    label: person  

  elapsed_time: 0.022478818893432617  

  image_size:  

  - 480  

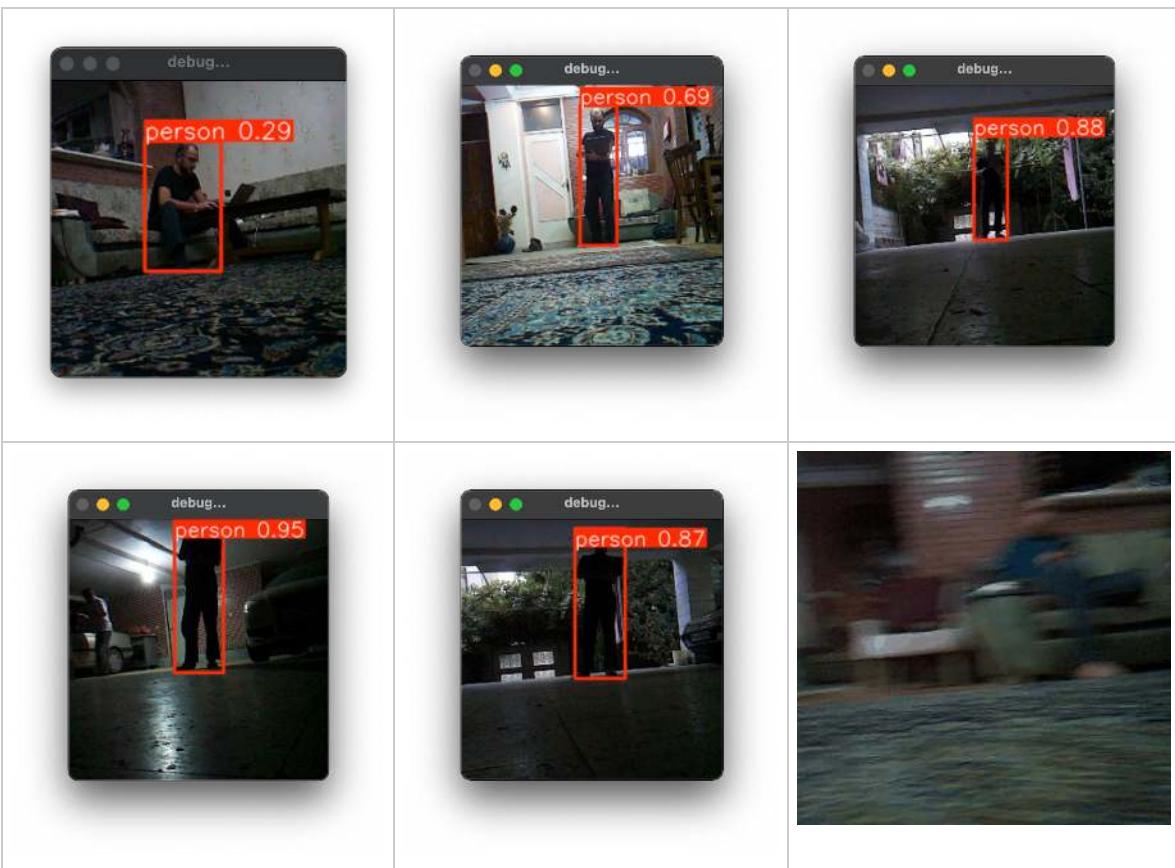
  - 640
```

predict (rpi)

```
@select swallow-debug-$(@timestamp)  

@swallow debug .  

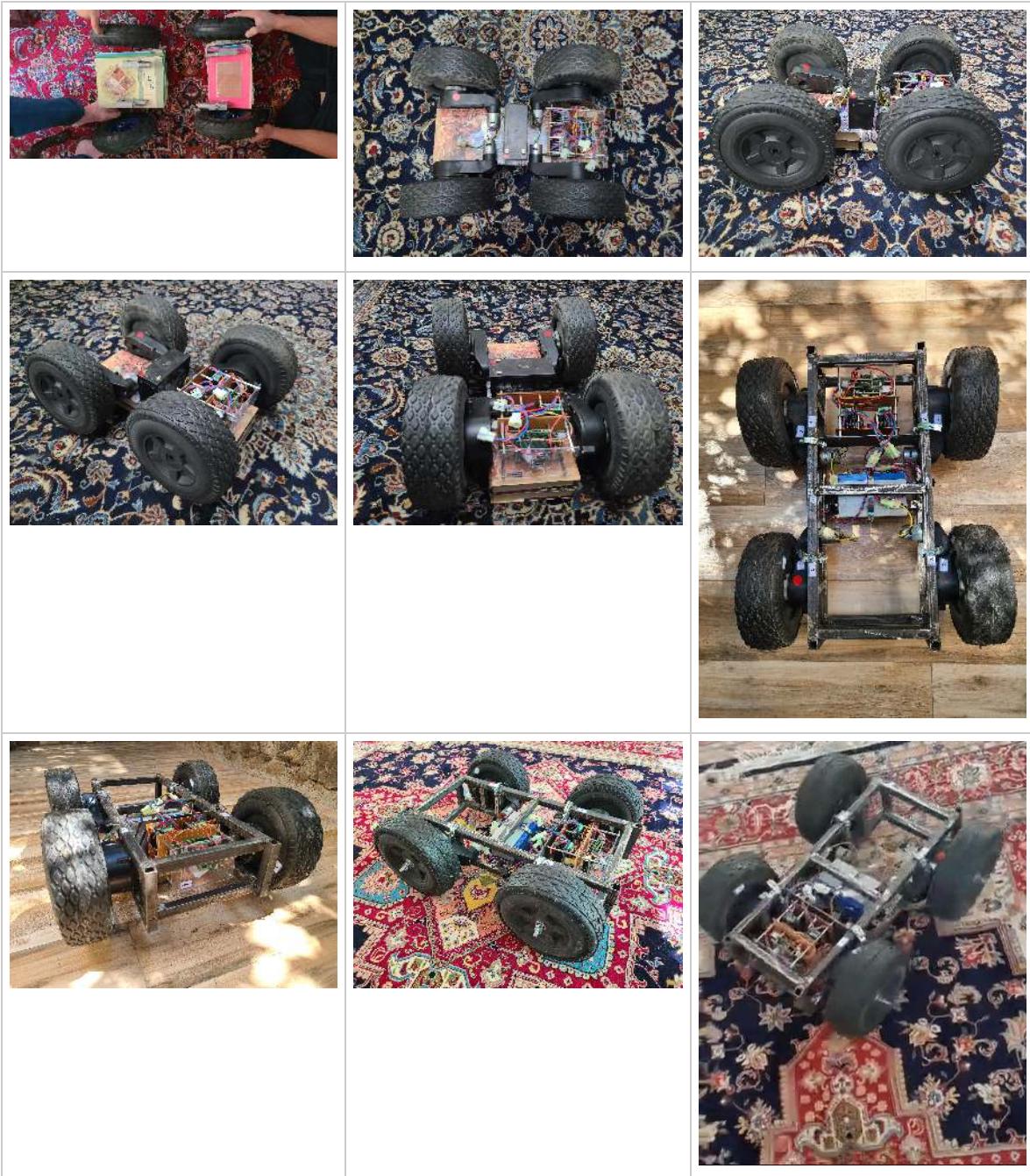
@assets publish extensions=gif,push
```



arzhang

[swallow](#)'s little sister, formerly known as sparrow, arzhang (ارزنگ) is named after the demon in the Shahnameh. It embodies noble strength and quiet defense.

- [design](#)
- [algo](#)





arzhang: design

- [specs](#)
- [mechanical](#)
- [parts](#)
- [steering setpoint](#)

arzhang: design: specs

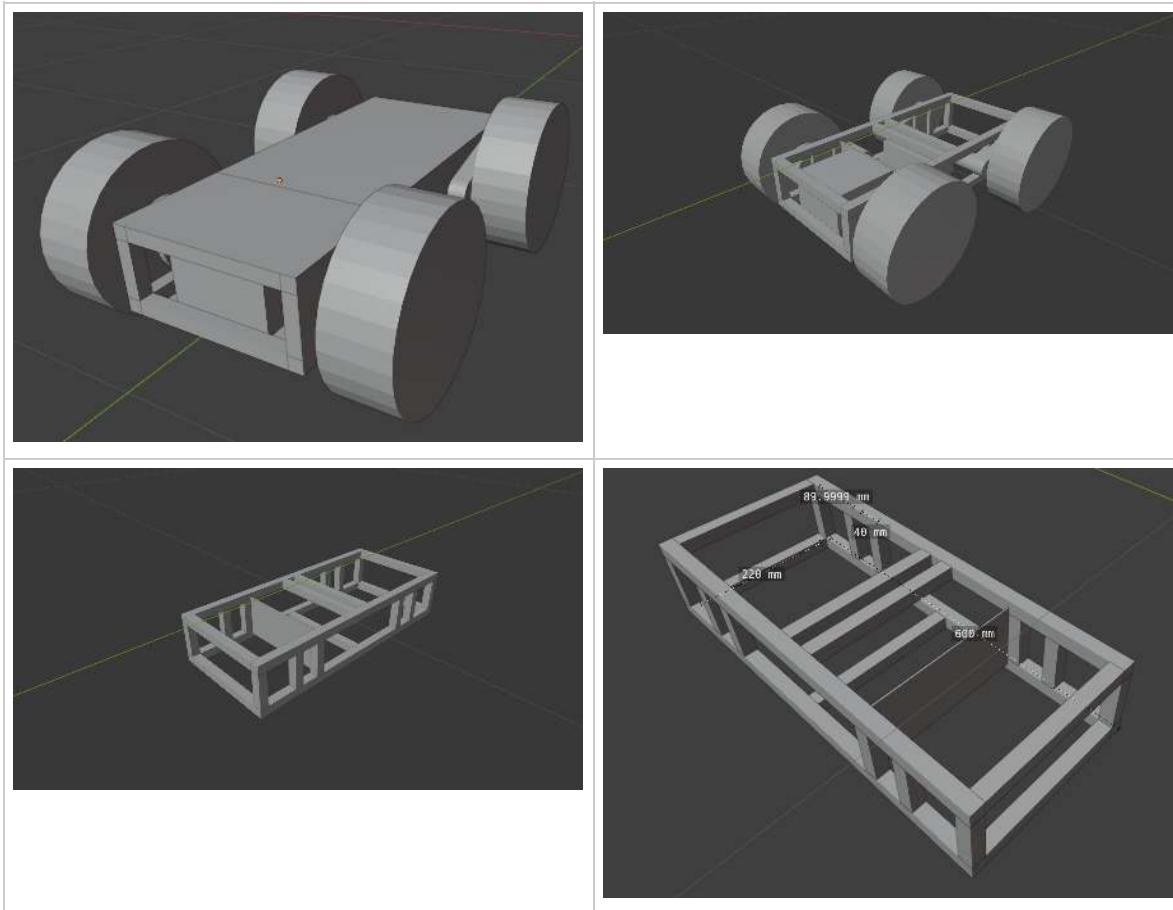
- dimensions: 300 mm x 600 mm
- speed: 10 km / h (max)
- weight: ~15 kg
- wheel diameter: 20 cm (270 RPM) - 30 cm (180 RPM)
- motor + gearbox: metal, 90-degree, 180 - 270 RPM, 12 V, 10 kg cm torque.
- drives: 50 – 100 W

speed calculations

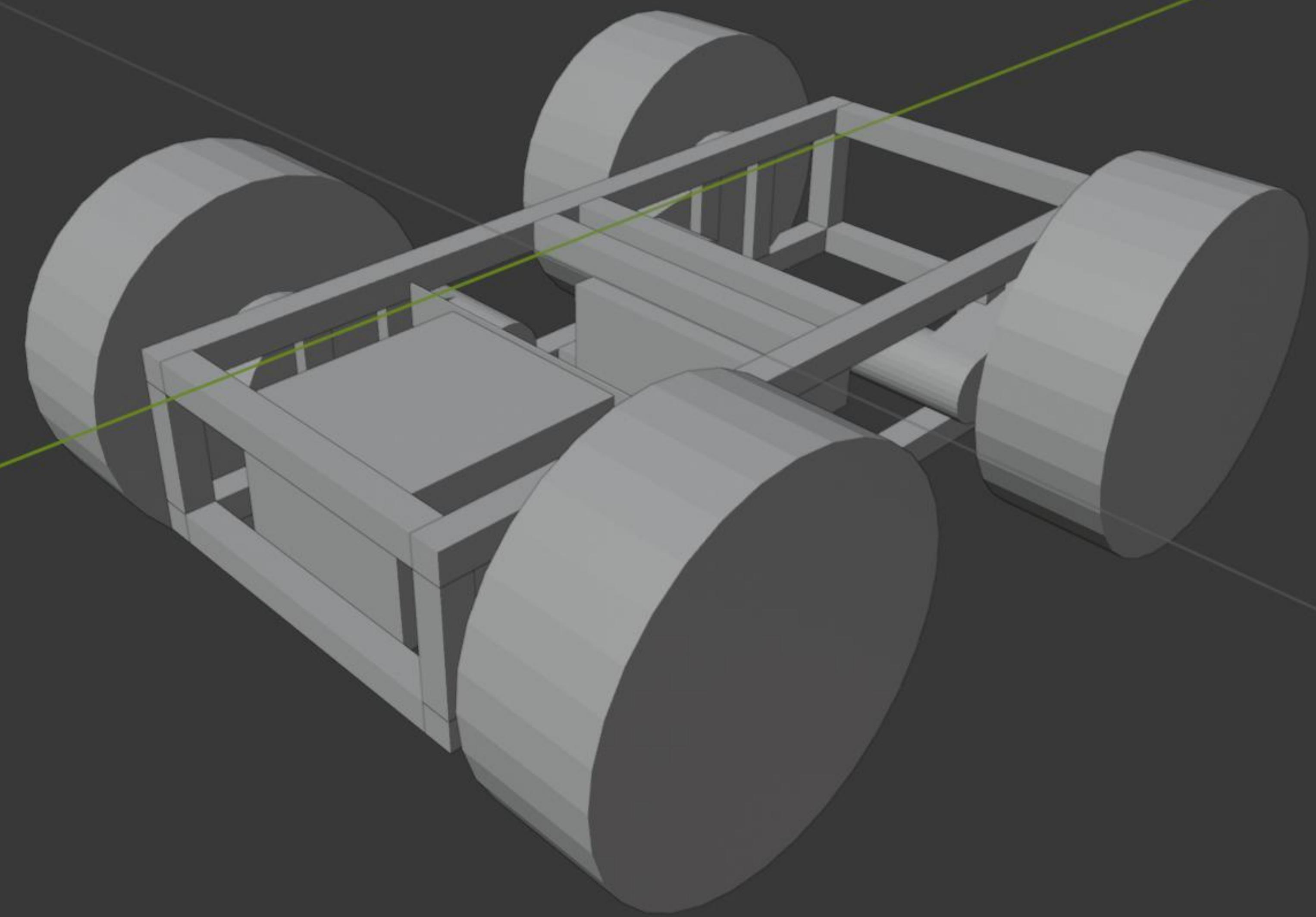
- $90 \text{ RPM} * 20 \text{ cm} * 3.14 = 5,652 \text{ cm / min} = 56.52 \text{ m / min} == 3.4 \text{ km / h}$
- $270 \text{ RPM} \approx 10 \text{ km / h}$

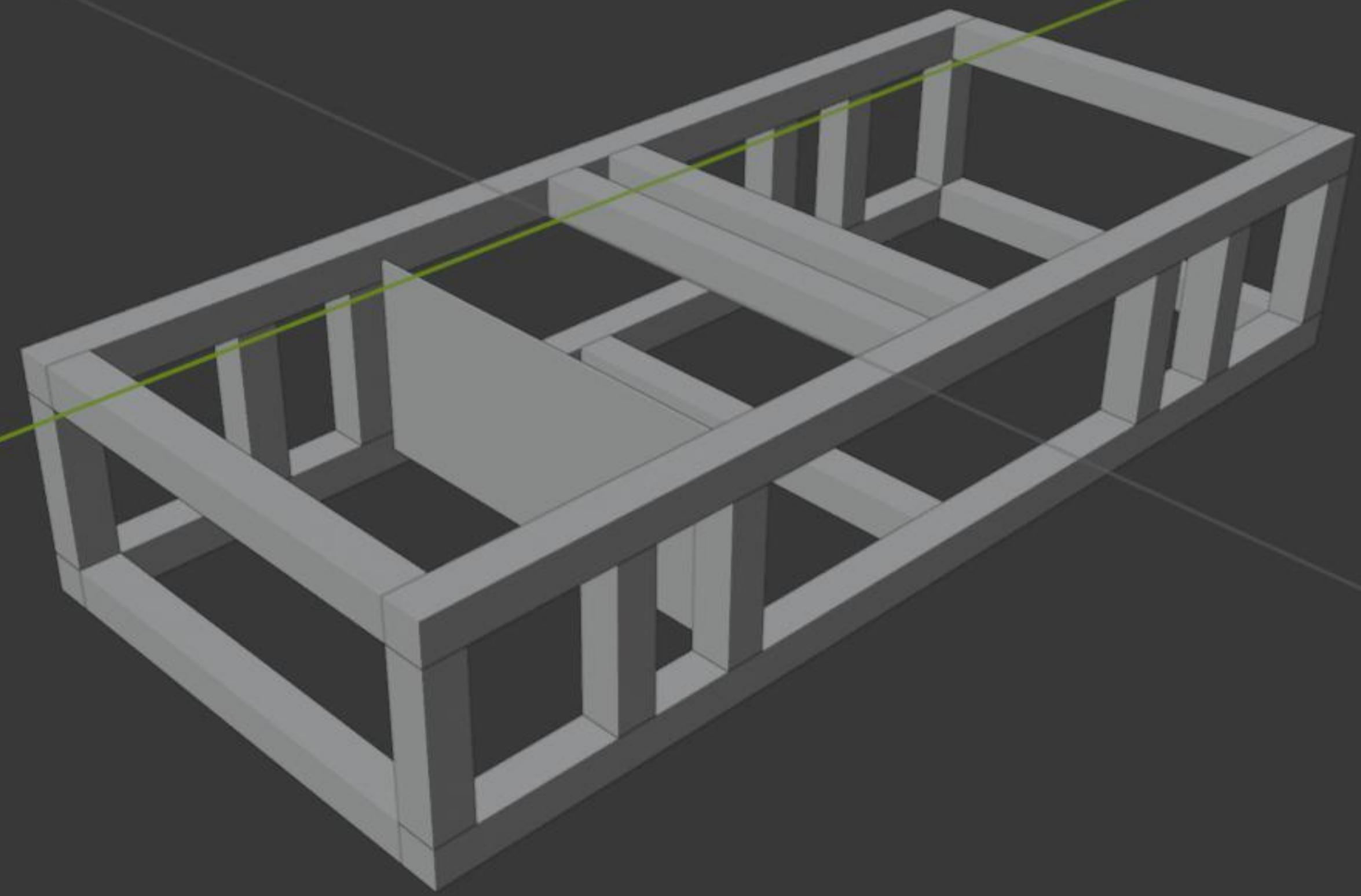
arzhang: design: mechanical

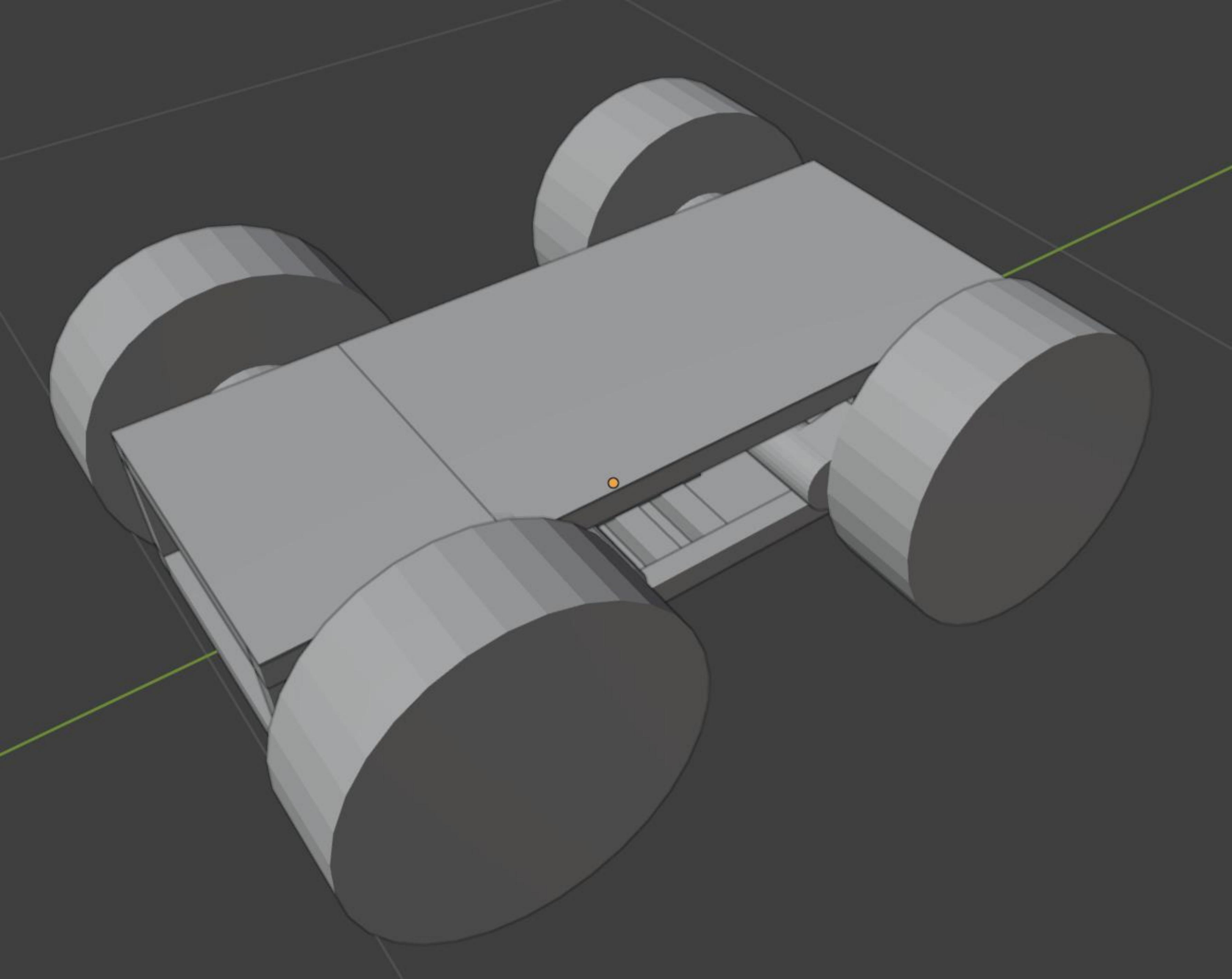
- [Blender files](#)
- metal profile: 20 mm x 20 mm x 1.5 mm x 6000 mm
 - parts:
 - 4 x 600 mm
 - 7 x 220 mm
 - 12 x 80 mm
 - total: 5060 mm (5.06 m)
 - cut into 4 x (600 mm + 2 x 220 mm + 3 x 80 mm = 1280 mm) ≈ 1500 mm
- plexiglass
 - 260 mm x 170 mm x 2 mm
 - 260 mm x 430 mm x 2 mm
- thin metal sheet
 - 260 mm x 600 mm x 0.5 mm
 - 2 x 120 mm x 600 mm x 0.5

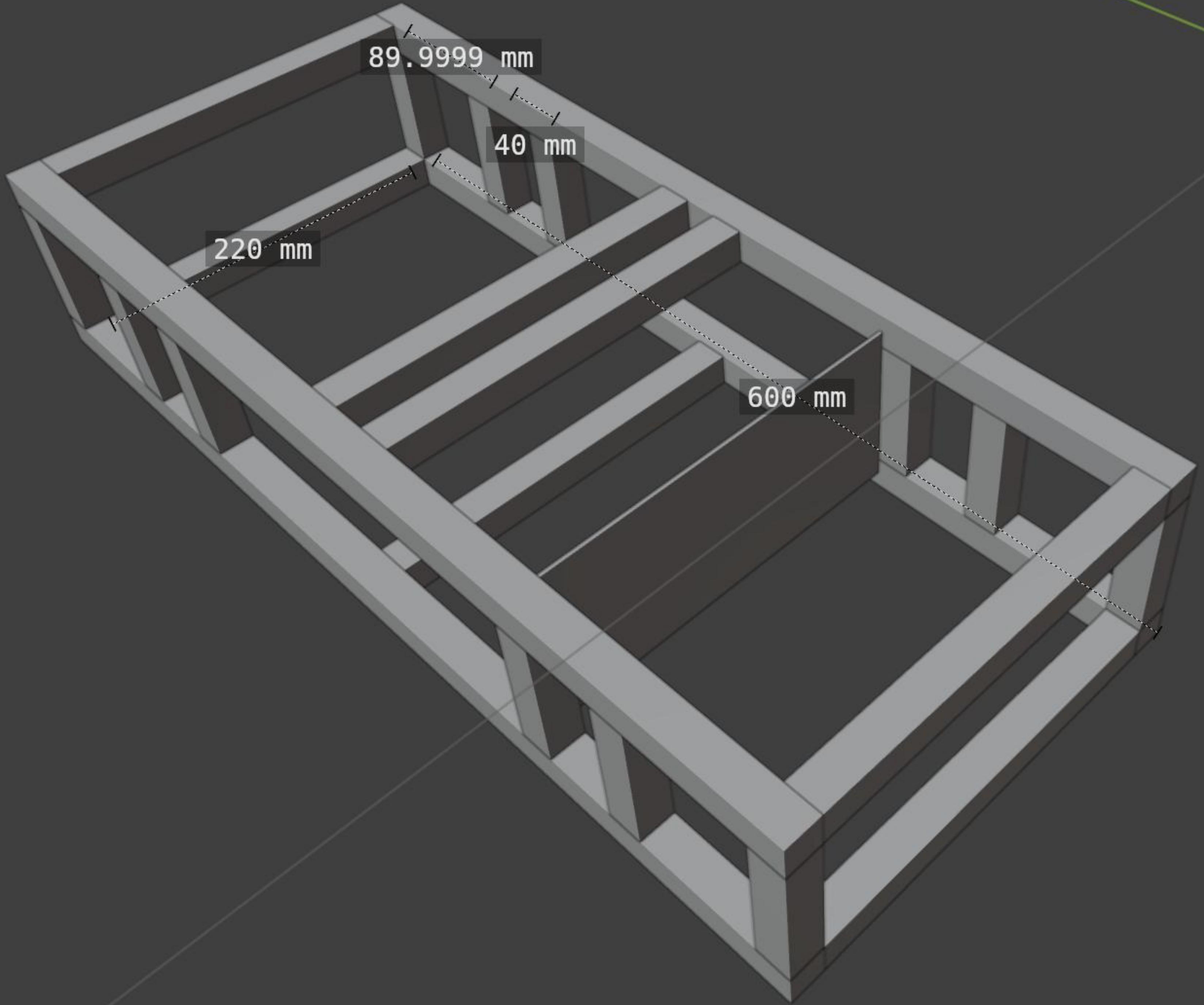


- [v1](#)









arzhang: design: parts

[swallow design](#) parts + 2 optional [swallow-head computers](#) + [!\[\]\(642cff3cbbe1a19b5b6c1472ce9ec6fb_img.jpg\)](#).

1. [Gearboxed DC Motor, 12 V \(3-24 V\), 3A, 120 RPM, 1:91, 15 Kg.cm](#):
proposed, not used..

[Gearboxed DC Motor, 12 V \(3-24 V\), 3A, 120 RPM, 1:91, 15 Kg.cm](#)



proposed, not used.

validations

- [timing-review](#)
- [village-1](#)
- [village-2](#)
- [village-3](#)
- [village-4](#)
- [village-5](#)
- [village-6](#)
- [village-7](#)



[timing-review](#)

yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image size: 256 | took 1 s, 636 ms | 0 detection(s):


[village-5](#)

yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image size: 256 | took 4 s, 150 ms | 0 detection(s):


[village-6](#)

yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image size: 256 | took 1 s, 431 ms | 0 detection(s):


[village-7](#)

yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image size: 256 | took 1 s, 431 ms | 0 detection(s):



validations: timing-review

UGV(s):  [arzhang2](#)

single thread

```
:: in 1 min called 8 function(s):
:: # 0 - ClassicalUltrasonicSensor: called 388 time(s), total 46 s, avg
120 ms
:: # 1 - ClassicalYoloCamera: called 388 time(s), total 29 s, avg 076 ms
:: # 2 - ClassicalKeyboard: called 388 time(s), total 489 ms, avg 001 ms
:: # 3 - ClassicalLeds: called 388 time(s), total 081 ms, avg < 1 ms
:: # 4 - ClassicalRightMotor: called 388 time(s), total 072 ms, avg < 1
ms
:: # 5 - ClassicalLeftMotor: called 388 time(s), total 060 ms, avg < 1
ms
:: # 6 - ClassicalPushButton: called 388 time(s), total 025 ms, avg < 1
ms
:: # 7 - ClassicalSetPoint: called 388 time(s), total 005 ms, avg < 1 ms
```

loop frequency (Hz): 7

► yaml

multi-threaded

yolo and ultrasonic run on individual threads.

```
:: in 5 min called 8 function(s):
:: # 0 - session.update: called 5,881 time(s), total 10 s, avg 002 ms
:: # 1 - ClassicalKeyboard: called 5,881 time(s), total 5 s, avg 001 ms
:: # 2 - ClassicalRightMotor: called 5,881 time(s), total 1 s, avg < 1
ms
:: # 3 - ClassicalLeds: called 5,881 time(s), total 1 s, avg < 1 ms
:: # 4 - ClassicalLeftMotor: called 5,881 time(s), total 888 ms, avg < 1
ms
:: # 5 - ClassicalPushButton: called 5,881 time(s), total 417 ms, avg <
1 ms
:: # 6 - ClassicalSetPoint: called 5,881 time(s), total 061 ms, avg < 1
ms
:: # 7 - ClassicalYoloCamera: called 5,881 time(s), total 033 ms, avg <
1 ms
```

loop frequency (Hz): 570.76

► yaml

```
yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image  
size: 256 | took 1 s, 636 ms | 0 detection(s):
```

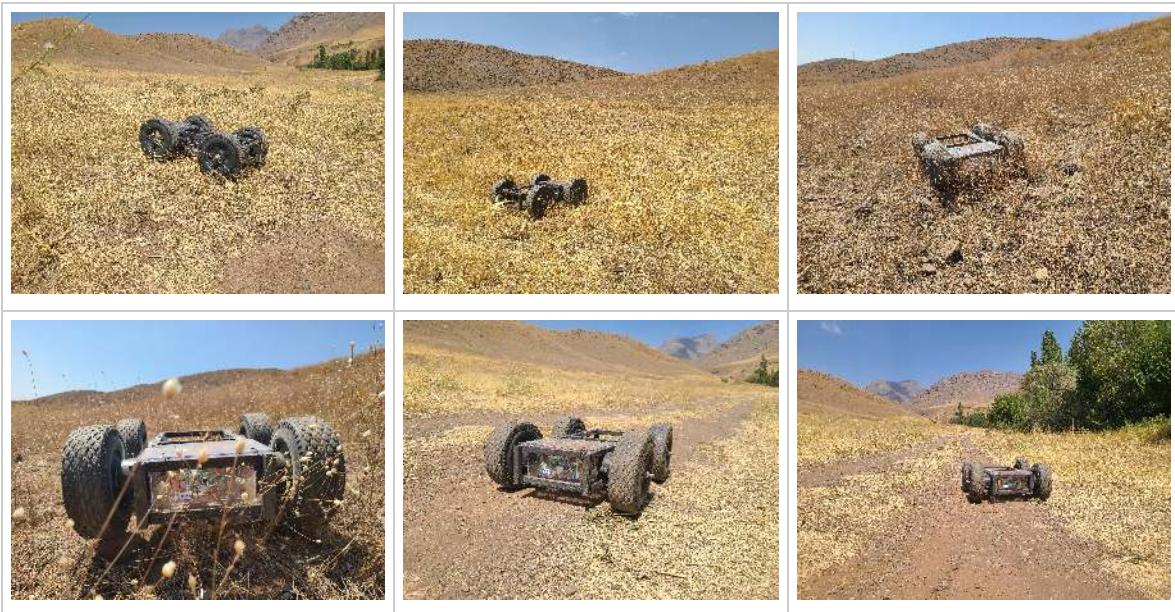


```
bluer_algo-4.428.1 | bluer_ai-12.288.1-main | bluer_objects-6.301.1 |  
bluer_options-5.181.1 | blueness-3.118.1 | torch-2.8.0+cpu | Python 3.11.2 |  
Linux 6.12.25+rpt-rpi-v8 | sparrow2 | 00000000b6eb173d | Sion
```

image

validations: village-1

UGV(s):  [arzhang](#)



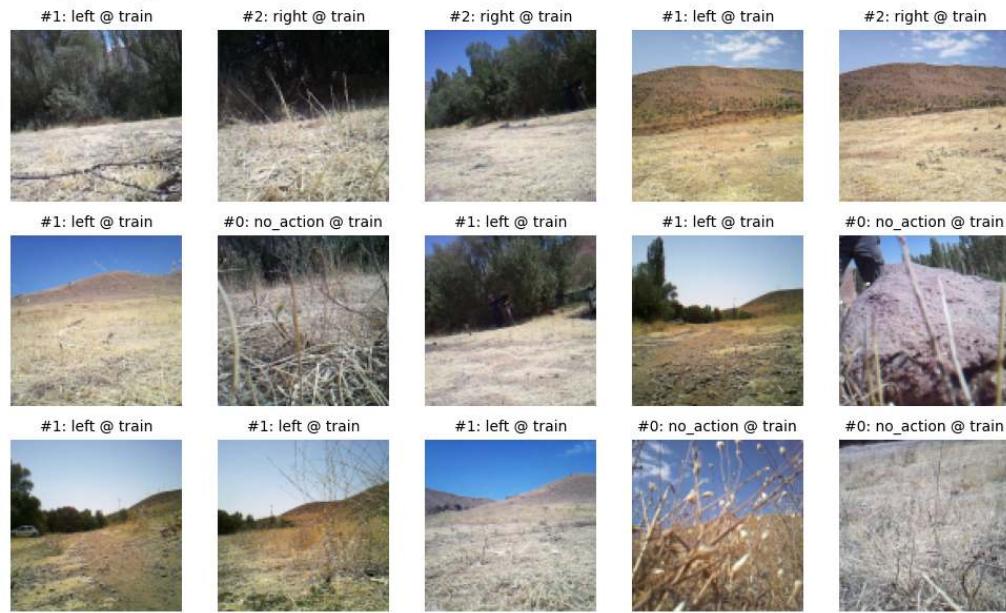
@select 2025-09-05-11-48-27-d56azo

@download policy=doesnt_exist

@upload public,zip

[2025-09-05-11-48-27-d56azo](#)

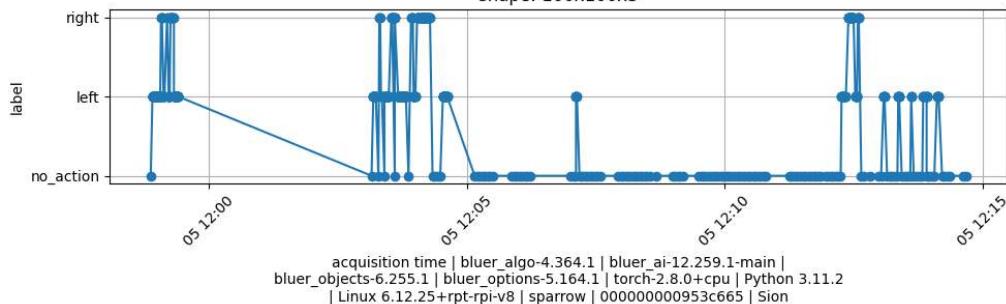
2025-09-05-11-48-27-d56azo/grid.png | 05 September 2025 | 12:19:53 (+0330) | 100x100x3:uint8 | count: 560 | 3 subset(s): train: 560 [%100.0], test: 0 [%0.0], eval: 0 [%0.0] | 3 class(es): no_action: 131 [%23.4], left: 298 [%53.2], right: 131 [%23.4]



bluer_algo-4.364.1 | bluer_ai-12.259.1-main | bluer_objects-6.255.1 | bluer_options-5.164.1 | torch-2.8.0+cpu | Python 3.11.2 | Linux 6.12.25+rpt-rpi-v8 | sparrow | 000000000953c665 | Sion

image

2025-09-05-11-48-27-d56azo | 05 September 2025 | 12:19:53 (+0330) | 560 record(s) | 3 subset(s): train: 560 [%100.0], test: 0 [%0.0], eval: 0 [%0.0] | 3 class(es): no_action: 131 [%23.4], left: 298 [%53.2], right: 131 [%23.4] | shape: 100x100x3



image

```
dataset:
class_count: 3
classes:
  0: no_action
  1: left
  2: right
count: 560
shape:
- 100
- 100
- 3
source: 000000000953c665
subsets:
  eval: 0
  test: 0
  train: 560
```



image

observations

- image-classifier is validated.

validations: village-2

UGV(s):  [arzhang](#)



debug objects

► collection

swallow-debug-2025-09-22-09-47-32-85hag3 	swallow-debug-2025-09-22-09-59-29-emj29v 	swallow-debug-2025-09-22-10-01-01-uzray6
swallow-debug-2025-09-22-10-06-19-hcyl1v	swallow-debug-2025-09-22-10-09-44-z6q9kn	swallow-debug-2025-09-22-10-19-35-mobajm



observations

- two wheel nuts loosened every few minutes. ⚡
- one wheel nut tightened every few minutes. ⚡

-> fixed in [village-3](#)



image

validations: village-3

UGV(s):  [arzhang](#)



debug object

- ▶ collection

[swallow-debug-2025-09-25-13-16-59-rnm7jd](#)



image

session object

```
dataset:  
  count: 0  
names:  
  0: target  
source: 000000000953c665  
train: images/train  
val: images/val
```

observations

- no wheel nuts loosened or tightened (: [village-2](#)) ✓
- bottom cover broke. ->

validations: village-4

UGV(s):  [arzhang2](#)

debug object

```
runme() {
    @select $1
    @upload public,zip
    @assets publish extensions=gif,push
}
runme swallow-debug-2025-09-26-17-44-51-6pb87y
```

observations

- wheels functioned as expected. ✓
- robot rebooted after a minute of operation and again. ⚠ loose power connections on shield found and fixed, subsequent testing validated the fix. ✓
- camera is upside down - fixed. ✓

runme swallow-debug-2025-09-27-19-15-31-6iq5vz

swallow-debug-2025-09-26-17-44-51-6pb87y	swallow-debug-2025-09-27-19-15-31-6iq5vz
	



validations: village-5

UGV(s):  [arzhang](#),  [arzhang2](#)

script

```
runme() {
    @select $1
    @upload public,zip
    @assets publish extensions=gif,push
    # ---
    @select $2
    @swallow ultrasonic review \
        upload .
    @assets publish extensions=gif+png,push
}
```

objects

arzhang

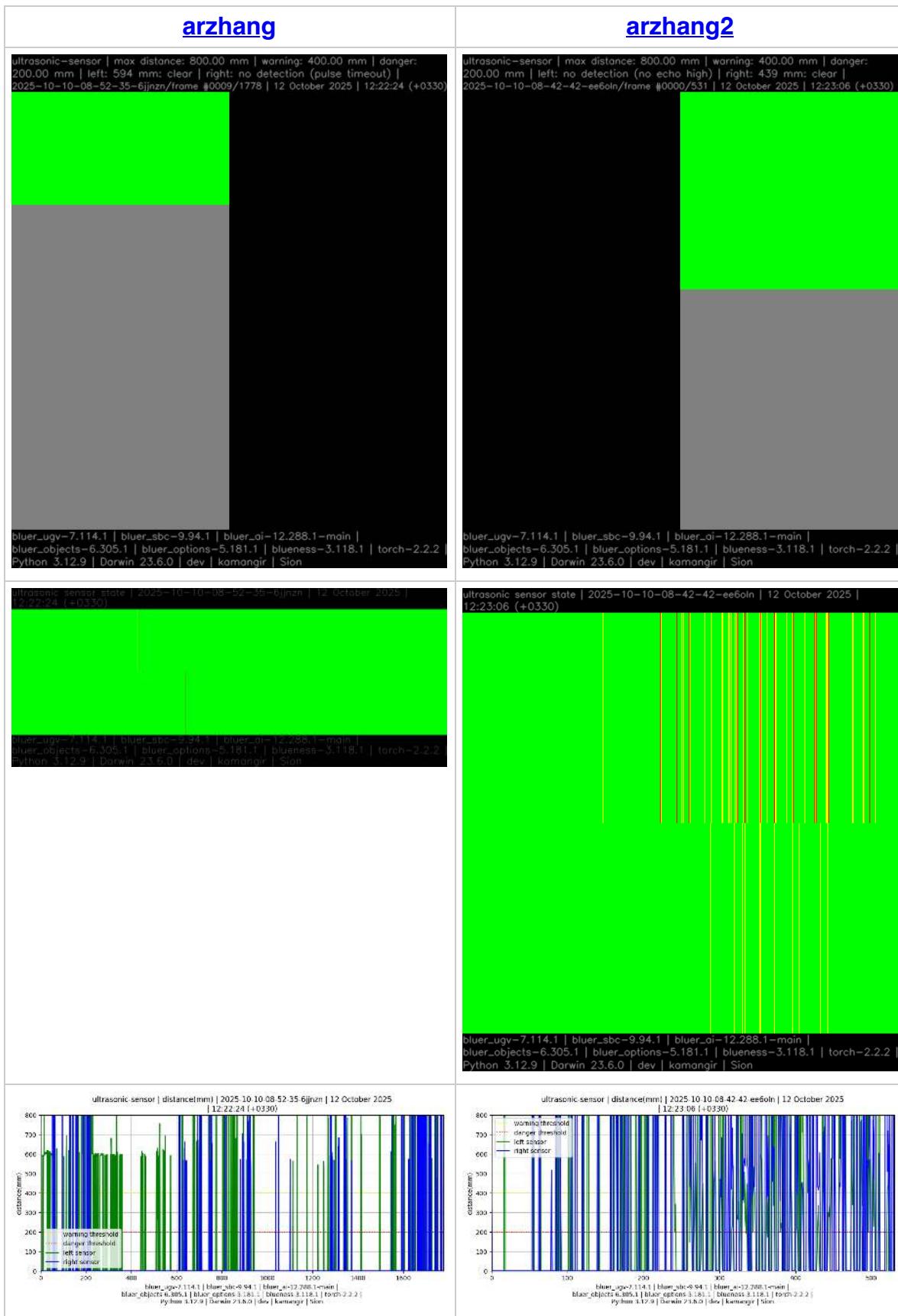
```
runme \
    swallow-debug-2025-10-10-08-49-45-yk18ei \
    2025-10-10-08-52-35-6jjnzn
```

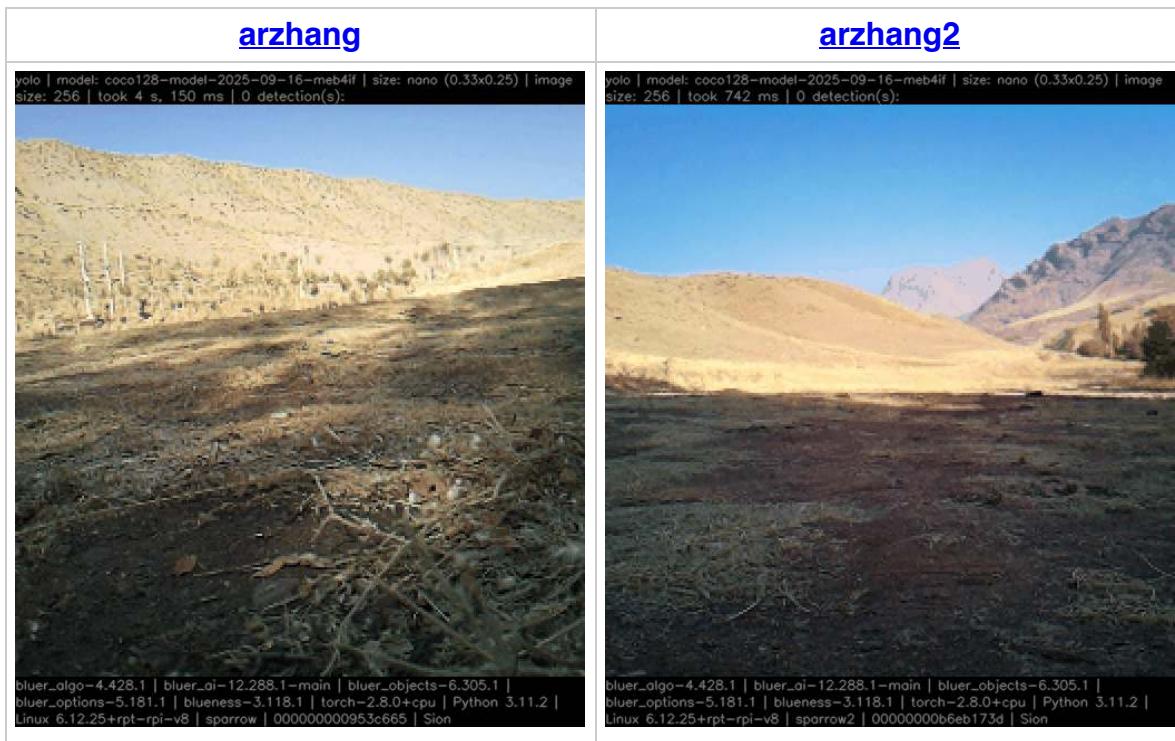
loop frequency (Hz): 310.87

arzhang2

```
runme \
    swallow-debug-2025-10-10-08-40-38-k8oc2p \
    2025-10-10-08-42-42-ee6oln
```

loop frequency (Hz): 408.95





observations

- ultrasonic sensor is activated when the surface is uneven - will adjust the sensor.
- yolo may not have time to perform the action. -> 



validations: village-6

UGV(s):  [arzhang](#),  [arzhang2](#)

scripts

```
@select swallow-debug-2025-10-19-14-14-23-ectn97
@upload public,zip
@assets publish extensions=gif,push
runme() {
    @select $1
    @swallow ultrasonic review \
        upload .
    @assets publish extensions=gif+png,push
}
runme 2025-10-19-14-16-36-tunrlm
runme 2025-10-19-14-16-07-75yxbw
```

objects

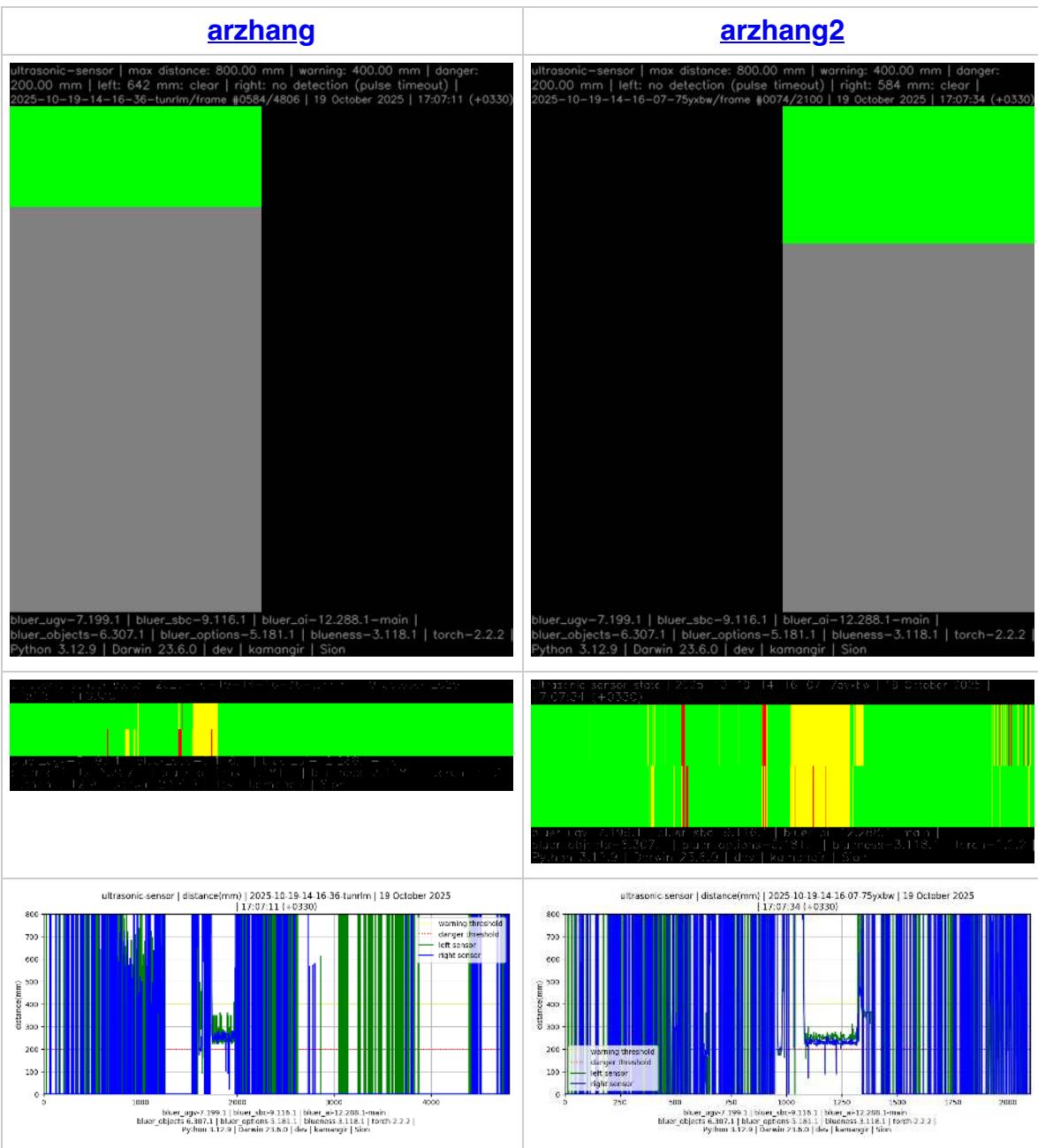
arzhang

loop frequency (Hz): 321.78

arzhang2

loop frequency (Hz): 242.58

arzhang	arzhang2
2025-10-19-14-16-36-tunrlm	2025-10-19-14-16-07-75yxbw



```
yolo | model: coco128-model-2025-09-16-meb4if | size: nano (0.33x0.25) | image  
size: 256 | took 1 s, 431 ms | 0 detection(s):
```



```
bluer_algo-4.428.1 | bluer_ai-12.288.1-main | bluer_objects-6.307.1 |  
bluer_options-5.181.1 | blueness-3.118.1 | torch-2.8.0+cpu | Python 3.11.2 |  
Linux 6.12.25+rpi-rpi-v8 | sparrow2 | 00000000b6eb173d | Sion
```

image

observations

- the range of numpad is ~10-20 m, noticeably lower than that of the full keyboard, which is ~50 m.





validations: village-7

UGV(s):  [arzhang](#),  [arzhang2](#),  [arzhang3](#)

- publication

arzhang

loop frequency (Hz): 185.88

[2025-11-06-10-50-35-myadvn](#)

arzhang2

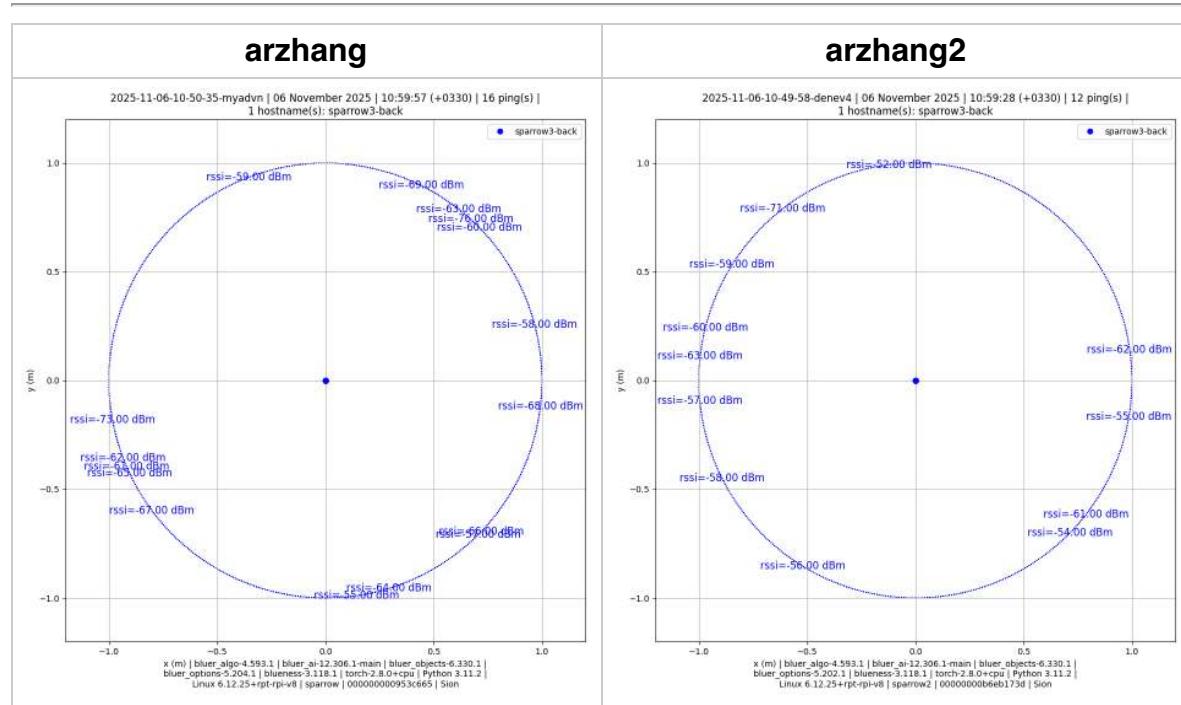
loop frequency (Hz): 190.13

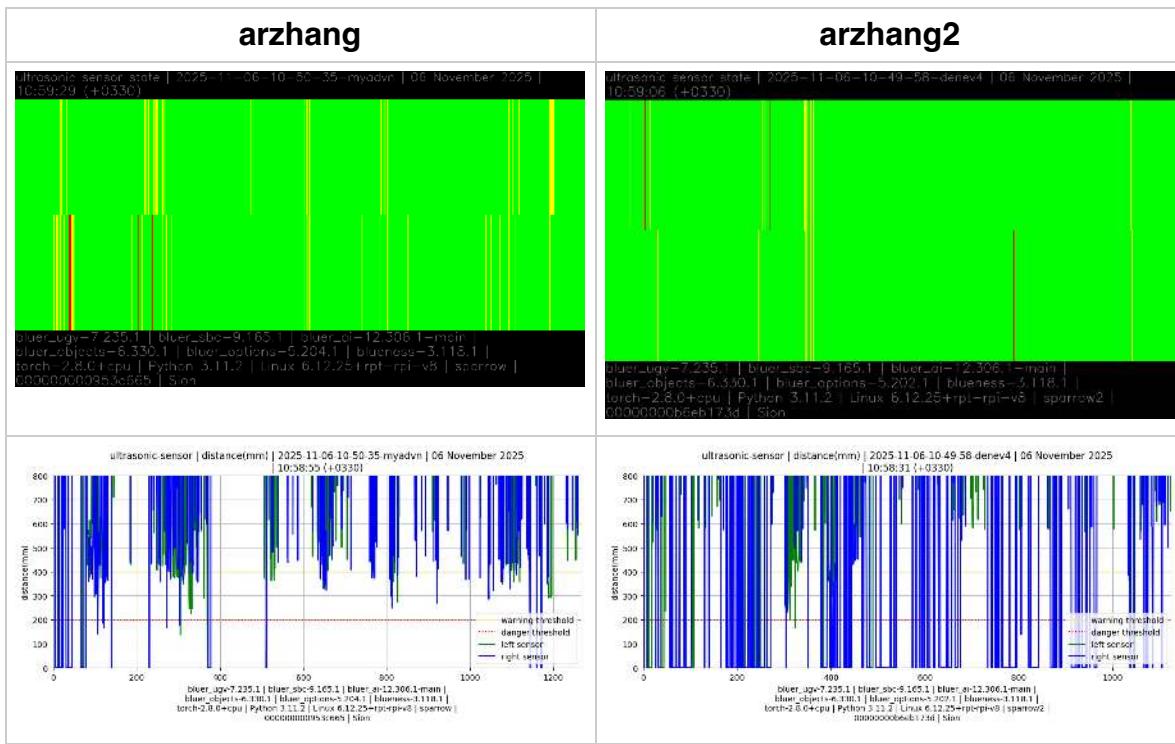
[2025-11-06-10-49-58-denev4](#)

arzhang3

loop frequency (Hz): 183.64

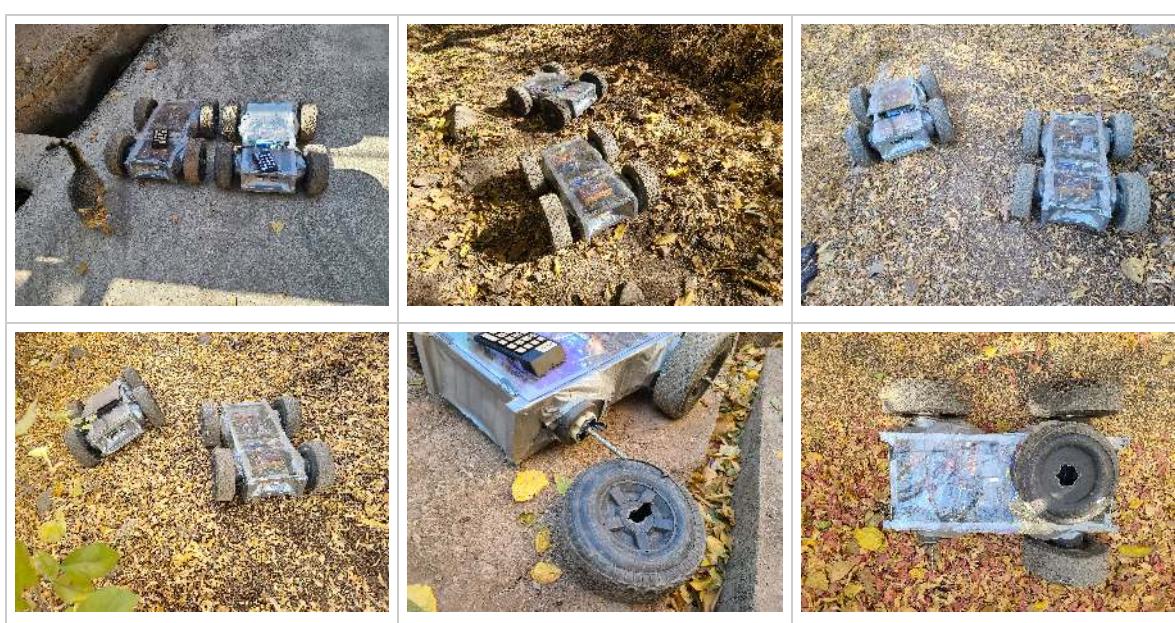
[2025-11-06-10-49-36-koxzf3](#)



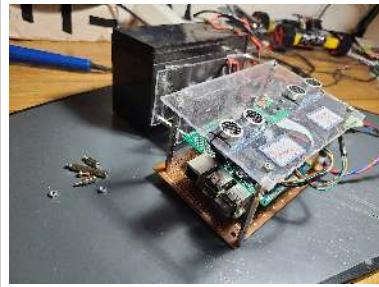


observations

- one of the back wheels on arzhang broke. ->
- the on/off switch on arzhang broke ->
- the two arzhangs did not receive each other's advertisements. they both received advertisements from arzhang3. - will review in the next validation.

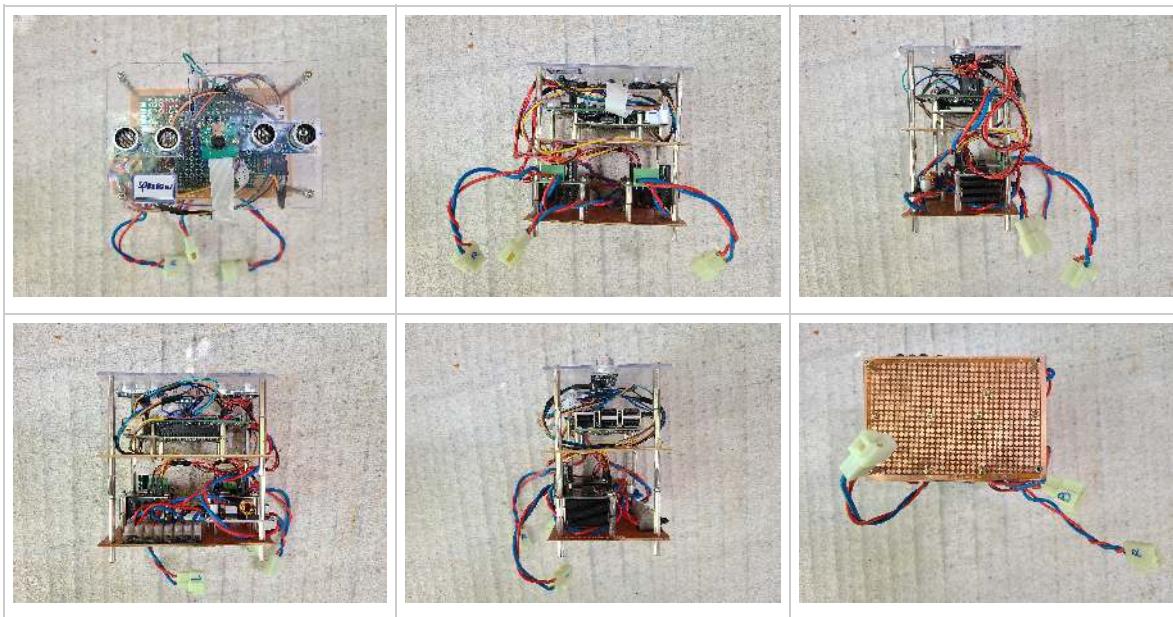


11/9/25, 3:22 AM



swallow

- minimum hardware for [UGV control](#).
- Raspberry Pi, 5 MP camera, 2+2 ultrasonic sensors, 2 x 12 VDC 43A DC motor drivers.
- 12 VDC, A.
- width x height x length: 137 mm x 93 mm x 150 mm
- [terraform](#).
- [dev notes](#).



parts

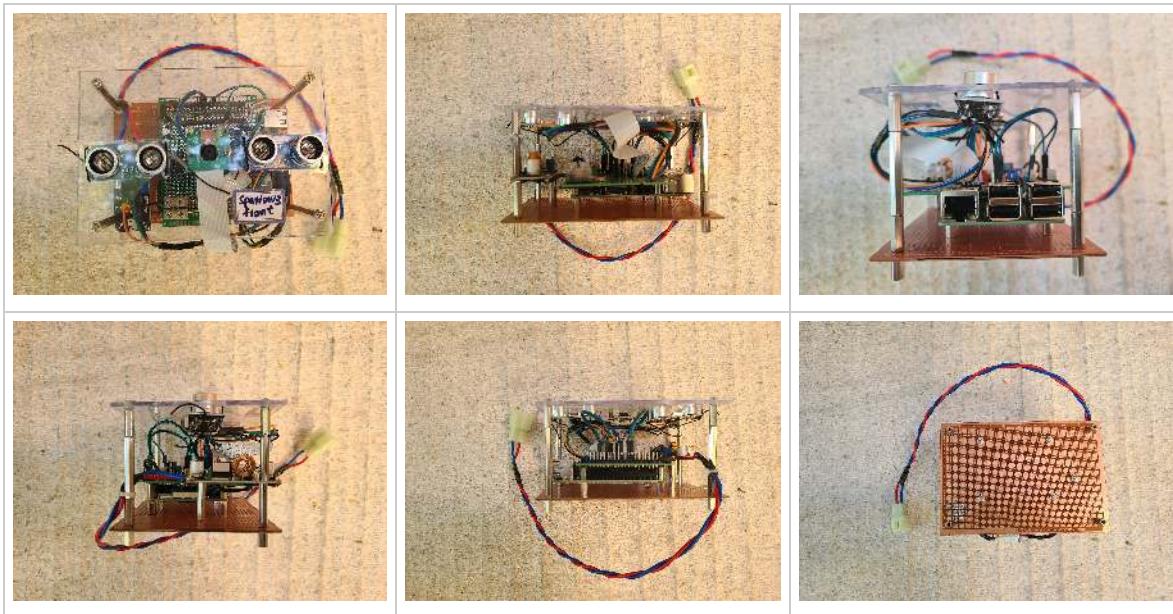
[swallow-head](#) +

43 A, H-Bridge Motor Driver	auto power connectors	nuts, bolts, and spacers	solid cable 1-1.5 mm²	white terminal
 2 x	 2 females	 M3: (8 x 25 mm spacer + 4 x 30 mm spacer + 4 x nut)	 20 cm x (red + black/blue)	 8 x

1. [43 A, H-Bridge Motor Driver](#): 2 x.
2. [auto power connectors](#): 2 females.
3. [nuts, bolts, and spacers](#): M3: (8 x 25 mm spacer + 4 x 30 mm spacer + 4 x nut).
4. [solid cable 1-1.5 mm²](#): 20 cm x (red + black/blue).
5. [white terminal](#): 8 x.

swallow-head

- head of [swallow computer](#), without the DC motor drivers.
- width x height x length: 137 mm x 93 mm x 90 mm
- used for [swallow design](#) backs and [ranging tops](#).
- [dev notes](#).



parts

<u>16 AWG wire</u>  16 AWG 40 cm x (red + black/ blue)	<u>HC-SR04: ultras onic- sensor</u>  4 x	<u>LED, ~2 V</u>  green + red + yellow + 4 x blue	<u>Polyfu se,</u> <u>forwar d</u> <u>voltag e, 10- 20 mA</u>	<u>1.1 A</u> <u>hold,</u> <u>2.2 A</u> <u>trip,</u> <u>16 V,</u> <u>resett able,</u> <u>throug h- hole,</u> <u>e.g.,</u> <u>MF- R110</u>	<u>Raspbe rry Pi</u> <u>Camera</u> <u>1.1 ht tps:// 16 V,</u> <u>V1.3ht trip,</u> <u>16 V,</u> <u>resett able,</u> <u>throug h- hole,</u> <u>e.g.,</u> <u>MF- R110</u>	<u>Raspbe rry Pi.</u> 	<u>Resist or,</u> <u>1/4 watt,</u> <u>5% tolera nce</u>	<u>SD card,</u> <u>32 GB</u> 	<u>TVS diode,</u> <u>unidir ection</u> <u>600 W,</u> <u>6.8 V clamp,</u> <u>e.g.,</u> <u>P6KE6.</u> <u>8A,</u> <u>D0-15 packag e</u>	<u>XL4015 :</u> <u>12 VDC -> 5 VDC,</u> 
--	---	---	--	--	--	--	---	--	---	--

<u>auto power connectors</u>	<u>capacitor, 470 µF</u>	<u>double-sided PCB, 9 cm x 7 cm</u>	<u>dupont cables</u>	<u>nuts, bolts, and spacer</u>	<u>pin headers</u>	<u>plexiglass, 2 mm or 2.5 mm thicknes</u>	<u>push button</u>	<u>single-sided PCB, 14 cm x 9.5 cm</u>	<u>solid cable 1-1.5 mm^2</u>
									
1 female	1000 µF, 16 V or 25 V, Electr olytic, 105 °C rated if possib le.		2 x	1 x 30 cm + 1 x 10 cm	M2: (2 x bolt + 2 x 5 mm spacer + 4 x nut) + M2.5: (4 x bolt + 8 x 10 mm spacer + 4 x nut) + M3: (1 x bolt + 3 x 35 mm spacer + 3 x 25 mm spacer + 7 x 15 mm spacer + 4 x 5 mm spacer + 5 x nut)	1 x (female, 2 x 40) -> 2 x 20 + 2 x (male, 1 x 40) -> 4 x 6 + 4 x 2 + 2 x 20	14 cm x 9.5 cm	2 x	10 cm x (red + black/blue)
<u>strong thread</u>	<u>white terminal</u>								
1 m			2 x						

1. 16 AWG wire: 40 cm x (red + black/blue).
2. HC-SR04: ultrasonic-sensor: 4 x.
3. LED, ~2 V forward voltage, 10-20 mA: green + red + yellow + 4 x blue.
4. Polyfuse, 1.1 A hold, 2.2 A trip, 16 V, resettable, through-hole, e.g., MF-R110: optional.

5. [Raspberry Pi Camera, V1.3](https://www.raspberrypi.com/documentation/accessories/camera.html)
6. [Raspberry Pi..](#)
7. [Resistor, 1/4 watt, 5% tolerance](#): 7 x 330-470 Ω + N x 2.2 kΩ + N x 3.3 kΩ.
8. [SD card, 32 GB.](#)
9. [TVS diode, unidirectional, 600 W, 6.8 V clamp, e.g. P6KE6.8A, DO-15 package.](#)
10. [XL4015: 12 VDC -> 5 VDC, 4A.](#)
11. [auto power connectors](#): 1 female.
12. [capacitor, 470 μF to 1000 μF, 16 V or 25 V, Electrolytic, 105 °C rated if possible..](#)
13. [double-sided PCB, 9 cm x 7 cm](#): 2 x.
14. [dupont cables, female to female](#): 1 x 30 cm + 1 x 10 cm.
15. [nuts, bolts, and spacers](#): M2: (2 x bolt + 2 x 5 mm spacer + 4 x nut) + M2.5: (4 x bolt + 8 x 10 mm spacer + 4 x nut) + M3: (1 x bolt + 3 x 35 mm spacer + 3 x 25 mm spacer + 7 x 15 mm spacer + 4 x 5 mm spacer + 5 x nut).
16. [pin headers](#): 1 x (female, 2 x 40) -> 2 x 20 + 2 x (male, 1 x 40) -> 4 x 6 + 4 x 2 + 2 x 20.
17. [plexiglass, 2 mm or 2.5 mm thickness](#): 14 cm x 9.5 cm.
18. [push button](#).
19. [single-sided PCB, 14 cm x 9.5 cm](#): 2 x.
20. [solid cable 1-1.5 mm²](#): 10 cm x (red + black/blue).
21. [strong thread](#): 1 m.
22. [white terminal](#): 2 x.

image-classifier

navigation for a [ugv](#).

- [dataset](#)
- [model](#)

image-classifier: dataset

- [ingest](#)
- [review](#)
- [sequence](#)

image-classifier: dataset: ingest

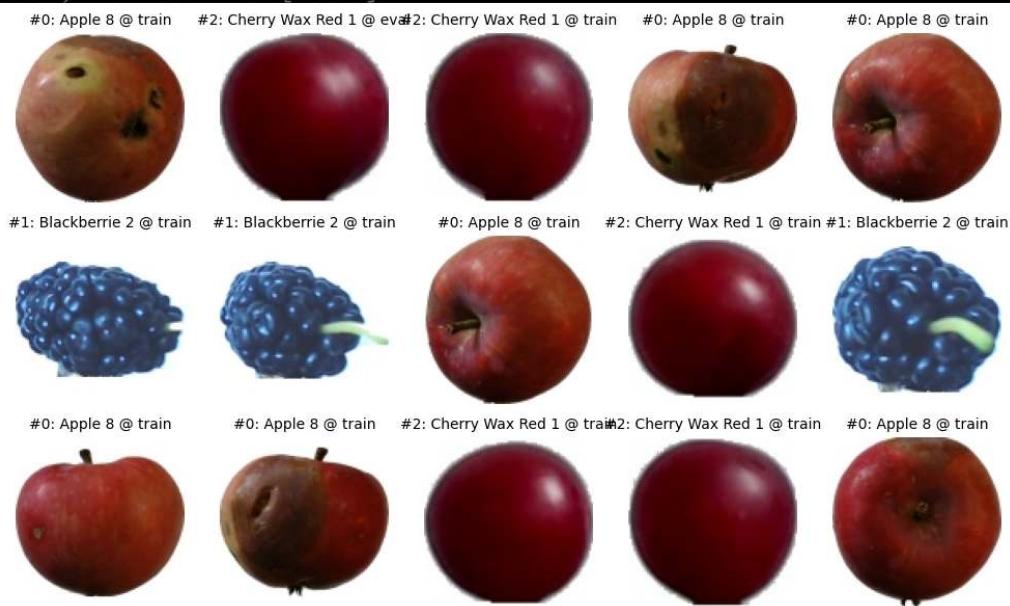
- continues <https://github.com/kamangir/image-classifier-2>.
- uses <https://github.com/fruits-360/fruits-360-100x100>

```
@select fruits-365-dataset-$(@timestramp)
```

```
@algo image_classifier dataset ingest \
clone, count=100, source=fruits_360, upload . \
--class_count 3
```

```
@upload public, zip .
@assets publish \
extensions=png, push .
```

```
fruits-365-dataset-2025-07-01-gn9up7/grid.png | 12 July 2025 | 19:57:53 (+0330)
| 100x100x3:uint8 | count: 99 | 3 subset(s): train: 83 [%83.8], test: 5 [%5.1],
eval: 11 [%11.1] | 3 class(es): Apple 8: 33 [%33.3], Blackberry 2: 33 [%33.3],
Cherry Wax Red 1: 33 [%33.3]
```



```
bluer_algo-4.211.1 | bluer_ai-12.239.1-wifi-fix-2025-07-09-oxrie | \
bluer_objects-6.231.1 | bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 | \
Darwin 23.6.0 | dev.local | kamangir | Zagros_5G
```

image

[fruits-365-dataset-2025-07-01-gn9up7](#)

```
dataset:
  class_count: 3
  classes:
    0: Apple 8
    1: Blackberry 2
    2: Cherry Wax Red 1
  count: 99
  ratios:
    eval: 0.09999999999999998
    test: 0.1
    train: 0.8
```

```
shape:  
- 100  
- 100  
- 3  
source: fruits_360  
subsets:  
  eval: 11  
  test: 5  
  train: 83
```

image-classifier: dataset: review

uses [ingest](#).

```
@select $BLUER_ALGO_FRUITS_360_TEST_DATASET
@algo image_classifier dataset review - .
@upload public,zip .
@assets publish \
extensions=png,push .
```

```
fruits-365-dataset-2025-07-01-gn9up7/grid.png | 12 July 2025 | 19:57:53 (+0330)
| 100x100x3:uint8 | count: 99 | 3 subset(s): train: 83 [%83.8], test: 5 [%5.1],
eval: 11 [%11.1] | 3 class(es): Apple 8: 33 [%33.3], Blackberry 2: 33 [%33.3],
Cherry Wax Red 1: 33 [%33.3]
```



```
bluer_algo-4.211.1 | bluer_ai-12.239.1-wifi-fix-2025-07-09-oxrieh |
bluer_objects-6.231.1 | bluer_options-5.155.1 | torch-2.2.2 | Python 3.12.9 |
Darwin 23.6.0 | dev.local | kamangir | Zagros_5G
```

image

[fruits-365-dataset-2025-07-01-gn9up7](#)

```
dataset:
  class_count: 3
  classes:
    0: Apple 8
    1: Blackberry 2
    2: Cherry Wax Red 1
  count: 99
  ratios:
    eval: 0.09999999999999998
    test: 0.1
    train: 0.8
  shape:
    - 100
    - 100
```

```
- 3
source: fruits_360
subsets:
  eval: 11
  test: 5
  train: 83
```

image-classifier: dataset: sequence

uses bluer-ugv/swallow/digital/dataset/combination.

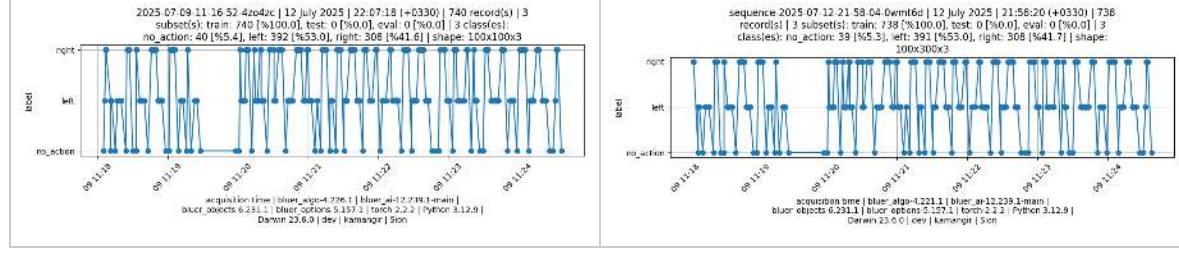
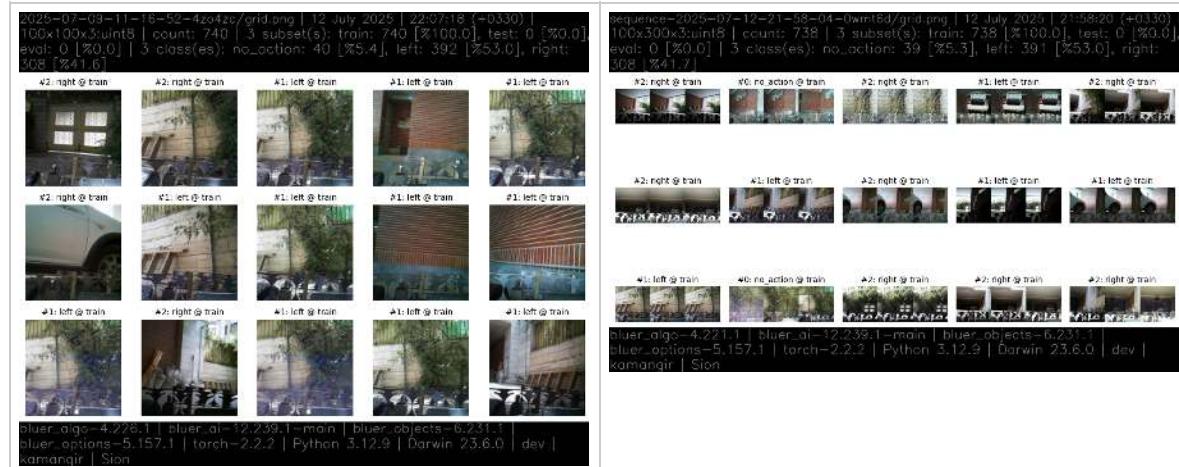
```
@select 2025-07-09-11-16-52-4zo4zc
```

```
@upload public,zip .
@assets publish \
    extensions=png,push . \
--prefix grid
```

```
@select sequence-$(@timestamp)
```

```
@algo image_classifier dataset sequence \
~download,length=3 ... .
```

```
@upload public,zip .
@assets publish \
    extensions=png,push . \
--prefix grid
```



[2025-07-09-11-16-52-4zo4zc](https://bluer-ugv/swallow/digital/dataset/combination)

```
dataset:
class_count: 3
classes:
  0: no_action
  1: left
  2: right
count: 740
shape:
```

```
- 100
- 100
- 3
source: 00000000c74cf7d2
subsets:
  eval: 0
  test: 0
  train: 740
```

[sequence-2025-07-12-21-58-04-0wmt6d](#)

```
dataset:
  class_count: 3
  classes:
    0: no_action
    1: left
    2: right
  count: 738
  length: 3
  shape:
- 100
- 300
- 3
source: 2025-07-09-11-16-52-4zo4zc
subsets:
  eval: 0
  test: 0
  train: 738
```

image_classifier: model

- [train](#)
- [prediction](#)

image-classifier: model: train

- [small dataset](#)
- [large dataset](#)

image-classifier: model: train: small

uses [ingest](#).

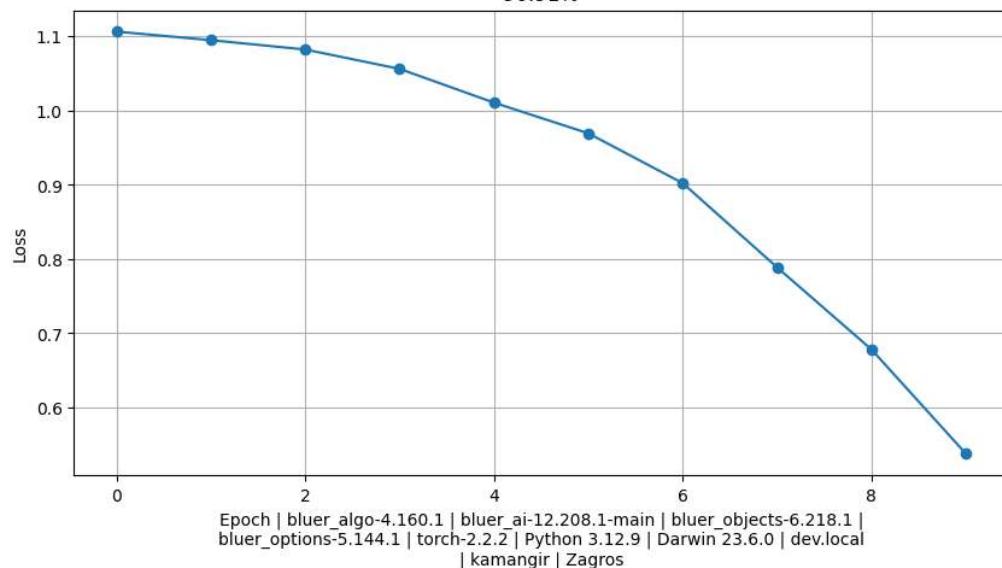
```
@select $BLUER_ALGO_FRUITS_360_TEST_DATASET
@select fruits-365-model-$(@timestamp)
```

```
@algo image_classifier model train \
    upload ...
```

```
@upload public,zip .
```

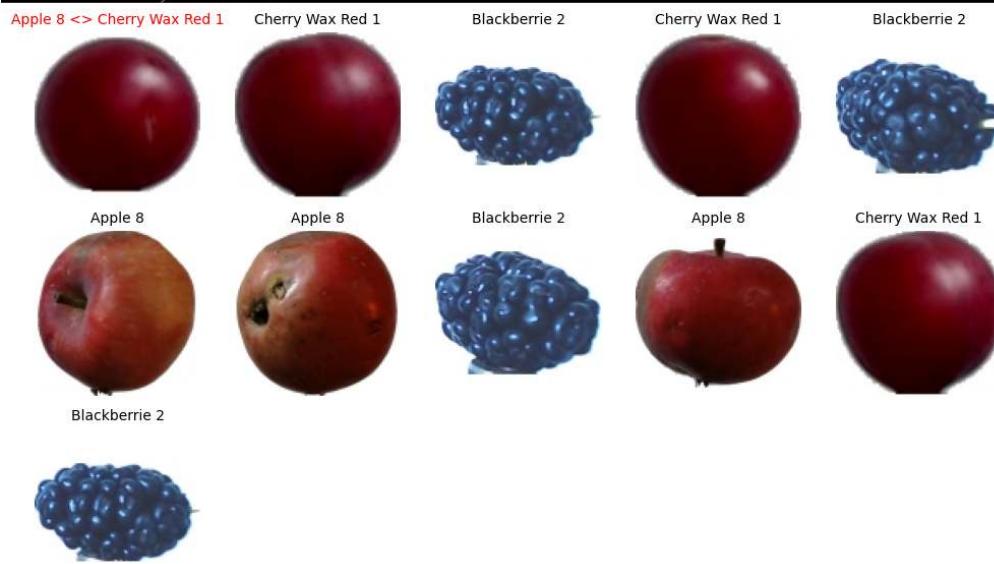
```
@assets publish \
    extensions=png,push .
```

```
fruits-365-dataset-2025-07-01-gn9up7 | 02 July 2025 | 09:19:32 (+0330) | 99
record(s) | 3 subset(s): train: 83 [%83.8], test: 5 [%5.1], eval: 11 [%11.1] | 3
class(es): Apple 8: 33 [%33.3], Blackberrie 2: 33 [%33.3], Cherry Wax Red 1: 33
[%33.3] | shape: 100x100x3 | batch_size: 16 | num_epochs: 10 | eval_accuracy:
90.91%
```



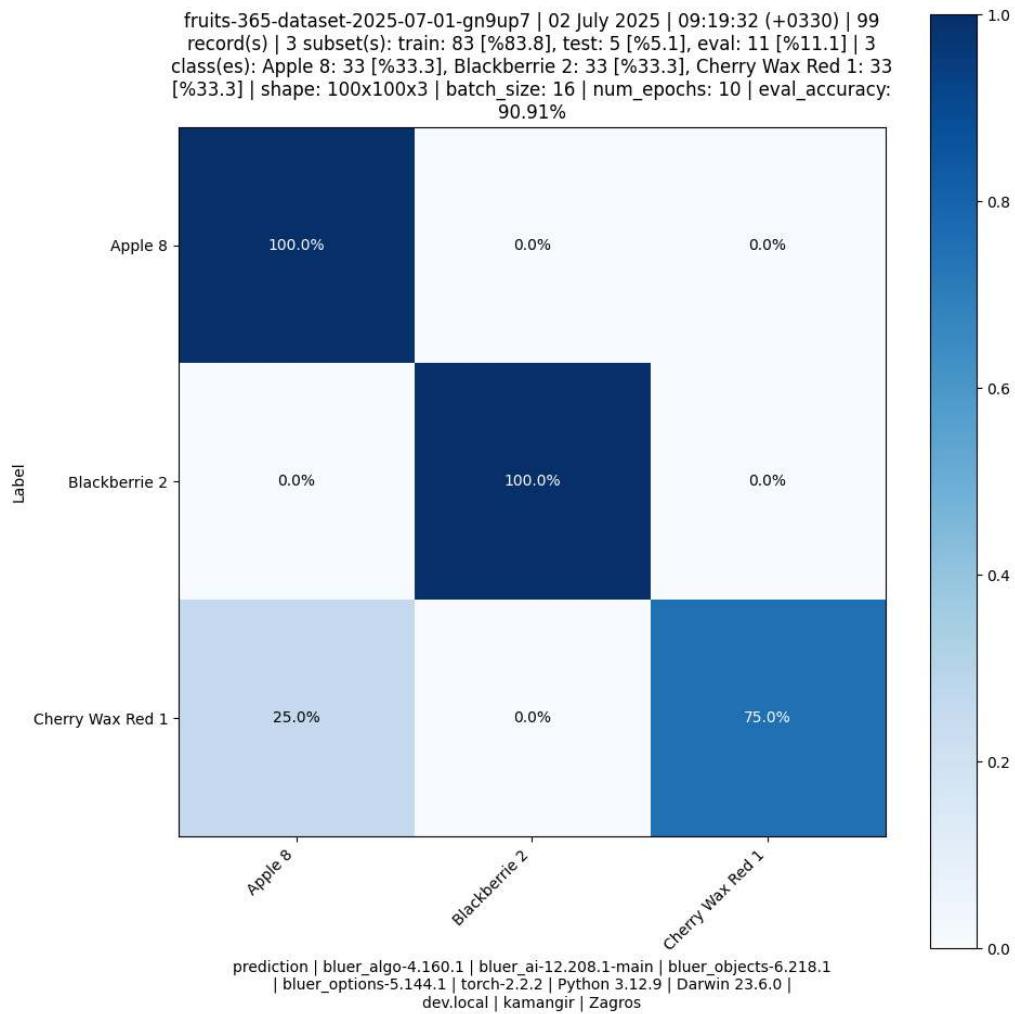
image

fruits-365-model-2025-07-02-fvfmt/evaluation.png | 02 July 2025 | 09:19:33 (+0330) | - | fruits-365-dataset-2025-07-01-gn9up7 | 02 July 2025 | 09:19:32 (+0330) | 99 record(s) | 3 subset(s): train: 83 [%83.8], test: 5 [%5.1], eval: 11 [%11.1] | 3 class(es): Apple 8: 33 [%33.3], Blackberry 2: 33 [%33.3], Cherry Wax Red 1: 33 [%33.3] | shape: 100x100x3 | batch_size: 16 | num_epochs: 10 | eval_accuracy: 90.91%



bluer_algo-4.160.1 | bluer_ai-12.208.1-main | bluer_objects-6.218.1 | bluer_options-5.144.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev.local | kamangir | Zagros

image



image

[fruits-365-model-2025-07-02-fvfmt](#)

```
model:  
  dataset:  
    class_count: 3  
    classes:  
      0: Apple 8  
      1: Blackberry 2  
      2: Cherry Wax Red 1  
    count: 99  
    shape:  
      - 100  
      - 100  
      - 3  
  evaluation:  
    class_accuracy:  
      0: 1.0  
      1: 1.0  
      2: 0.75  
    eval_accuracy: 0.9090909090909091  
  inputs:  
    batch_size: 16  
    num_epochs: 10  
  training:  
    loss:  
      - 1.1059847007314842  
      - 1.0944474734455707  
      - 1.081830551825374  
      - 1.0557171982454967  
      - 1.0102007518331688  
      - 0.9688218681209059  
      - 0.9016210835382162  
      - 0.7881300334470818  
      - 0.6774013846753592  
      - 0.5369924443313875
```

image-classifier: model: train: large

```
@select fruits-365-dataset-2000-$(@@timestamp)

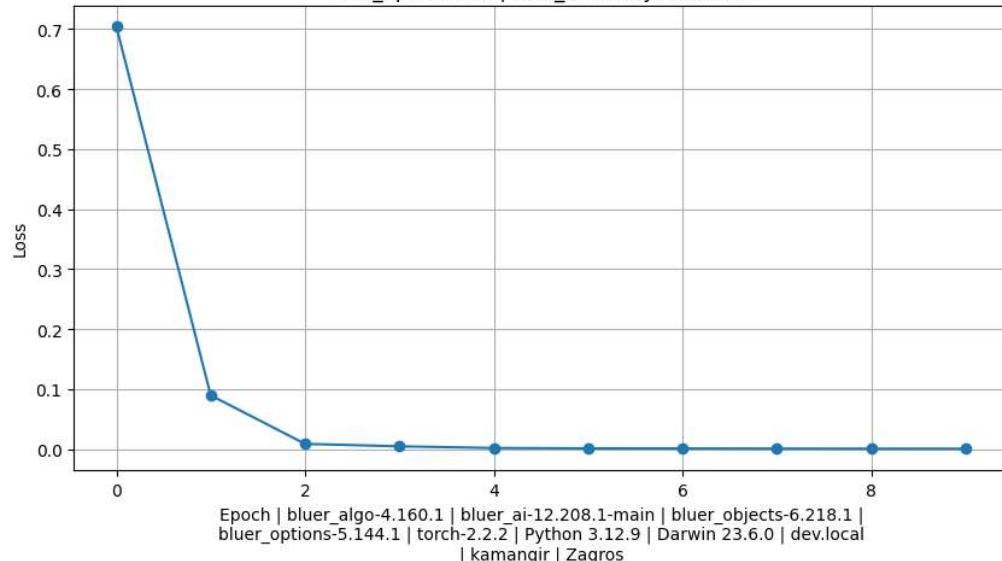
@algo image_classifier dataset ingest \
  clone,count=10000,source=fruits_360 . \
  --class_count 3

@select fruits-365-model-2000-$(@@timestamp)

@algo image_classifier model train \
  ~download,upload ... .

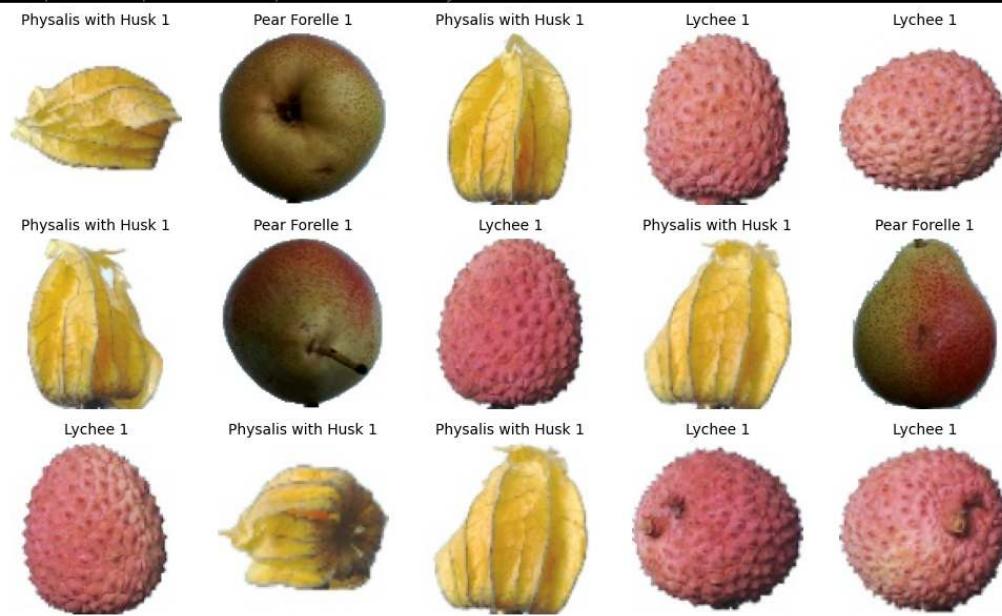
@upload public,zip .
@assets publish \
  extensions=png,push .
```

fruits-365-dataset-2000-2025-07-02-uzhotr | 02 July 2025 | 09:20:55 (+0330) |
 1684 record(s) | 3 subset(s): train: 1368 [%81.2], test: 165 [%9.8], eval: 151
 [%9.0] | 3 class(es): Lychee 1: 490 [%29.1], Pear Forelle 1: 702 [%41.7],
 Physalis with Husk 1: 492 [%29.2] | shape: 100x100x3 | batch_size: 16 |
 num_epochs: 10 | eval_accuracy: 100.00%



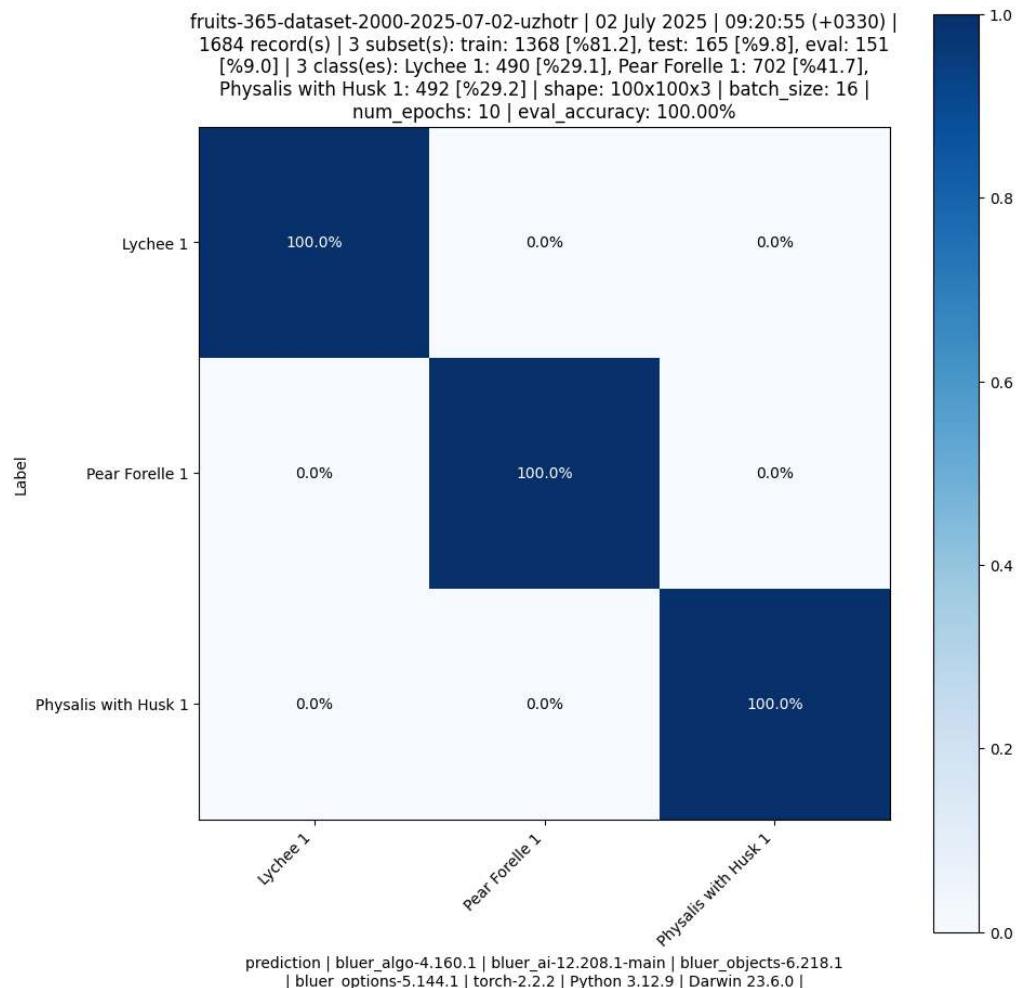
image

fruits-365-model-2000-2025-07-02-k318ju/evaluation.png | 02 July 2025 | 09:20:56
 (+0330) | - | fruits-365-dataset-2000-2025-07-02-uzhotr | 02 July 2025 |
 09:20:55 (+0330) | 1684 record(s) | 3 subset(s): train: 1368 [%81.2], test: 165
 [%9.8], eval: 151 [%9.0] | 3 class(es): Lychee 1: 490 [%29.1], Pear Forelle 1:
 702 [%41.7], Physalis with Husk 1: 492 [%29.2] | shape: 100x100x3 | batch_size:
 16 | num_epochs: 10 | eval_accuracy: 100.00%



bluer_algo-4.160.1 | bluer_ai-12.208.1-main | bluer_objects-6.218.1 |
 bluer_options-5.144.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev.local
 | kamangir | Zagros

image



image

[fruits-365-model-2000-2025-07-02-k318ju](#)

```

model:
dataset:
  class_count: 3
  classes:
    0: Lychee 1
    1: Pear Forelle 1
    2: Physalis with Husk 1
  count: 1684
  shape:
    - 100
    - 100
    - 3
evaluation:
  class_accuracy:
    0: 1.0
    1: 1.0
    2: 1.0
  eval_accuracy: 1.0
inputs:
  batch_size: 16
  num_epochs: 10
training:
  loss:
    - 0.7043376307912738
    - 0.08943271208881286
    - 0.008551867894572343

```

- **0.004447948475223993**
- **0.0016600372687074743**
- **0.0010692106054555244**
- **0.0008688547763543214**
- **0.0006009176277234873**
- **0.0004981696757456296**
- **0.0004295692094449789**

Image_classifier: model: predict

- [dev](#)
- [rpi](#)

image-classifier: model: prediction: dev

uses [train](#).

[image_classifier_prediction.ipynb](#)

```
image_classifier-prediction-2025-07-02-13-33-50-lgr380 | 02 July 2025 | 13:33:50  
(+0330) | model: fruits-365-model-2025-07-02-fvfomt | 3 class(es): #0: Apple 8,  
#1: Blackberrie 2, #2: Cherry Wax Red 1 | shape: 100x100x3 | prediction: Apple 8  
[#0] | correct | took 015 ms | bluer_algo-4.179.1 | bluer_ai-12.208.1 |  
bluer_objects-6.220.1 | bluer_options-5.144.1 | torch-2.2.2 | Python 3.12.9 |  
Darwin 23.6.0 | | Jupyter-Notebook
```



image

```
prediction:  
elapsed_time: 0.015111207962036133  
predicted_class: 0
```

image-classifier: model: prediction: rpi

uses [train](#).

```
@algo test what=test_bluer_algo_image_classifier_model_prediction_test \
upload
```

```
image_classifier-prediction-2025-07-02-16-11-03-0kb1d3 | 02 July 2025 | 16:11:39
(+0330) | model: fruits-365-model-2025-07-02-fvfomt | 3 class(es): #0: Apple 8,
#1: Blackberry 2, #2: Cherry Wax Red 1 | shape: 100x100x3 | prediction: Apple 8
[#0] | correct | took 081 ms | bluer_algo-4.185.1 | bluer_ai-12.208.1-main |
bluer_objects-6.220.1 | bluer_options-5.139.1 | torch-2.4.1 | Python 3.8.10 |
Linux 5.4.0-1129-raspi | swallow | 7p7yq | Sion
```



image

```
prediction:
elapsed_time: 0.08102989196777344
predicted_class: 0
```

tracker

target tracking for a [ugv](#) using [mean/cam-shift](#).

- sandbox: [sandbox/mean-cam-shift](#)
- [camshift](#)
- [meanshift](#)

 conclusion: camshift tracks distinct colors robustly and adjusts to object size.

tracker: camshift

on a video file

```
@select tracker-camshift-$(@timestamp)

@algo tracker \
  algo=camshift . \
  --log 1

@assets publish \
  extensions=gif,push .
```

on camera feed

```
@select tracker-camshift-$(@timestamp)

@algo tracker \
  algo=camshift,camera . \
  --log 1 \
  --show_gui 1

@assets publish \
  extensions=gif,push .
```



tracker: meanshift

on a video file

```
@select tracker-meanshift-$(@timestamp)

@algo tracker \
  algo=meanshift . \
  --log 1

@assets publish \
  extensions=gif,push .
```

on camera feed

```
@select tracker-meanshift-$(@timestamp)

@algo tracker \
  algo=meanshift,camera . \
  --log 1 \
  --show_gui 1

@assets publish \
  extensions=gif,push .
```



yolo

target tracking for a [ugv](#) using [yolo](#).

- [dataset](#)
- [model](#)

yolo: dataset

- [ingest and review](#)

yolo: dataset: ingest-and-review

full

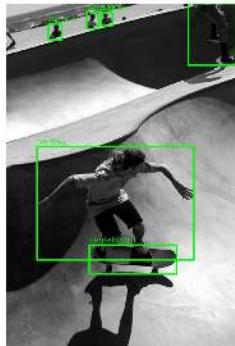
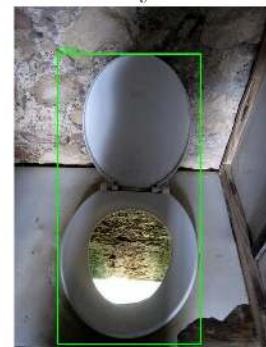
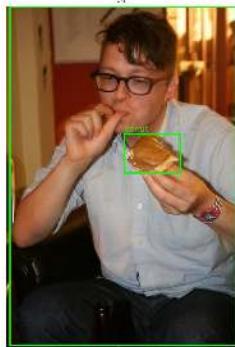
```
@select coco128-$(@@timestamp)

@yolo dataset ingest \
    upload .

@yolo dataset review \
    ~download .

@upload public,zip .
@assets publish \
    extensions=png,push .
```

coco128-2025-09-14-q6oxj4/review.png | 14 September 2025 | 15:56:08 (+0330) |
 480x640x3:uint8 | count: 126 | 80 class(es): person, bicycle, car, motorcycle,
 airplane, ...



bluer_algo-4.378.1 | bluer_ai-12.259.1-main | bluer_objects-6.258.1 |
 bluer_options-5.164.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
 kamangir | Sion

image

[coco128-2025-09-14-q6oxj4](#)

```
dataset:
  count: 126
names:
  0: person
  1: bicycle
  2: car
  3: motorcycle
```

4: airplane
5: bus
6: train
7: truck
8: boat
9: traffic light
10: fire hydrant
11: stop sign
12: parking meter
13: bench
14: bird
15: cat
16: dog
17: horse
18: sheep
19: cow
20: elephant
21: bear
22: zebra
23: giraffe
24: backpack
25: umbrella
26: handbag
27: tie
28: suitcase
29: frisbee
30: skis
31: snowboard
32: sports ball
33: kite
34: baseball bat
35: baseball glove
36: skateboard
37: surfboard
38: tennis racket
39: bottle
40: wine glass
41: cup
42: fork
43: knife
44: spoon
45: bowl
46: banana
47: apple
48: sandwich
49: orange
50: broccoli
51: carrot
52: hot dog
53: pizza
54: donut
55: cake
56: chair
57: couch
58: potted plant
59: bed
60: dining table
61: toilet
62: tv
63: laptop
64: mouse
65: remote
66: keyboard
67: cell phone
68: microwave

```

69: oven
70: toaster
71: sink
72: refrigerator
73: book
74: clock
75: vase
76: scissors
77: teddy bear
78: hair drier
79: toothbrush
source: coco_128
train: coco128/images/train2017
val: coco128/images/train2017

```

select classes

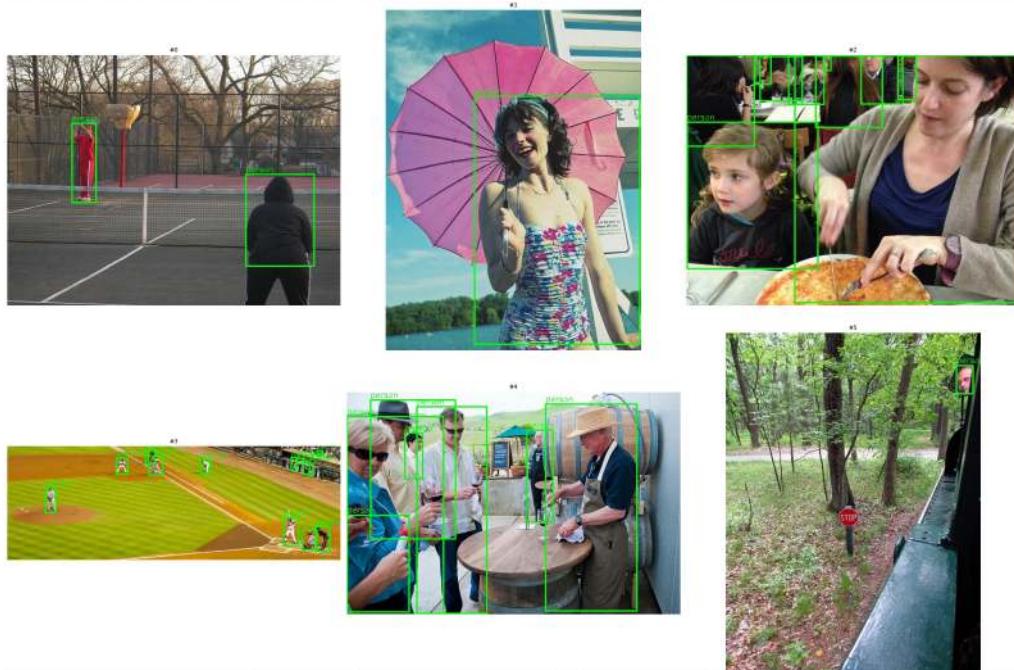
```
@select coco128-$(@@timestamp)
```

```
@yolo dataset ingest \
upload . \
--classes person+boat
```

```
@yolo dataset review \
~download .
```

```
@upload public,zip .
@assets publish \
extensions=png,push .
```

coco128-2025-09-14-tub72c/review.png | 14 September 2025 | 16:02:32 (+0330) |
640x480x3:uint8 | count: 63 | 2 class(es): person, boat



bluer_algo-4.380.1 | bluer_ai-12.259.1-main | bluer_objects-6.258.1 |
bluer_options-5.164.1 | torch-2.2.2 | Python 3.12.9 | Darwin 23.6.0 | dev |
kamangir | Sion

image

[coco128-2025-09-14-tub72c](#)

```
dataset:
count: 63
```

```
names:  
  0: person  
  1: boat  
source: coco_128  
train: coco128/images/train2017  
val: coco128/images/train2017
```

yolo: model

- [validation](#)

yolo: model: validation

ingest

```
@select coco128-$(@@timestamp)

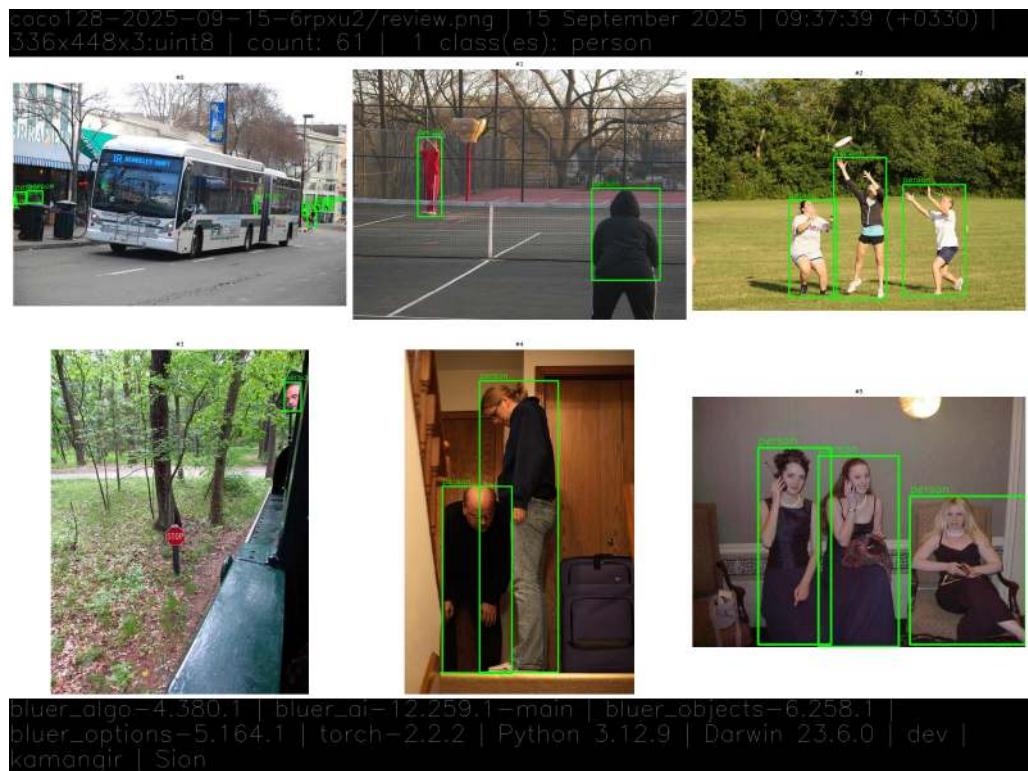
@yolo dataset ingest - . \
    --classes person

@yolo dataset review \
    ~download .

@upload public,zip

@assets publish \
    extensions=png,push .
```

[coco128-2025-09-15-6rpxu2](#)



image

```
dataset:
  count: 61
names:
  0: person
source: coco_128
train: coco128/images/train2017
val: coco128/images/train2017
```

train

```
@select coco128-model-$(@@timestamp)
```

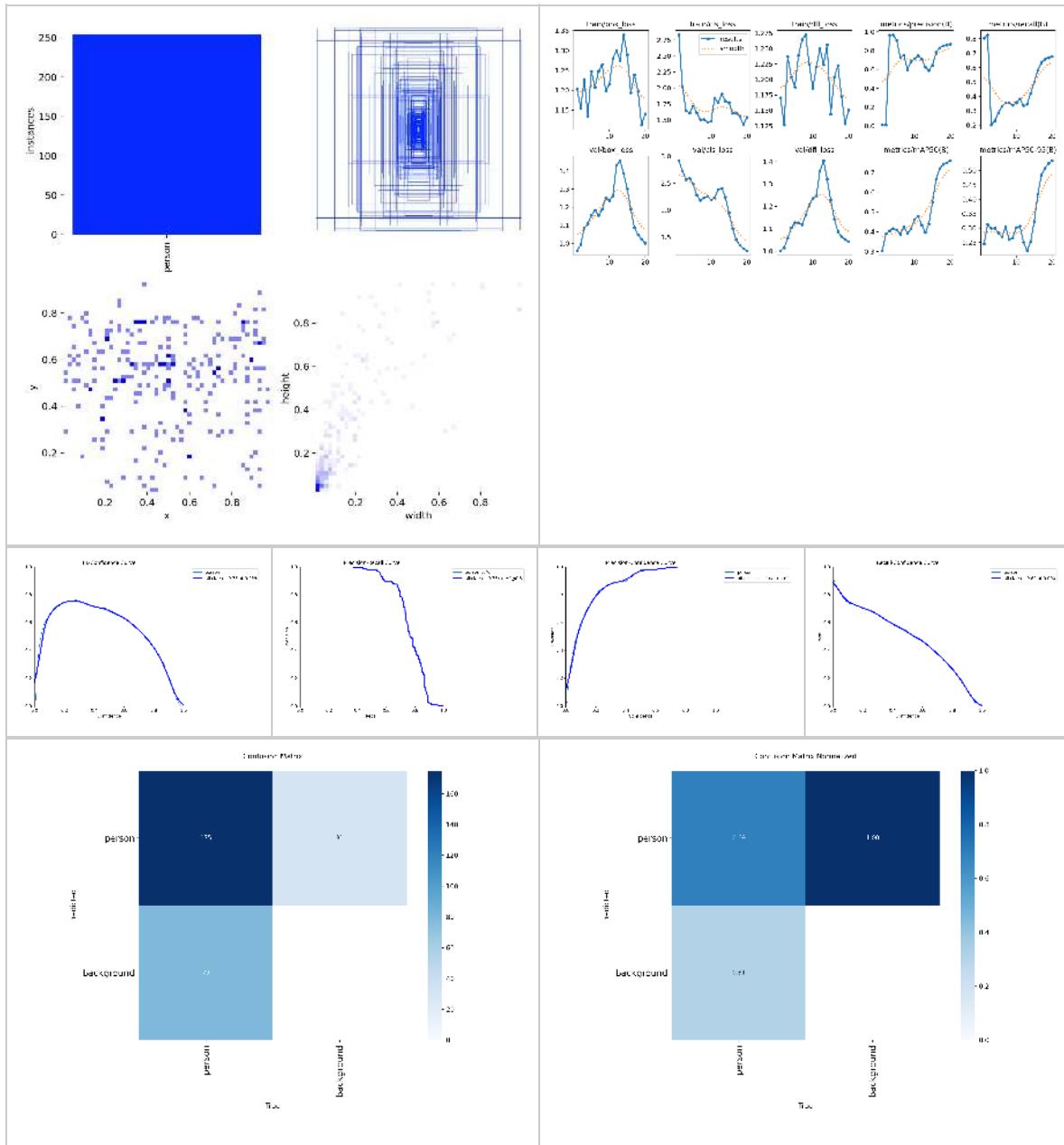
```
@yolo model train \
~download,upload ... \
--epochs 20
```

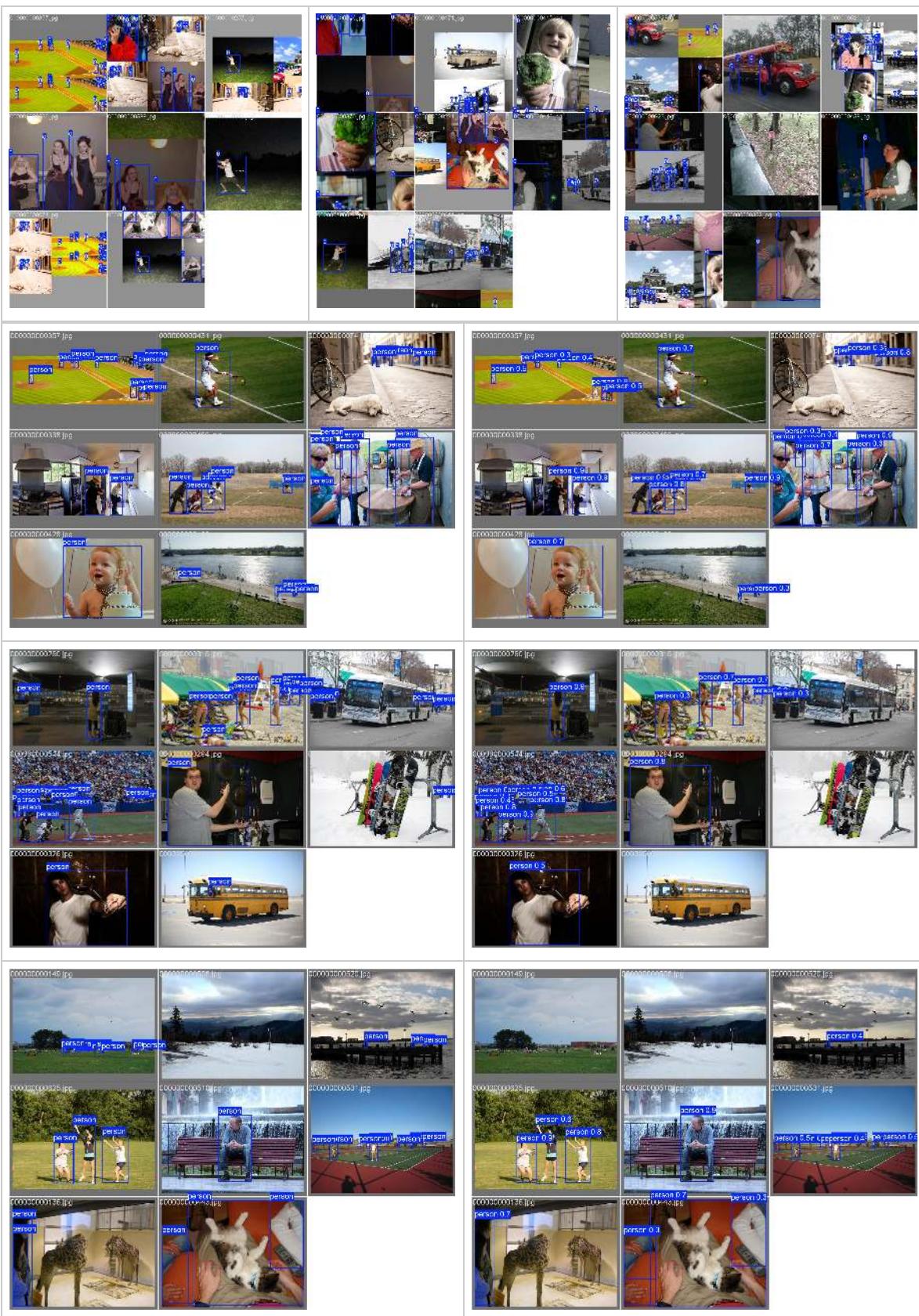
```
@upload public,zip
```

```
@assets publish \
extensions=jpg+png . \
--prefix train/
```

```
@assets publish \
extensions=jpg+png,push . \
--prefix validation/
```

[coco128-model-2025-09-15-3m0mmj](#)





► metadata

predict (dev)

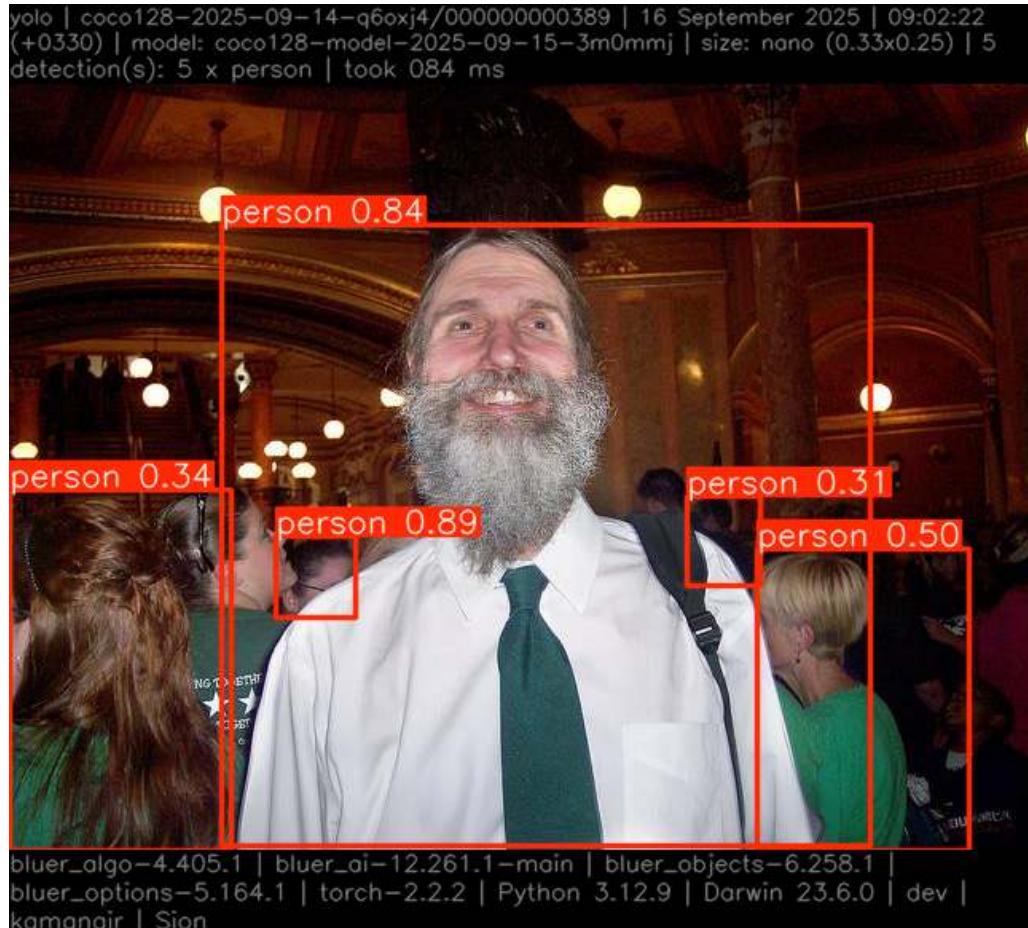
[yolo_prediction.ipynb](#)

predict

```
@select yolo-prediction-test-$(@timestamp)
```

```
@yolo model prediction_test \
upload \
$BLUER_ALGO_COCO128_TEST_DATASET \
$BLUER_ALGO_COCO128_TEST_MODEL . \
--record_index 3
```

```
@assets publish extensions=png,push
```



image

```
000000000389:
detections:
- bbox_xyxy:
  - 167.85377502441406
  - 283.70269775390625
  - 217.58558654785156
  - 334.81072998046875
  class_id: 0
  confidence: 0.8922585248947144
  label: person
- bbox_xyxy:
  - 133.1409912109375
  - 88.43701171875
  - 540.6605834960938
  - 477.65869140625
  class_id: 0
  confidence: 0.842871367931366
  label: person
```

```

- bbox_xyxy:
  - 469.0397644042969
  - 291.564697265625
  - 602.921875
  - 480.0
  class_id: 0
  confidence: 0.49844595789909363
  label: person
- bbox_xyxy:
  - 0.58984375
  - 254.95266723632812
  - 139.97897338867188
  - 480.0
  class_id: 0
  confidence: 0.33697929978370667
  label: person
- bbox_xyxy:
  - 425.3193359375
  - 258.2589416503906
  - 471.283935546875
  - 314.4782409667969
  class_id: 0
  confidence: 0.31127116084098816
  label: person
elapsed_time: 0.08387088775634766
image_size:
- 480
- 640

```

predict (rpi)

```

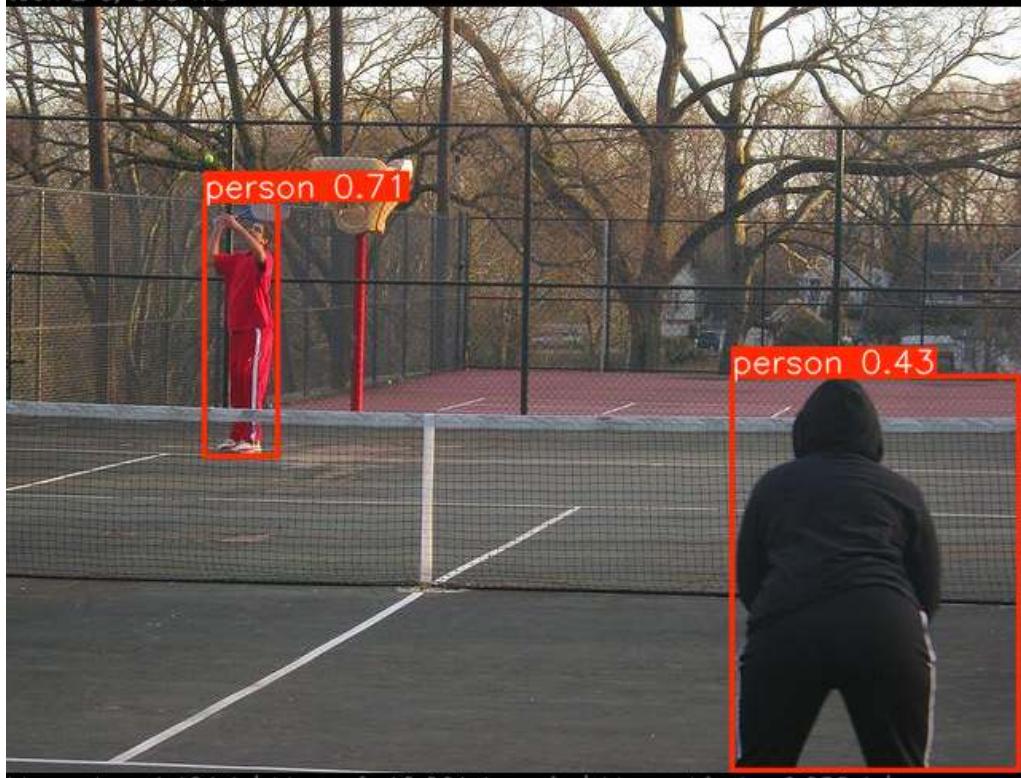
@select yolo-prediction-test-$(@timestamp)

@yolo model prediction_test \
  upload \
  $BLUER_ALGO_COCO128_TEST_DATASET \
  $BLUER_ALGO_COCO128_TEST_MODEL . \
  --record_index 3

@assets publish extensions=png,push

```

yolo | coco128-2025-09-14-q6oxj4/000000000419 | 16 September 2025 | 09:03:53
 (+0330) | model: coco128-model-2025-09-15-3m0mmj | 2 detection(s): 2 x person |
 took 2 s, 340 ms



bluer_algo-4.404.1 | bluer_oi-12.261.1-main | bluer_objects-6.258.1 |
 bluer_options-5.164.1 | torch-2.8.0+cpu | Python 3.11.2 | Linux 6.12.25+rpt-
 rpi-v8 | sparrow | 000000000953c665 | Sion

image

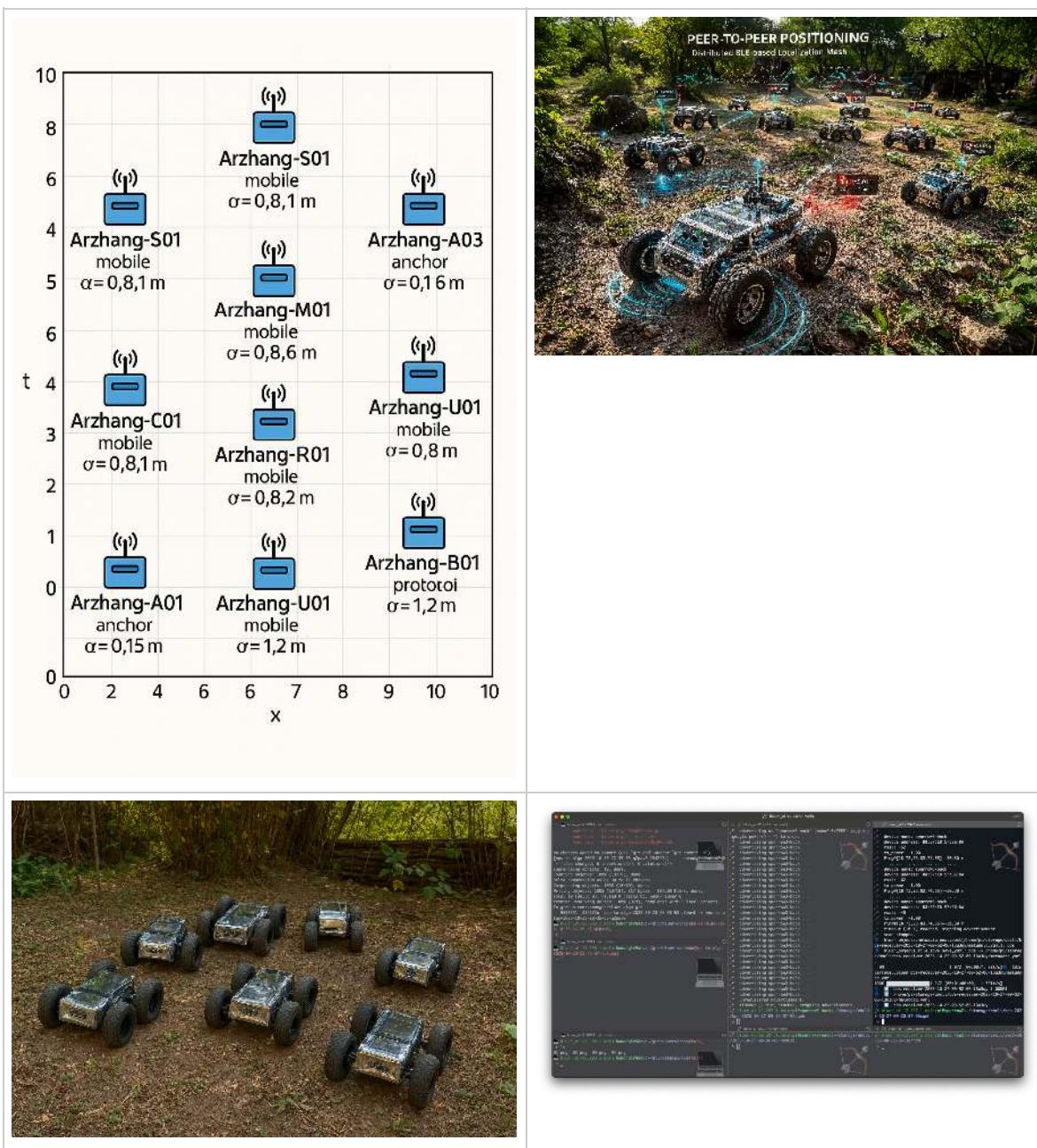
```
000000000419:
detections:
- bbox_xyxy:
  - 124.66294860839844
  - 121.88188934326172
  - 170.0297393798828
  - 281.0235595703125
  class_id: 0
  confidence: 0.7116779088973999
  label: person
- bbox_xyxy:
  - 455.524658203125
  - 231.17758178710938
  - 635.5400390625
  - 478.5752868652344
  class_id: 0
  confidence: 0.4342321455478668
  label: person
elapsed_time: 2.339571714401245
image_size:
- 480
- 640
```

bps

bluer-positioning system.

A distributed, serverless positioning mesh where every Raspberry Pi acts as both a tag and a local estimator: each node continuously advertises a compact BLE packet containing its node ID, mobility flag, and its current best-estimate pose (x,y,σ,seq), and simultaneously scans for neighbor adverts to collect RSSI + neighbor poses. Anchors are marked static with known coordinates and low uncertainty; mobiles fuse anchors + neighbor pseudo-ranges (RSSI \rightarrow distance, calibrated per site) with a lightweight weighted least-squares + EKF/Covariance-Intersection step, adapt advertising/scan rates when moving, and use short rolling HMAC tokens to prevent simple spoofing. The result is a gossiping swarm that converges locally to consistent positions, degrades gracefully to coarse proximity when RF is noisy, and can be migrated later to UWB without changing the peer-to-peer protocol.

- [AI convo](#)
- [literature](#)
- [mathematics](#)
- [sandbox](#)
- [bluer algo.bps](#)
- [validations](#)
- [simulations](#)



works on rpi only.

tx-power is not implemented in rpi. nominal value: 10-12 dBm. -1: indicates unknown.

bps: literature

- [Research Progress of Wireless Positioning Methods Based on RSSI](#) - 2024

A modern survey of RSSI-based positioning techniques across BLE, Wi-Fi, ZigBee, and 5G. It explains theoretical limitations of RSSI (path-loss variability, multipath fading) and outlines correction strategies.

- [Bluetooth indoor positioning system based on improved RSSI data](#) - 2024
- [Peer-To-Peer UWB Ranges as a Source of Training Data for Estimating BLE RSSI Path-Loss Exponents](#) - 2022

Proposes using precise UWB peer-to-peer ranges to calibrate BLE path-loss parameters automatically. Shows how multi-modal measurements improve BLE localization accuracy by up to 40%.

- [A Survey of Robot Swarms' Relative Localization Methods](#) - 2022

A comprehensive review of how robot swarms estimate relative positions using sensors such as BLE, UWB, infrared, and vision. It categorizes algorithms by communication type, sensing range, and scalability.

- [From Robot Self-Localization to Global-Localization: An RSSI Based Approach](#) (ArXiv) - 2021

presents a **simulation-based** method for enabling groups of mobile robots to localize themselves in GPS-denied environments. Some robots act as fixed beacons, broadcasting signals whose received strength (RSSI) is used by other robots to estimate their global position **through geometric reconstruction**. The approach was tested only in simulation (Webots) and not on physical robots. Reported average errors were about 0.6 m with 10 % Gaussian noise, 1.4 m with 20 % noise, and ≈0.1 m in near-ideal short-range conditions (3–3.5 m from beacons); within about 6 m of beacons, the total error remained under 1 m.

- [Bluetooth Low Energy Technology Applied to Indoor Positioning Systems: An Overview](#) - 2020
- [RSSI Filtering Methods Applied to Localization using Bluetooth Low Energy](#) - 2020

Focuses on the signal-processing side of BLE localization. It evaluates Kalman filters, moving averages, and histogram-based smoothing to mitigate RSSI noise.

- [Indoor localization using radio beacon technology](#) - 2018

bps: mathematics

- [timing](#)
- [localization](#)

bps: mathematics: timing

optimizing advertise/receive cycle timing - continues [v1](#).

problem setup

each machine runs the following infinite loop:

1. **advertise** for t_a seconds
2. **receive** (scan) for a random duration $U(t_{r1}, t_{r2})$ seconds

each process, advertise ($x = a$) or receive ($x = r$), takes:

- t_{xo} seconds to open
- t_{xc} seconds to close

define the total per-phase overhead:

$$t_{as} = t_{ao} + t_{ac}$$

$$t_{rs} = t_{ro} + t_{rc}$$

goal: Choose t_a , t_{r1} , and t_{r2} to maximize the expected number of advertisements received among all machines.

derivation

total cycle duration:

$$T = t_a + t_{as} + t_{rm} + t_{rs}$$

where,

$$\$ \$ t_{rm} = \frac{1}{2}(t_{r1} + t_{r2}) : \text{mean receive time} \$ \$$$

for many unsynchronized machines, the expected overlap fraction between “me listening” and “others advertising” is proportional to:

$$\$ \$ P \propto \frac{t_a}{T} * \frac{t_{rm}}{T} \$ \$$$

optimal relationship

for a fixed T ,

$$t_a + t_{rm} = T - (t_{as} + t_{rs})$$

is fixed and P is maximized when,

$$\$ \$ t_a = t_{rm} = \frac{1}{2}(T - t_{as} - t_{rs}) \$ \$$$

$$\$ \$ P \propto \frac{1}{4} \left(1 - \frac{t_{as} + t_{rs}}{T}\right)^2 \$ \$$$

therefore larger T is more optimal for higher overlap.

the expected reaction time is,

$$\$ \$ R \propto T - t_a = \frac{1}{2}(T + t_{as} + t_{rs}) \$ \$$$

therefore, smaller T is more optimal for faster reaction.

summary

- match advertise and mean receive times: $t_a = (t_{r1} + t_{r2})/2$
- keep both much longer than $t_{as} + t_{rs}$
- add 10–30% jitter to t_{r1}/t_{r2} to avoid synchronization collisions
- choose smaller t_a to for faster reaction.

practical choices

system parameters,

t_{ao}	t_{ac}	t_{as}	t_{ro}	t_{rc}	t_{rs}
~1 s	~0 s	~1 s	~3 s	~0.5 s	~3s

chose:

- $t_a = 30$ s
- $r_{r1} = 24$ s
- $r_{r2} = 36$ s
- jitter: 20 %
- $T: 64$ s
- $P: \sim 22\%$
- $R: 34$ s

bps: mathematics: localization

BLE Localization Cost Function with σ

This document defines the mathematical model used to estimate the position of a UGV based on BLE advertisements.

Each beacon advertises its position (x_i, y_i, z_i) and uncertainty σ_i along with its transmitted power.

1. RSSI–Distance Model

The received signal strength indicator (RSSI) is related to distance by the log-distance path-loss model:

$$\text{RSSI}_i = P_{\text{tx}, i} - 10 n_i \log_{10}(d_i) + \varepsilon_i$$

where:

- $P_{\text{tx}, i}$ is the transmit power (dBm) of beacon i
- n_i is the path-loss exponent (≈ 2 for open space, higher indoors)
- ε_i is zero-mean Gaussian noise with standard deviation σ_{rss}, i

Rearranging gives an estimate of the distance to beacon i :

$$d_i = 10^{(P_{\text{tx}, i} - \text{RSSI}_i)/(10 n_i)}$$

2. Measurement Model

Let the UGV's unknown position be

$$\mathbf{p} = (x, y, z)$$

Each beacon i has a known position

$$\mathbf{p}_i = (x_i, y_i, z_i)$$

The measured distance d_i relates to the true distance by

$$r_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_i\| - d_i$$

where r_i is the residual of beacon i .

3. Variance and σ Incorporation

Each measurement has an associated variance s_i^2 that combines:

1. Range noise from RSSI
2. Beacon's advertised position uncertainty

3.1 Range variance (from RSSI noise)

Using first-order error propagation:

$$\text{Var}(d_i) \approx \left(\frac{\partial d_i}{\partial \text{RSSI}_i} \right)^2 \text{Var}(\text{RSSI}_i) = \left(\frac{\ln(10)}{10 n_i} \right)^2 s_{\text{RSSI}, i}^2$$

(All variables are scalar; this is LaTeX-safe.)

3.2 Beacon position variance

If a beacon advertises an isotropic uncertainty σ_i :

$$\text{Var}_{\text{beacon}, i} \approx \sigma_i^2$$

3.3 Total variance per measurement

$$s_i^2 = \text{Var}(d_i) + \text{Var}_{\text{beacon}, i}$$

4. Weighted Least Squares Cost Function

The UGV's estimated position $\hat{\mathbf{p}}$ minimizes

$$J(\mathbf{p}) = \sum_i \frac{r_i(\mathbf{p})^2}{s_i^2} = \sum_i \frac{(p_i - d_i)^2}{s_i^2}$$

This is equivalent to maximum likelihood estimation under Gaussian noise with heterogeneous variances.

5. Estimating Global σ

After solving for $\hat{\mathbf{p}}$, compute the overall uncertainty:

$$\hat{\sigma} = \sqrt{\frac{1}{N-3} \sum_i r_i(\hat{\mathbf{p}})^2}$$

where N is the number of beacons and 3 is the number of position parameters (x, y, z).

6. Robust Form (Optional)

To reduce the effect of outliers (for example, multipath), replace the quadratic cost with a robust loss such as Huber or Cauchy:

$$\$ \$ J(\mathbf{p}) = \sum_i \rho\left(\frac{r_i(\mathbf{p})}{s_i}\right) \$ \$$$

where $\rho(\cdot)$ is a smooth function that limits the influence of large residuals.

7. Summary

Symbol	Meaning
\mathbf{p}	Current UGV position (x, y, z)
\mathbf{p}_i	Beacon i position (x_i, y_i, z_i)
d_i	Distance inferred from RSSI
r_i	Residual = geometric distance – estimated distance
s_i	Combined standard deviation per beacon
σ_i	Advertised beacon position uncertainty
$\hat{\sigma}$	Estimated global uncertainty

The weighted least-squares estimator is:

$$\$ \$ \hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \sum_i \frac{\left(\mathbf{p} - \mathbf{p}_i \right)^T \mathbf{p} - d_i^2}{s_i^2} \$ \$$$

bps: validations

- [test -> introspect](#)
- [beacon -> receiver](#)
- loop: [2 rpis, 3 rpis](#)
- [review](#)
- [data collection](#)
- live: [1 rpi, 2 rpis, 2 rpis, anchor + 2 nodes](#)

bps: validations: test-introspect

```
@bps install  
@bps test  
✓ bluer_algo.bps.utils.test: connected to system bus with unique name:  
:1.19  
✓ exported org.example.Hello at /org/example/Hello  
✓ run in another terminal: "@bps introspect unique_bus_name=:1.19"
```

in another terminal,

```
@bps introspect unique_bus_name=:1.19  
s "Pong"
```

validate in the first window,

```
✓ bluer_algo.bps.utils.test.ping() called by busctl!
```

tested on 2 rpis.

bps: validations: beacon-receiver

```
@bps beacon -- \
--generate 1 \
--sigma $($random --float 1) \
--x $($random --float 1) \
--y $($random --float 1) \
--z $($random --float 1) \
--timeout 60
```

on another rpi,

```
@bps receiver upload - \
--grep sparrow+swallow \
--timeout 10
```

```
history:
- hostname: sparrow3-back
  rssi: -52
  sigma: 88.34217834472656
  tx_power: -1.0
  x: 33.44688415527344
  y: 45.99796676635742
  z: 94.80066680908203
- hostname: sparrow3-back
  rssi: -51
  sigma: 88.34217834472656
  tx_power: -1.0
  x: 33.44688415527344
  y: 45.99796676635742
  z: 94.80066680908203
ping:
  hostname: sparrow2
  rssi: -1.0
  sigma: 1000.0
  tx_power: -1.0
  x: 0.0
  y: 0.0
  z: 0.0
```

bps: validations: loop-2

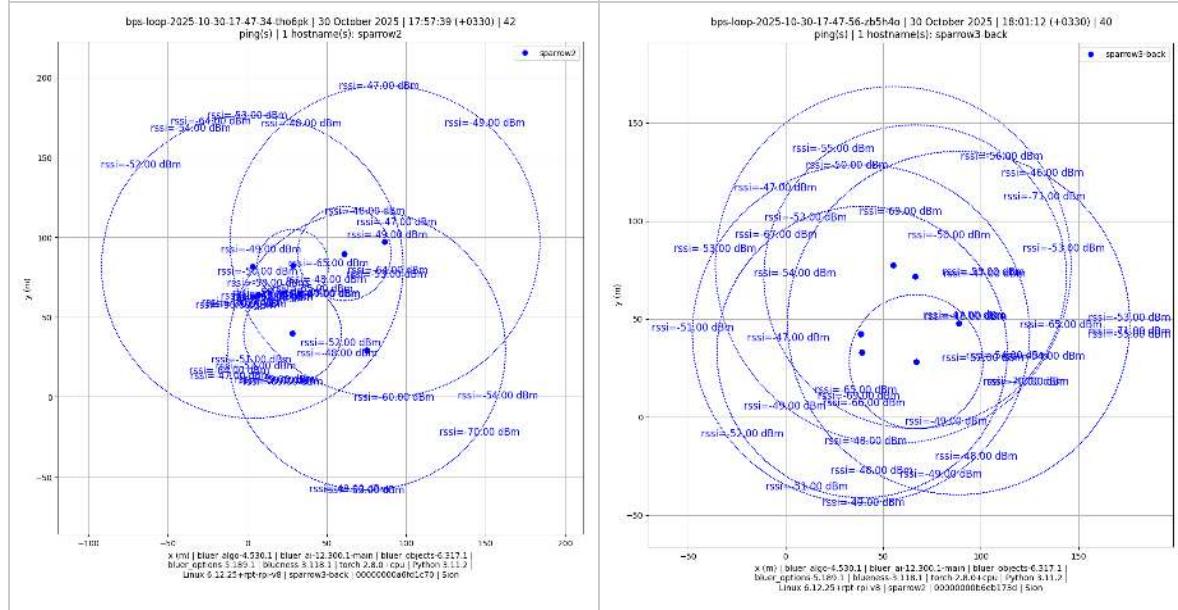
on 2 rpis,

@bps loop start simulate,upload

after a few minutes,

@bps loop stop [rpi <machine-name>]

► publication



bps: validations: loop-3

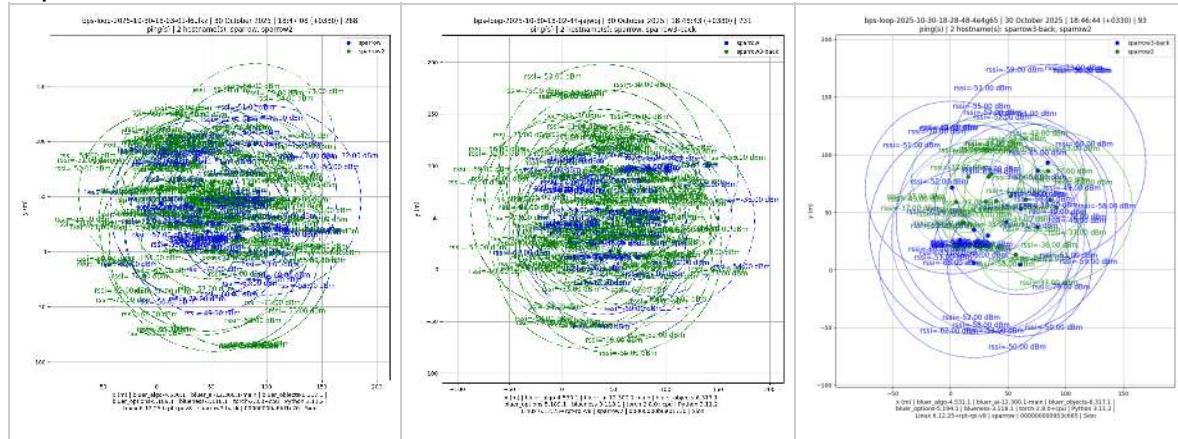
on 3 rpis,

```
@bps loop start simulate,upload
```

after a few minutes,

```
@bps loop stop [rpi <machine-name>]
```

► publication



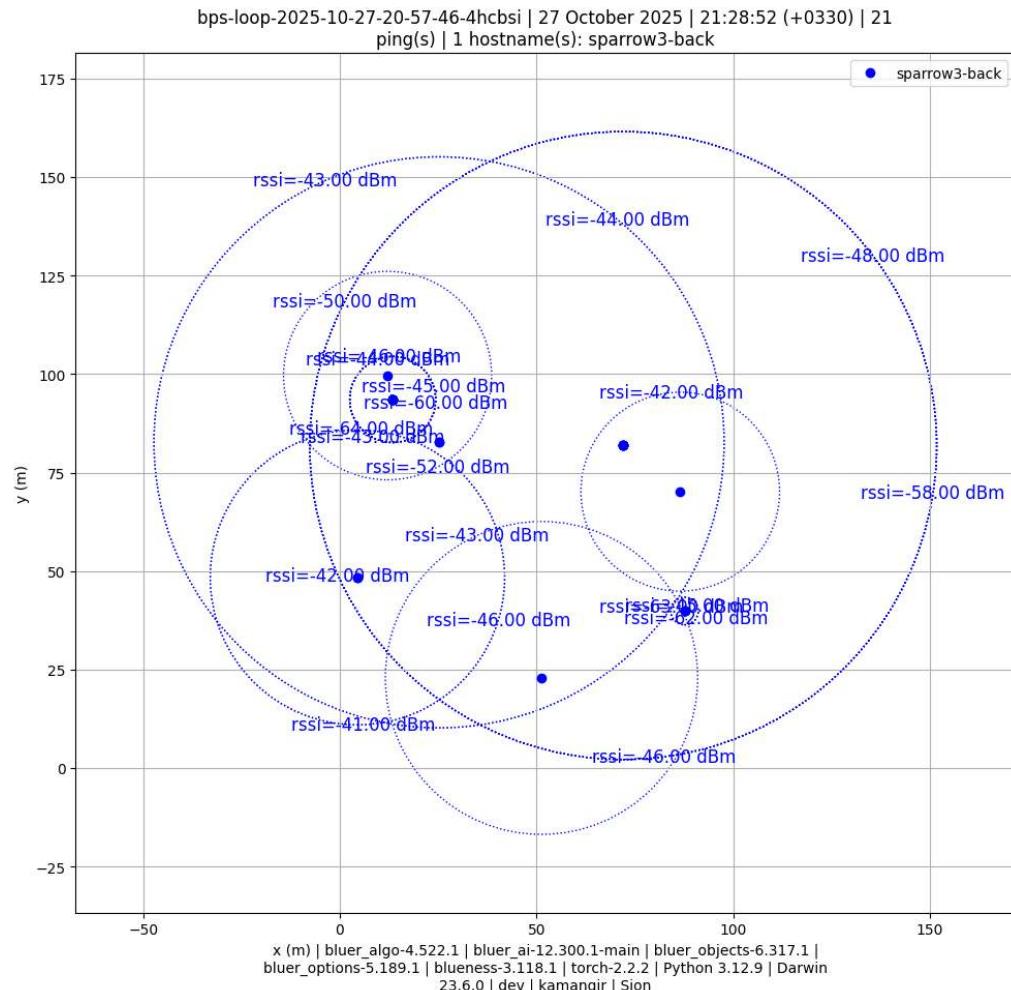
- bps-loop-2025-10-30-18-03-01-l6ulkz/metadata
- bps-loop-2025-10-30-18-02-44-jajwbj/metadata
- bps-loop-2025-10-30-18-28-48-4e4g65/metadata

bps: validations: review

```
@select bps-loop-2025-10-27-20-57-46-4hcbsi
```

```
@bps review upload .
```

```
@assets publish \
extensions=png,push .
```



image

► metadata

bps: validations: data-collection

on 1 rpi:

```
@select bps-stream-$(@timestamp)
@bps generate - . \
  --sigma 1.0 \
  --x 0 \
  --y 0 \
  --z 0
@bps loop start upload .
```

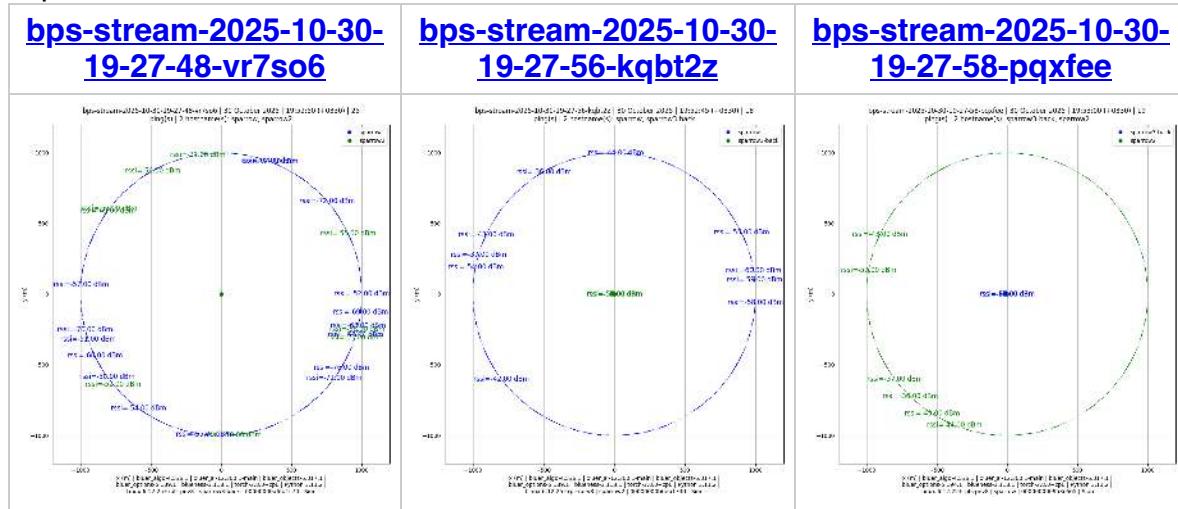
on 2 rpis,

```
@bps loop start upload
```

after a few minutes,

```
@bps loop stop [rpi <machine-name>]
```

► publication



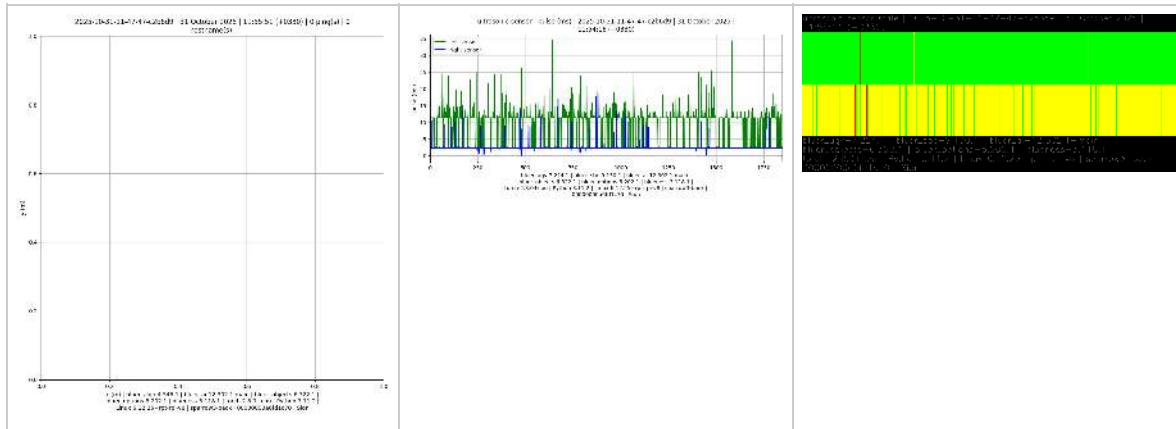
- [bps-stream-2025-10-30-19-27-48-vr7so6/metadata](#)
- [bps-stream-2025-10-30-19-27-56-kqbt2z/metadata](#)
- [bps-stream-2025-10-30-19-27-58-pqxfree/metadata](#)

bps: validations: live-1

on one rpi,

```
@swallow env set bps 1
@env HAS_BPS
BLUER_SBC_SWALLOW_HAS_BPS=1
@select; @session start
▶ publication
```

[2025-10-31-11-47-47-c2b6d9](#)

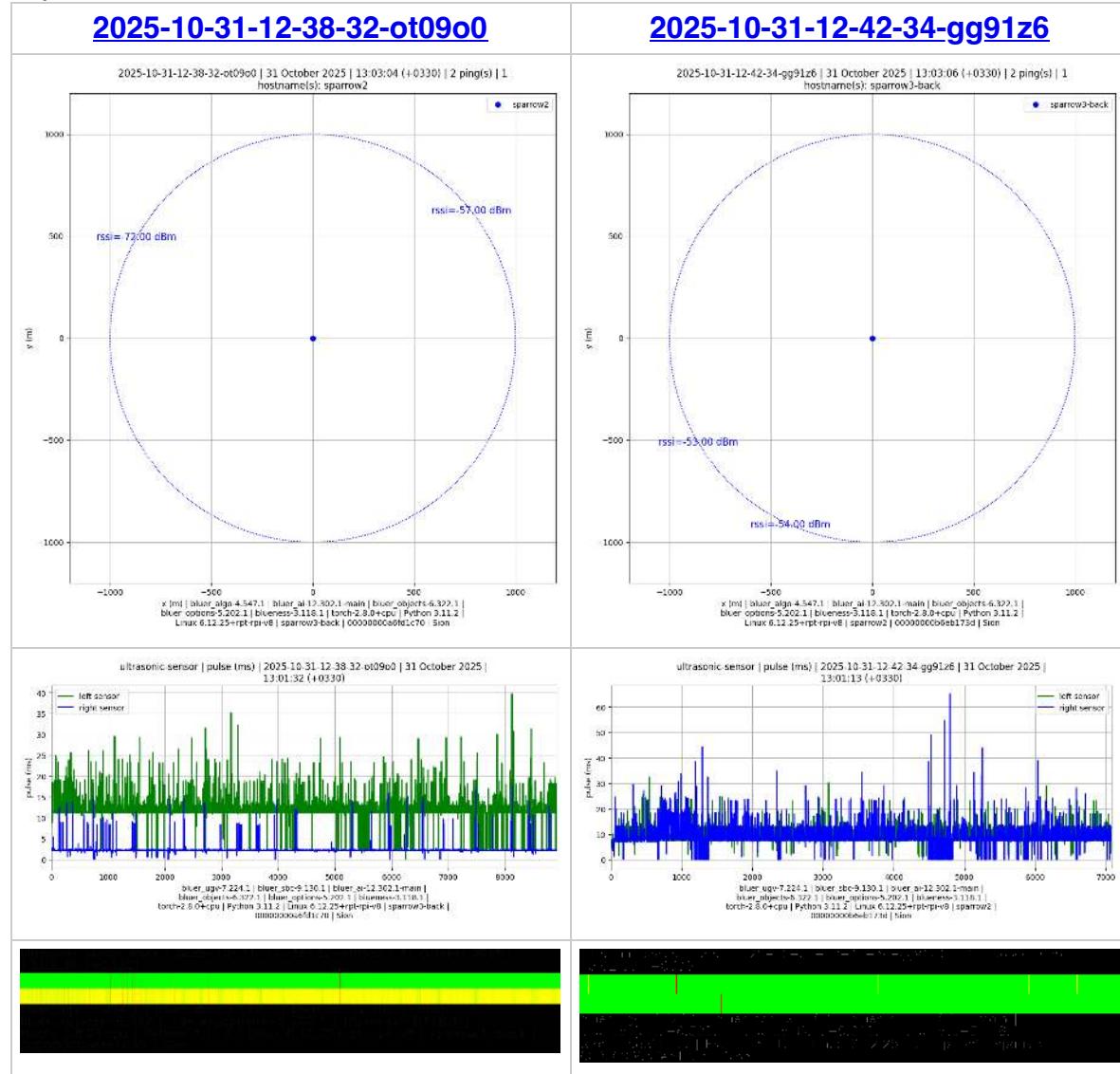


bps: validations: live-2

on 2 rpis,

```
@select; @session start
```

► publication

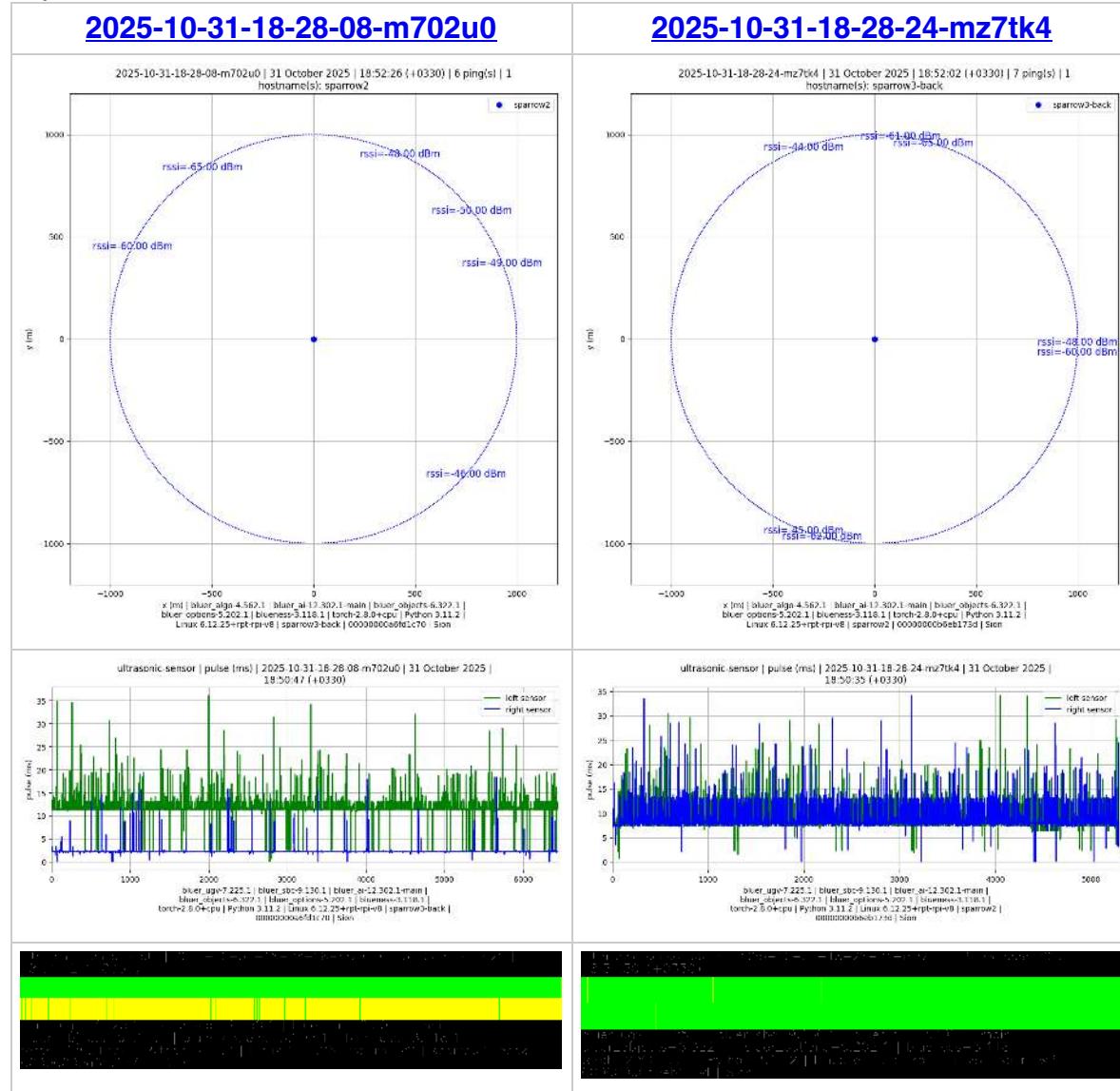


bps: validations: live-2b

on 2 rpis, continues [live-2](#) after timing adjustments.

@select; @session start

► publication



bps: validations: live-3

on 3 rpis (1 anchor), continues [live-2b](#) with an anchor.

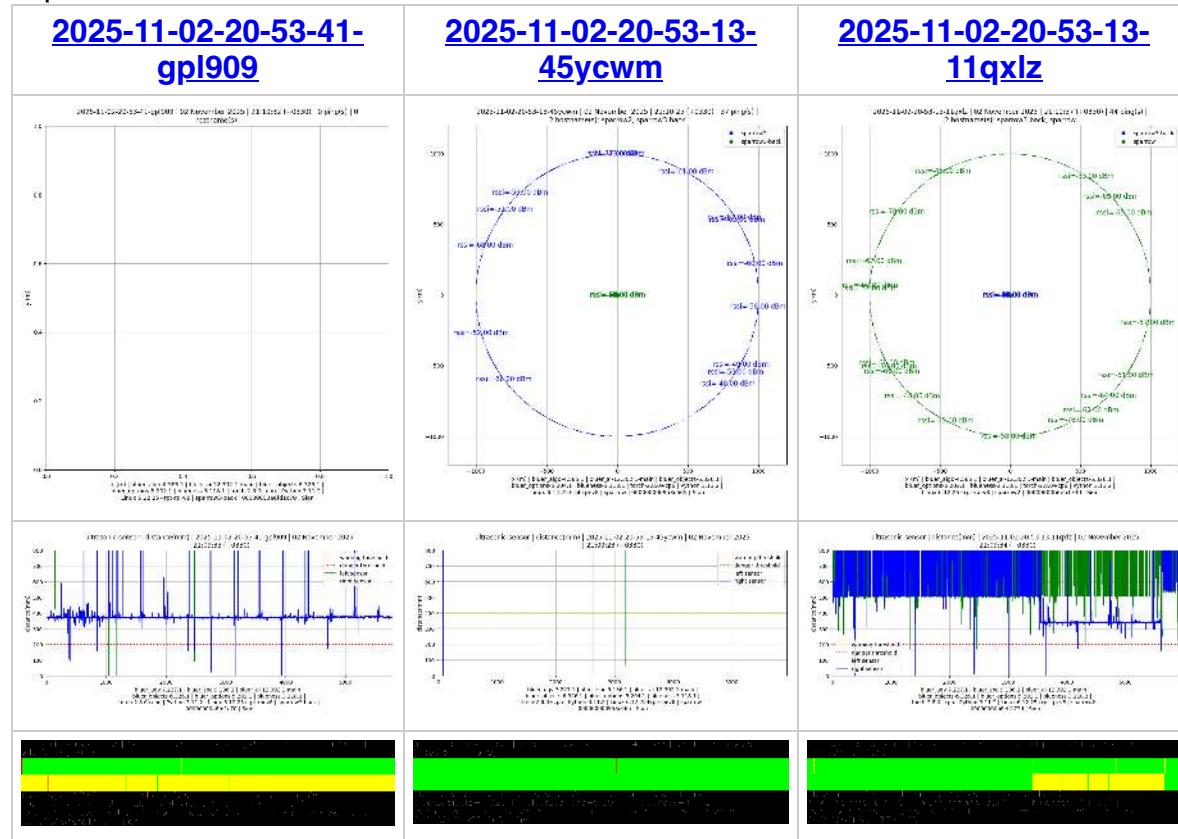
on the anchor run,

```
@bps set_anchor 0,0,0,1
```

on all run,

```
@select; @session start
```

► publication



bps: simulations

- [timing](#)

bps: simulations: timing

continues [v1](#).

3 nodes

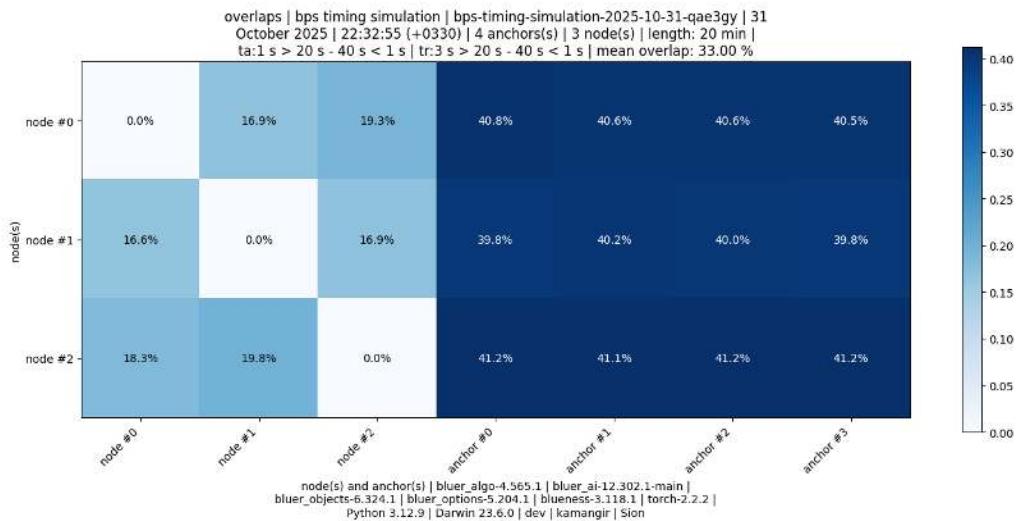
```
@select bps-timing-simulation-$(@@timestamp)
```

```
@bps simulate timing upload .
```

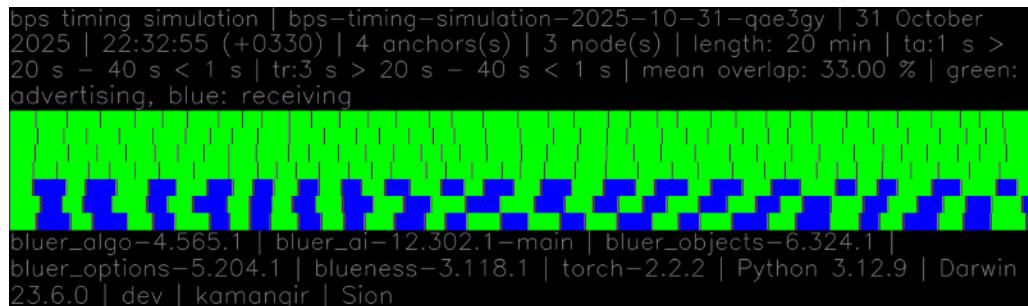
```
@assets publish extensions=png,push .
```

mean overlap: 0.33

► metadata



image



image

12 nodes

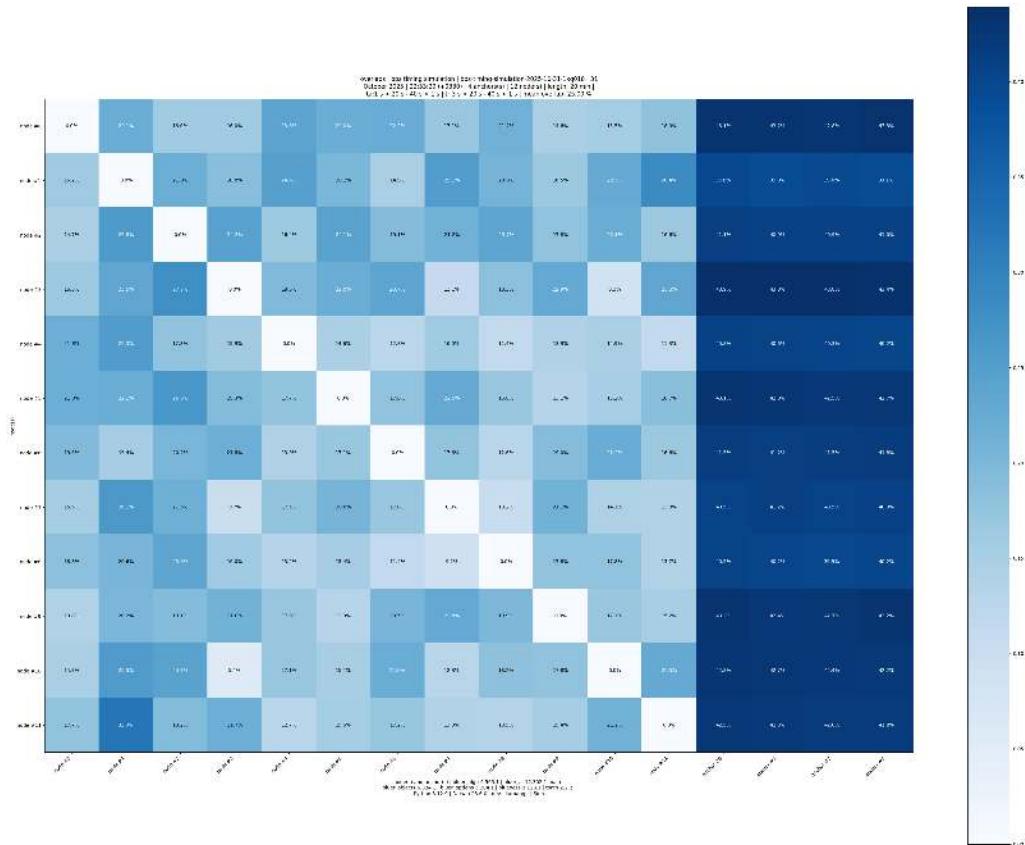
```
@select bps-timing-simulation-$(@@timestamp)
```

```
@bps simulate timing upload . \
--nodes 12
```

```
@assets publish extensions=png,push .
```

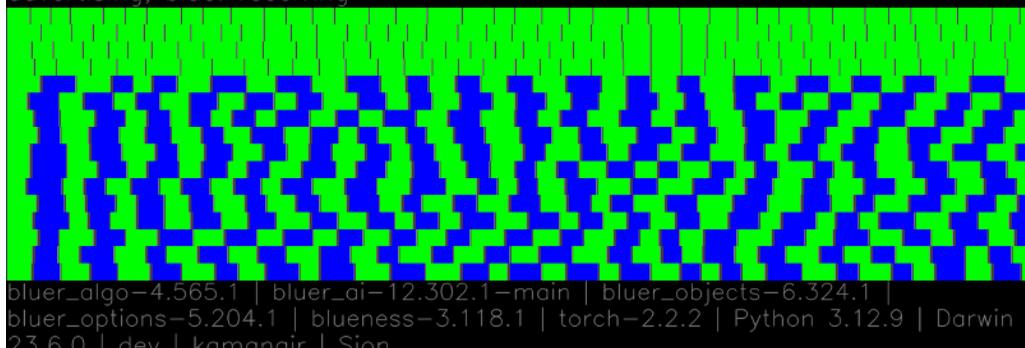
mean overlap: 0.25

► metadata



image

bps timing simulation | bps-timing-simulation-2025-10-31-1kq018 | 31 October 2025 | 22:33:29 (+0330) | 4 anchor(s) | 12 node(s) | length: 20 min | ta:1 s > 20 s - 40 s < 1 s | tr:3 s > 20 s - 40 s < 1 s | mean overlap: 25.00 % | green: advertising, blue: receiving



image