

**FIRE
DETECTION**



Forest FireFighter F³

1st sprint review

Zilly team members :

Axel JORGE

Corentin TATGER

Liao ZHANG

Youssef AMARI

Ruiqi HU

Amelie MAIER

TABLE OF CONTENTS

01 PLANNED/REALIZED STORIES
AND ACCEPTANCE TESTS

02 DEMONSTRATIONS

03 SCHEDULE CONTROL

04 NEXT SPRINT ?



01

PLANNED/REALIZED STORIES AND ACCEPTANCE TESTS

Caption:

TODO

...

Peanut

Simple

Medium

Hard

Hardcore/
impossible

Simple
Website

Send/get
data

Motor
command

Direction
management

Max position
accuracy

GPS
localization

Get
image

Image/flow test
in simulation

01

PLANNED/REALIZED STORIES AND ACCEPTANCE TESTS

Caption:					
Story points:		TODO	IN PROGRESS	TESTING	DONE
Peanut	3				Simple Website
Simple	5		Motor command	Image/flow test in simulation	Send/get data
Medium	8		Direction management	GPS localization	Max position accuracy
Hard	13				Get image
Hardcore/impossible	20				

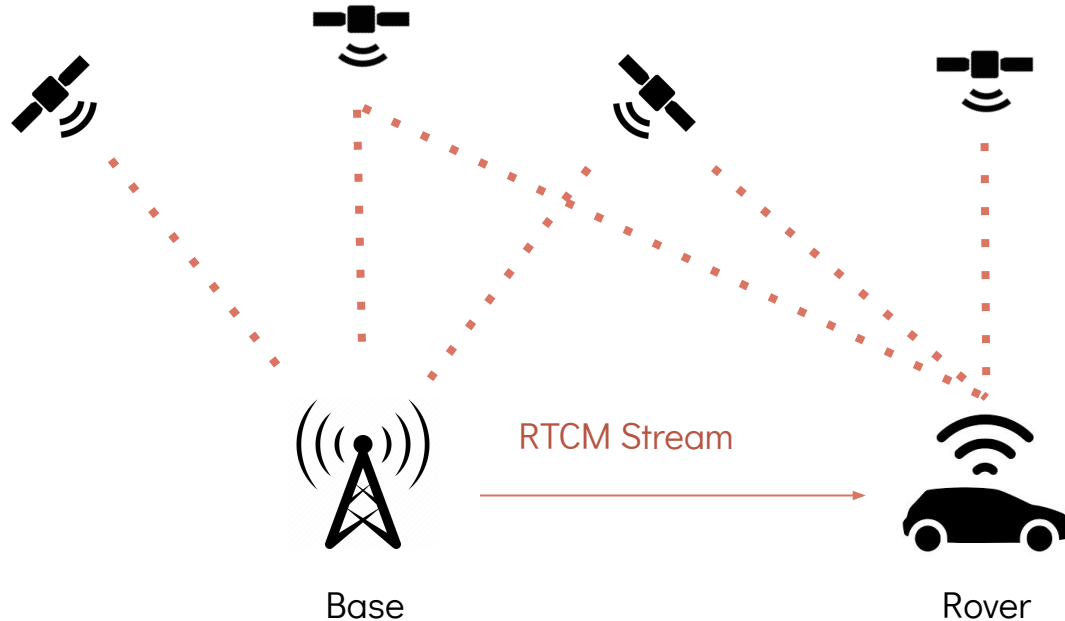
02 DEMONSTRATIONS



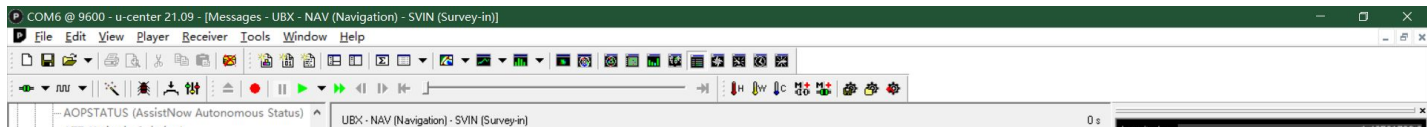
02

GNSS Module - RTK System

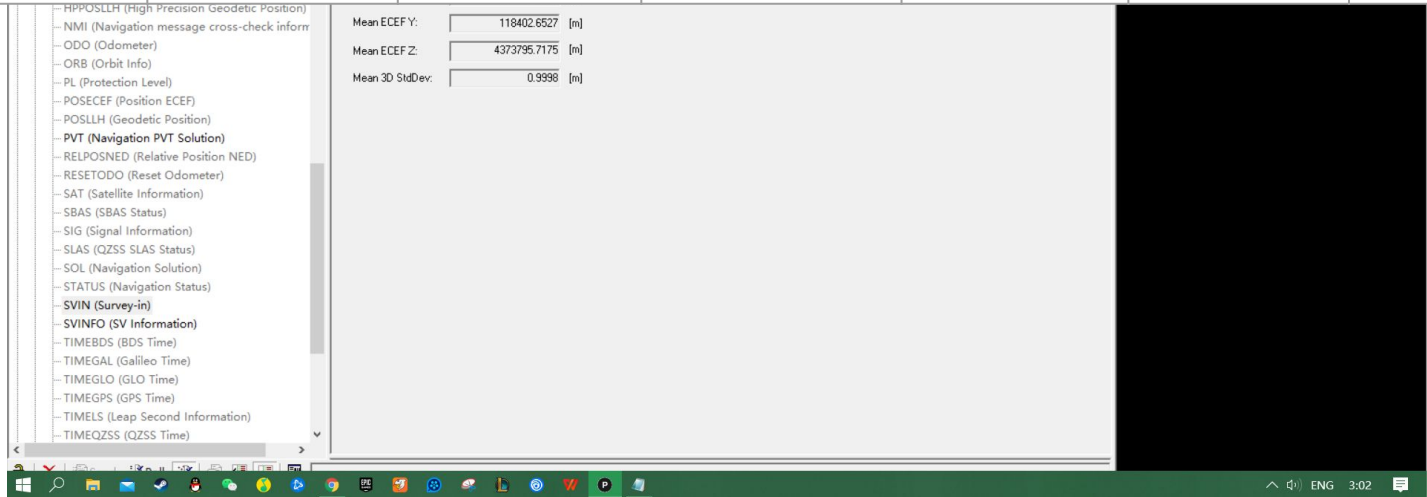
Operational scheme



02 Standard RTK



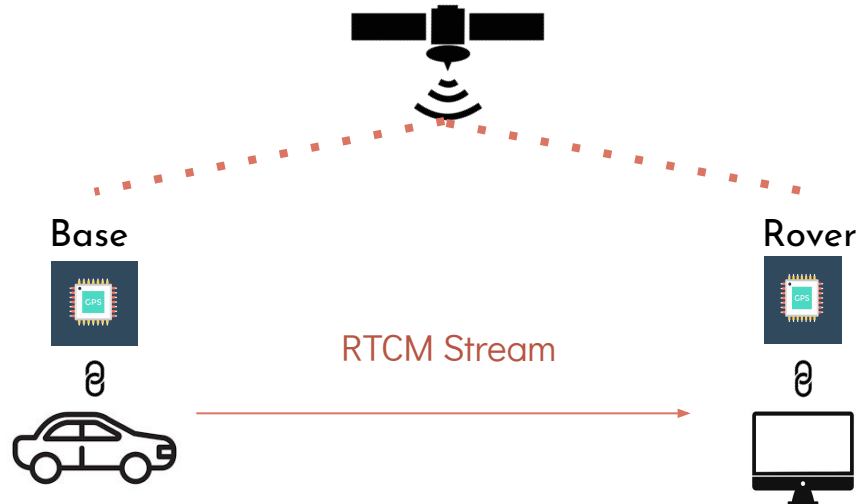
Accuracy(m)	10	5	4	3	2	1
Time cost(s)	10	28	39	63	137	802



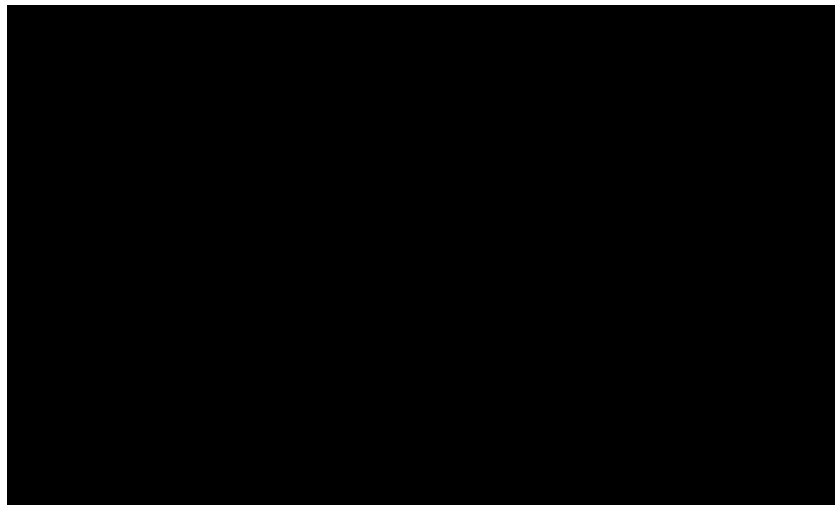
02

GNSS Module - Moving baseline RTK

Operational scheme Modified



02 Moving baseline RTK



02

COMPUTER VISION



User Story 1: Data from the camera must be retrieved and saved



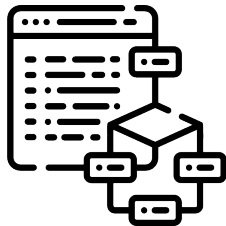
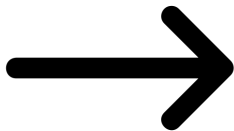
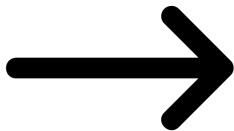
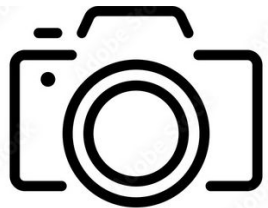
Test: Start the camera, take a video, and save it. The file must be possible to open



User Story 2: Images must be processed to detect a fire outbreak



Test: Give an image file to the algorithm and check the answer. Done if accuracy >50%



02

COMPUTER VISION

OpenCV

IMPLEMENTATION

- Color analysis
- Shape detection not possible

Machine learning

IMPLEMENTATION

- Train/Test Dataset
- Automatic color/shape analysis



OpenCV



python™

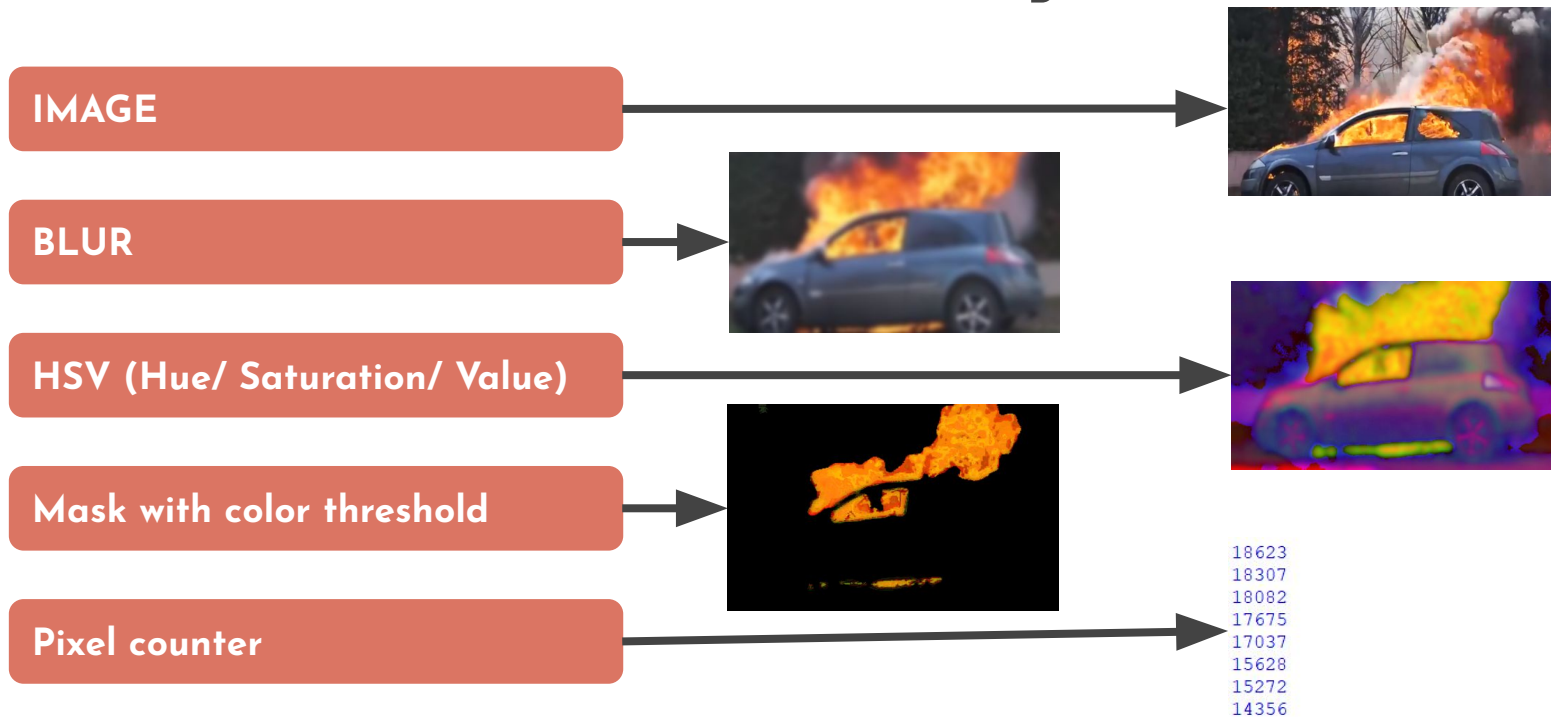


PyTorch

02

COMPUTER VISION

Without machine learning



02

COMPUTER VISION

Limitations of color analysis

Color threshold

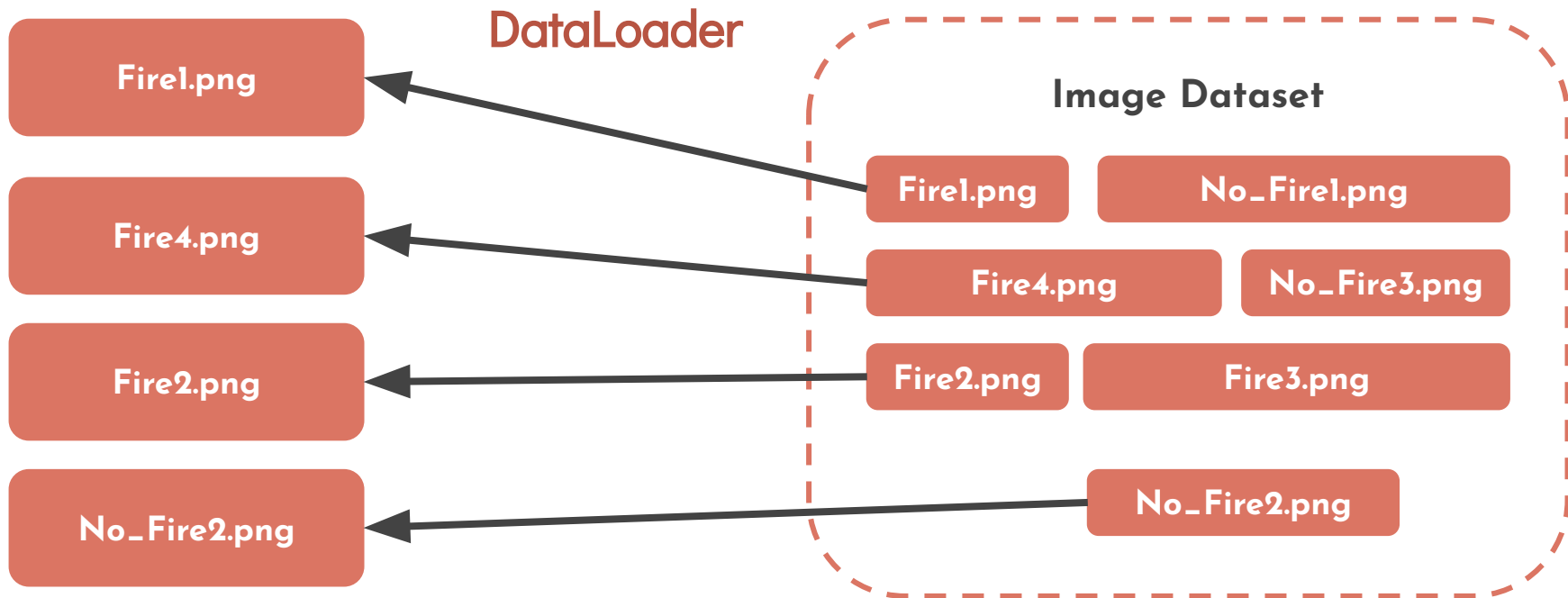
lower = [18,50,50]

upper = [50,255,255]



02

COMPUTER VISION



02

COMPUTER VISION

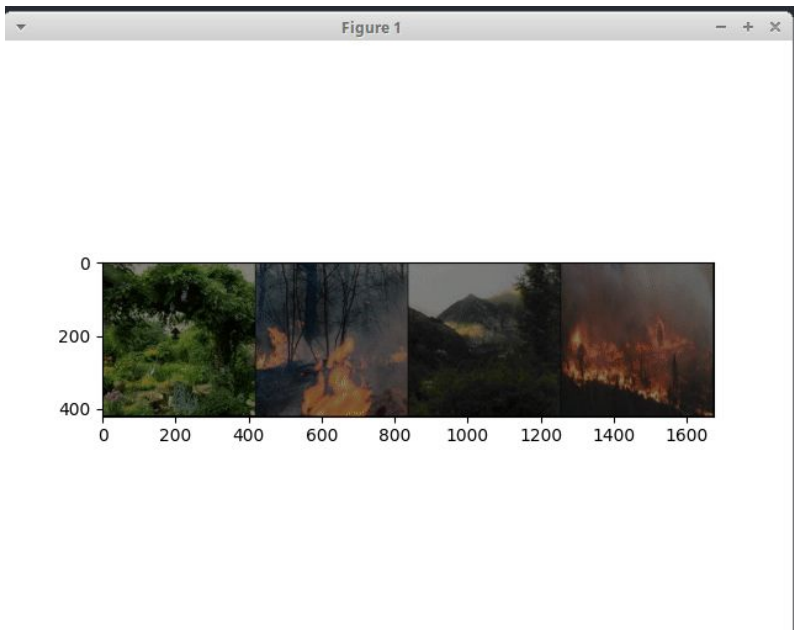


Image Dataset

Fire1.png

No_Fire1.png

Fire4.png

No_Fire3.png

Fire2.png

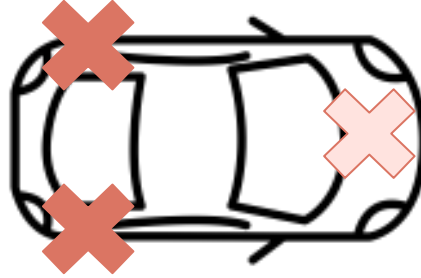
Fire3.png

No_Fire2.png

02 MOVEMENT AND DIRECTION

How it works ?

Two motors at the back
→ **POWER**



One servomotor at the front
→ **DIRECTION**

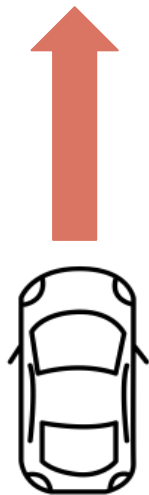
- One motor on each side
 - Allows differential command

- Command between 0 and 100
 - Turn left < 50
 - Turn right > 50
 - Go straight = 50

02 MOVEMENT AND DIRECTION

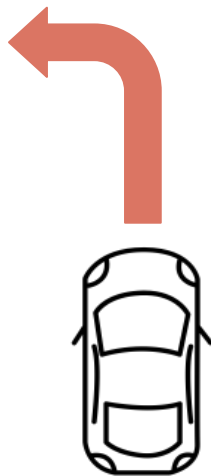
Tests

1st test : Go straight



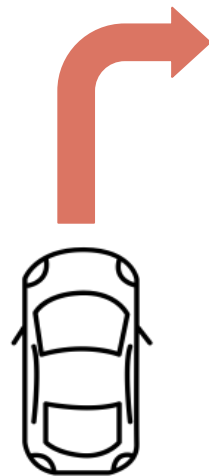
Front
command = 50

2nd test : Turn left



Front
command = ??

3rd test : Turn right



Front
command = ??

02

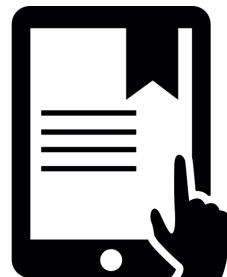
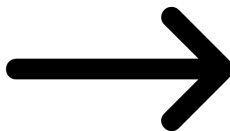
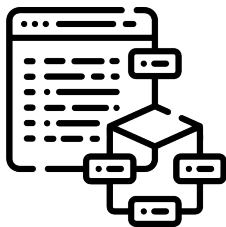
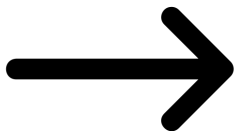
ISSUES ENCOUNTERED

- Download problem of Raspberry Pi
- Problem of CAN communication
- Problem when we flash the code to Nucleo card
- Simulation with Proteus :
 - Model mismatch
 - Fee for Matching model

02

WEBSITE AND DATA

- ▶ **User Story 1:** Create a simple website
- ✓ **Test:** Connection to local address and right port of communication and see the web page
- ▶ **User Story 2:** Send data to the website
- ✓ **Test:** Creation of a client sending data (JSON format) to the server. The data is printed on the website and stored on a local file.



02

WEBSITE AND DATA

Flask

IMPLEMENTATION

- Creation of a website
- Parsing of a file
- Request/Response mechanism



Flask



02

WEBSITE AND DATA

```
(venv) youssef@Youssef:~/Desktop/flask_app$ python client.py
{'latitude': '65', 'longitude': '121'}
(venv) youssef@Youssef:~/Desktop/flask_app$
```

```
(venv) youssef@Youssef:~/Desktop/flask_app$ python server.py
* Serving Flask app 'server' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 310-983-120
Value on the server {'longitude': '121', 'latitude': '65'}
127.0.0.1 - - [09/Nov/2021 02:37:36] "POST /pi HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2021 02:37:44] "GET /pi HTTP/1.1" 302 -
127.0.0.1 - - [09/Nov/2021 02:37:44] "GET /%22latitude%22:%20%2265%22%2C%20%22longi
tude%22:%20%22121%22 HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2021 02:37:48] "GET /favicon.ico HTTP/1.1" 200 -
```

02

WEBSITE AND DATA

```
{ "latitude": "20", "longitude": "10" } { "latitude": "99", "longitude": "45" } { "latitude": "99", "longitude": "45" } { "latitude": "99", "longitude": "45" } { "latitude": "99", "longitude": "45" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "60", "longitude": "52" } { "latitude": "200", "longitude": "100" } { "latitude": "200", "longitude": "100" } { "latitude": "200", "longitude": "100" } { "latitude": "54", "longitude": "88" } { "latitude": "54", "longitude": "88" } { "latitude": "65", "longitude": "121" } { "latitude": "65", "longitude": "121" }
```

02

WEBSITE AND DATA

Welcome to Zilly team Website!

These are the current coordinates of the car in degrees (°): "latitude": "65", "longitude": "121"

Refresh coordinates

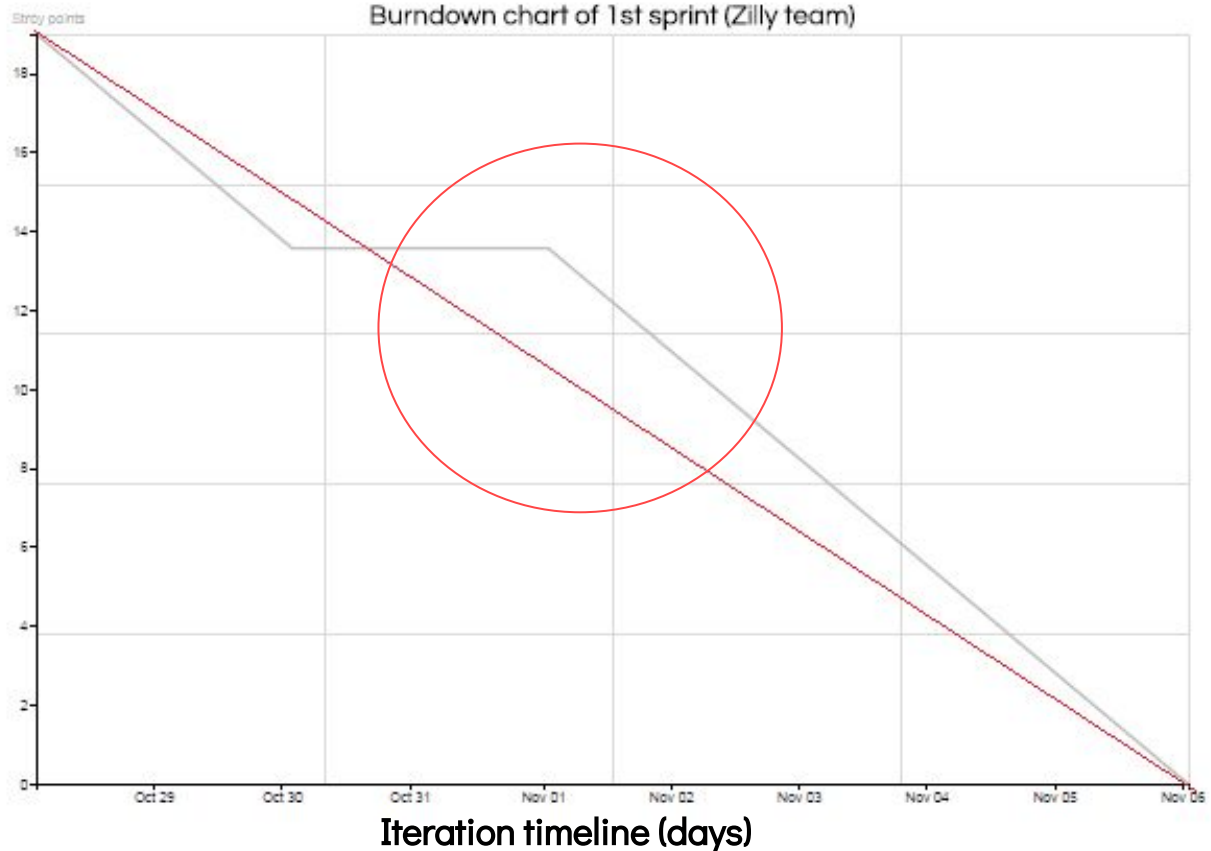
03

SCHEDULE CONTROL

Caption:

- Ideal tasks remaining
- Actual tasks remaining

Story points



04 NEXT SPRINT ?

SPRINT 2

- **Feature:** Movement according to GPS
-- **Test:** Move in a straight line between 2 GPS points without any obstacles
- **Feature:** Real-time image processing
-- **Test:** Process images as they are taken with camera
- **Feature:** Transmission of an image to the website
-- **Test:** Display an image file
- Improvement of the current website

07 CONCLUSION

*Thank you for your
attention*