Sprint 2 Review

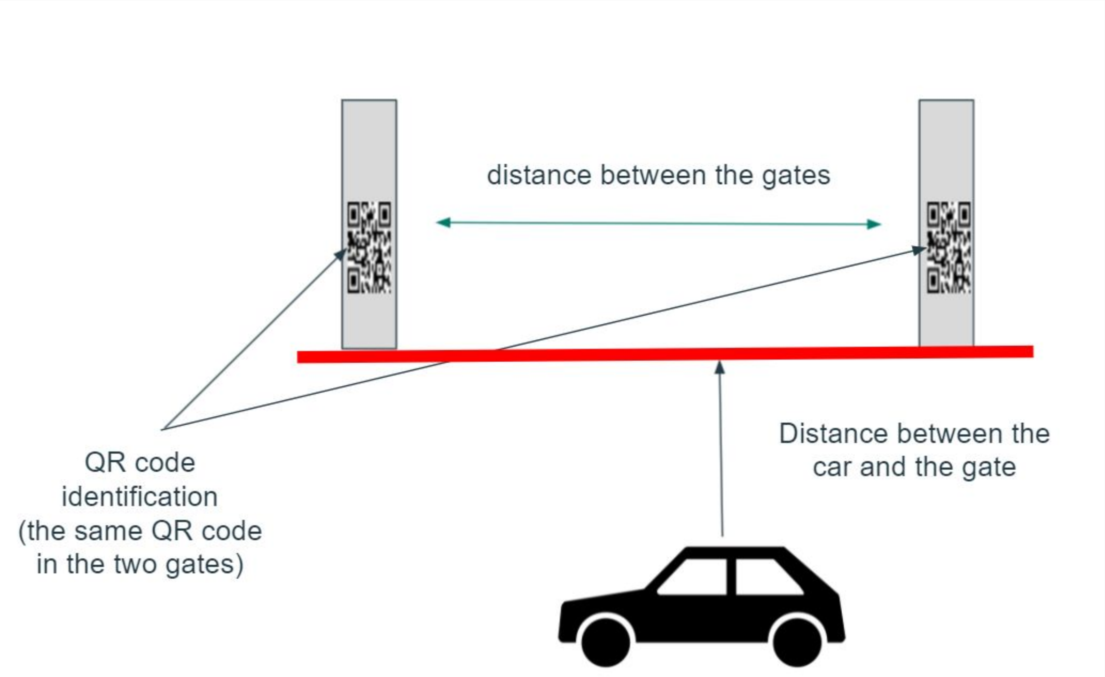
BURTON Nidishlall,
CHOUIYA Asmae,
EL HACHIMI Asmae,
MARTY Axel,
PIQUES Nicolas,
RAMIARA Maxime

Yankee Doodle Pigeon

INSA TOULOUSE

Contents

Reminder of the objectives of the project



Sprint 2

Reminder of the sprint 2 's objectives

From sprint 1:

Implement the QR code detection into the JETSON

Communicate between the Rasberry and the Nucleo to have a manual control

From sprint 2:

Moving the car forward on a simple trajectory

Gate identification using camera

Gate detection using a Lidar

Project organization

- Gate detection and QR code position detection:
    - Asmae El Hachimi
    - Maxime Ramiara

- Moving the car in a simple trajectory:
    - Asma Chouiya
    - Axel Marty

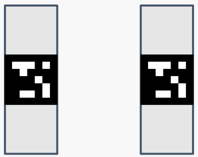
- LIDAR:
    - Nicolas Piques
    - Nidishlall Burton

- Bibliographical research :
    - Nidishlall Burton

QR Code detection

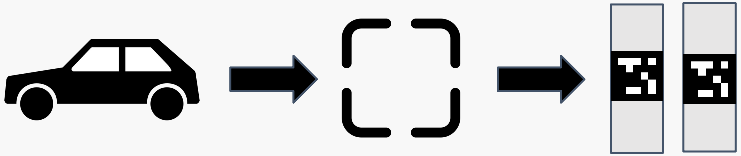
1) **Context and objectives**
2) Gate identification
3) QR codes position
4) Tests
5) Demonstration

Context:



- Gates marked with **QR codes**
- Tell if it’s the **right way or not**
- Choice of a “correct” QR Code -> pass through the gate associated
- Get the **position** of the QR codes.

Objectives:



- Detection time **< 500 ms** for a distance **< 2 m**
- Recognize **each gate’s ID** with 0.01% accuracy
- Get a point in the **center of the gate.**

QR Code detection

1) Context and objectives
2) **Gate identification**
3) **QR codes position**
4) Tests
5) Demonstration

__Tools used :__ **OpenCV** and **Aruco** libraries

Gate identification:

- Identification of **two identical QR codes**
- Each **pair of QR codes** represents a gate
- Compare the ID of the chosen product, if it’s the right ID, print that **it’s the right gate.**

QR codes position:

- Tried 2 methods: color detection and aruco library
- Calculate the **center of each aruco marker**
- Sending back the **position** of the centers
- Calculate **the position of the center** point **between** the aruco codes

QR Code detection

1) Context and objectives
2) Camera calibration
3) Tracking
4) **Tests**
5) **Demonstration**

Demonstration:

- **Camera** connected by USB to the jetson
- **Choice** of a gate
- **Tracking** and differentiation
- **Detection** of the gate and **identifying** it
- Draw a **rectangle** between the gates and draw the **center** of the gate

Acceptance tests:

- <u>Test 1 :</u> Distance of **2 m from the QR code**
- <u>Test 2 :</u> Delay of detection **less than 500 ms**

Gate detection using a Lidar

Context:

- Lidar informs us about the distance between the car and a gate
- Lidar + camera => path planner more precise

Objectives:

- See what are the data sent by the LIDAR (LaserScan Message)
- Collect only the interesting data (Distance and angle associated)
- Start the path planner program

Gate detection using a Lidar

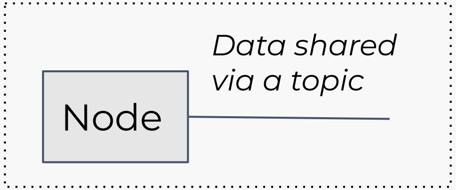
1) Context and objectives
2) **The ROS computation graph of our system**
3) The LIDAR application
4) Tests
5) Demonstration of the feature

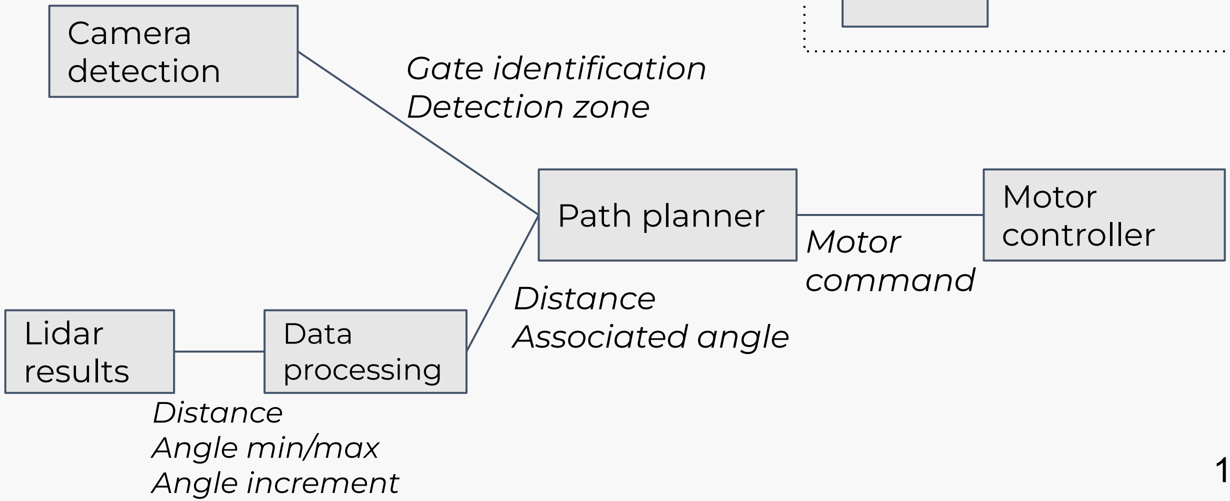
The ROS Software advantage:

- Separate the code into programs called nodes
- Share data between nodes using topics
- Publishing and subscribing to a topic

=> Easy way to share data between our programs

ROS computation graph:





Gate detection using a Lidar

LIDAR Accuracy

4 Tests were made :

1. Normal,
2. Changing the starting position of the Lidar by : 10°,
3. Same as 2 with 45°
4. By resetting the Lidar

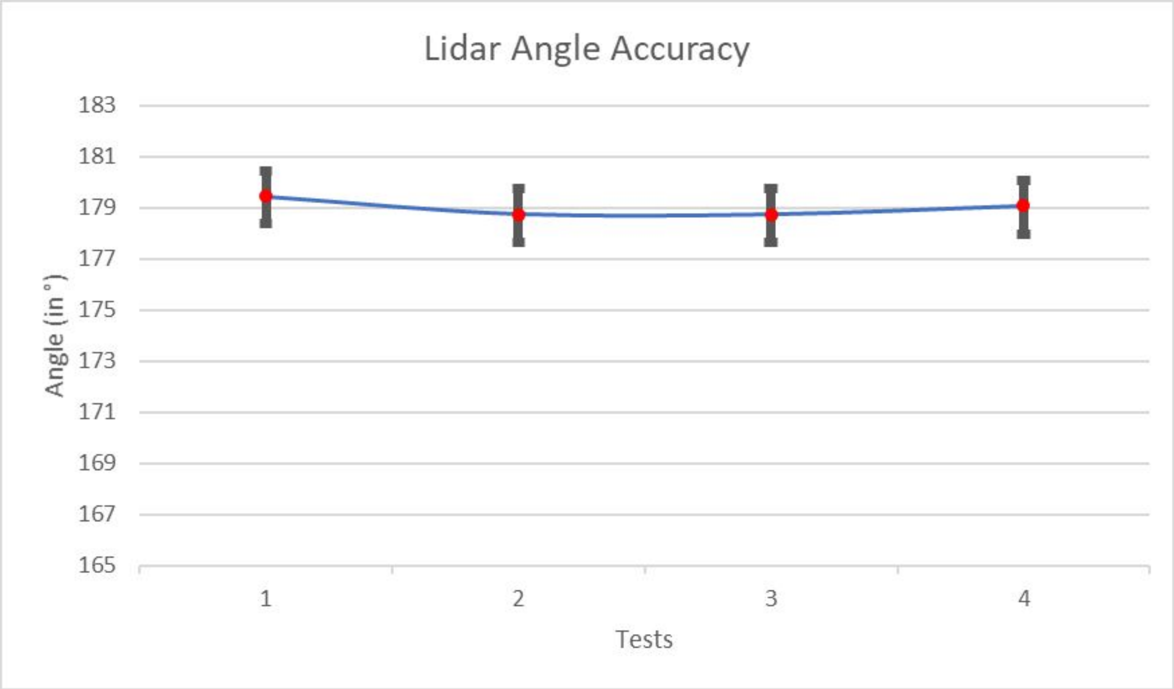
LIDAR => distance range 0.15 m to 6.5 m

Accuracy Test :
- Covering up most of the LIDAR
- One slight opening at the same place for all 4 tests

Gate detection using a Lidar

1) Context and objectives
2) The ROS computation graph of our system
3) **The LIDAR application**
4) Tests
5) Demonstration of the feature

LIDAR Accuracy



Gate detection using a Lidar

1) Context and objectives
2) The ROS computation graph of our system
3) The LIDAR application
4) **Tests**
5) **Demonstration of the feature**

Demonstration:

- 2D representation of a gate using RVIZ
- Data representation using ROS

Acceptance tests:

- Identification of a gate:
  - Delay < 500 ms

Car motion

1) **Context and objectives**
2) following a predefined trajectory
3) Tests
4) Demonstration

Context:

- control the car in emergency cases
- move the car in different directions

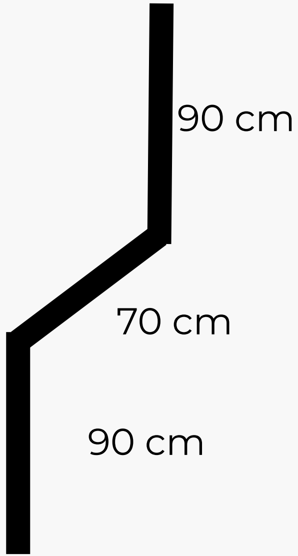
Objectives:

- follow a predefined trajectory
- control the car via the GUI

Car motion

Trajectory description

	Distance (centimeter)	Time (second)
Forward	90	5
Turn right	60	2
Forward	90	5
Turn left	70	2
Forward	90	5



Car motion

1) Context and objectives
2) following a predefined trajectory
3) **Tests**
4) **Demonstration**

Demonstration:

- facing a problem with the can bus configuration

  ==> no control via the GUI

- moving according to the predefined trajectory

Sprint results

Sprint successes:

- Implement the QR code detection into the JETSON, detect a gate and get it's center
- Gate detection using a LIDAR
- Moving the car in a simple trajectory

Improvement for next sprint:

- Communication between the Rasberry and the Nucleo to establish the manual control

Sprint 3

Sprint 3

1) **Sprint 3 's objectives**
2) **The planned tasks**
3) Acceptance tests
4) Demonstrations planification

Objectives:

- **Priority 1 :** Theory about the calculation of the path to a gate at any location
- **Priority 2 :** Communication between the Rasberry and the Nucleo: Manual and remote control of the car
- **Priority 3 :** Solve ROS publishing issues

Tasks:

- Theory about the calculation of the path (Nidishlall Burton, Maxime Ramiara, Asmae El Hachimi)
- Communication between the Rasberry and the Nucleo (Axel Marty, Asma Chouiya)
- Solve ROS publishing issues (Nicolas Piques)

Sprint 3

1) Sprint 3 ’s objectives
2) The planned tasks
3) **Acceptance tests**
4) Demonstrations planification

- Calculation of the trajectory to follow -> simulation:
  - Implement and test by simulation the Trajectory algorithm
    - PID corrector
- Communication between the Rasberry and the Nucleo: Manual and remote control of the car:
  - Range up to 10m
  - Command response time less than 1s
- Manual control using a GUI
  - Command response time < 1 second

Sprint 3

1) Sprint 3 ’s objectives
2) The planned tasks
3) Acceptance tests
4) **Demonstrations planification**

- Control the car from a distance with graphical interface
- Gate detection using ROS, Lidar and Camera

Thanks !

Any Questions?