Sprint 4 Review

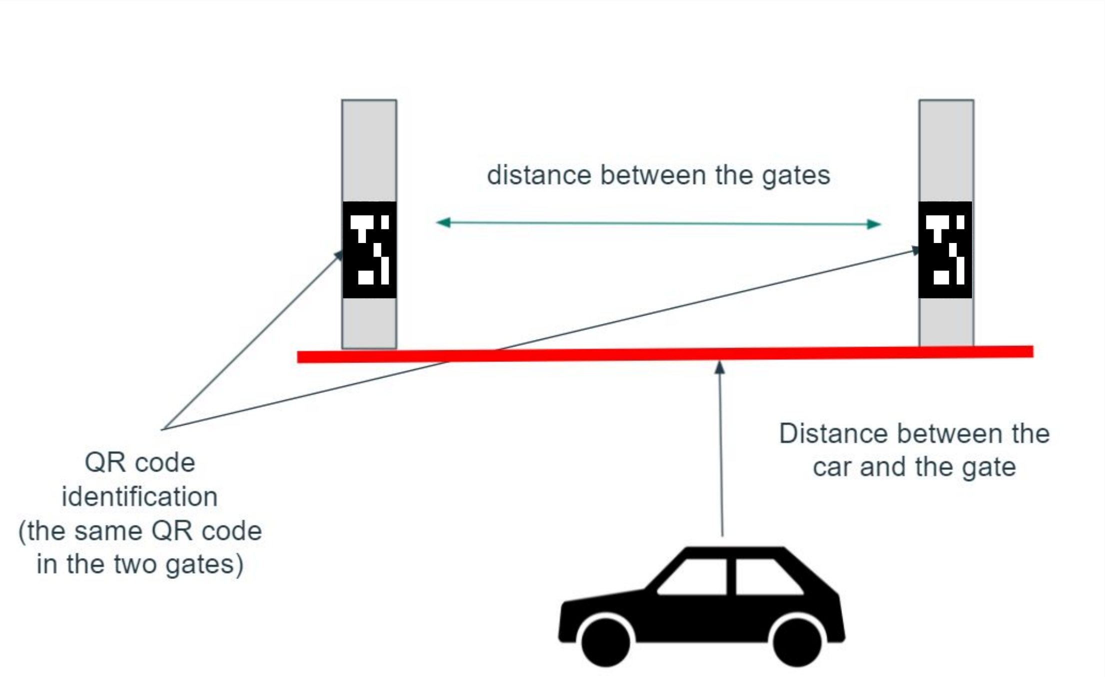
Yankee Doodle Pigeon

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

INSA TOULOUSE

Contents

Reminder of the objectives of the project



Sprint 3 Recap

<u>Tools used :</u> OpenCV and **Aruco** libraries

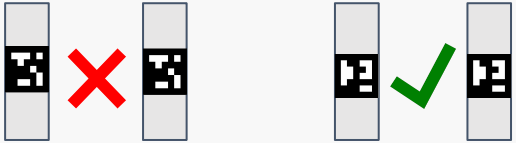
Camera: Gate detection

1) **Gate identification**
2) QR codes position

Gate identification :

- Detection of **two identical QR codes**
- Each **pair of QR codes** represents a gate
- Identify the gate with **the right ID**

“I want to go to warehouse ID 10”



Wrong gate : ID 5

Right gate : ID 10

Camera: Gate detection

1) Gate identification

2) QR codes position

<u>Tools used :</u> OpenCV and **Aruco** libraries

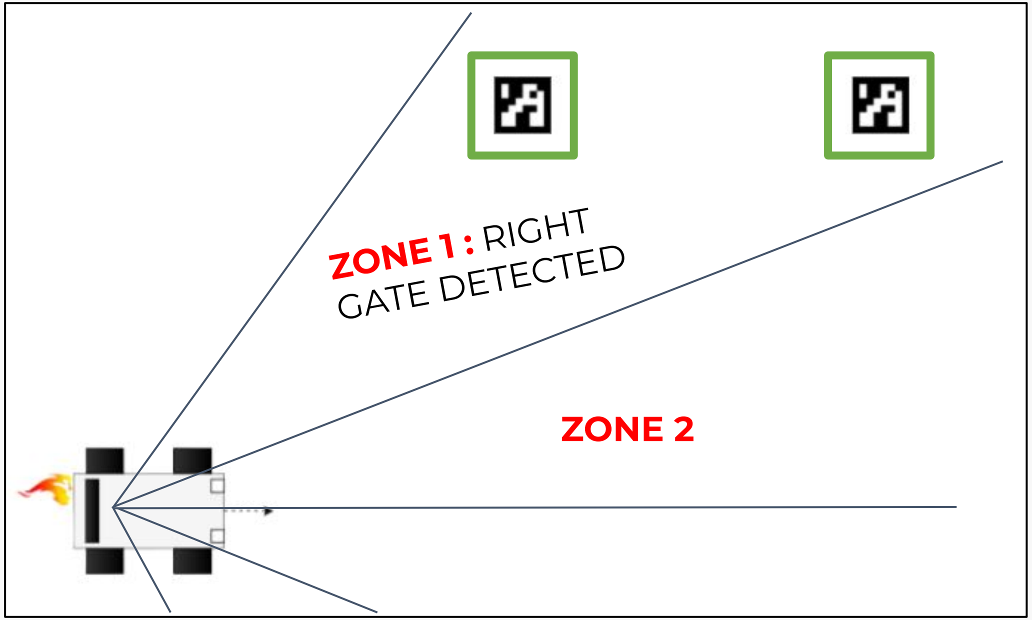
QR codes position :

- 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 -> path calculation

Theory of trajectory calculation

1) **Step 1 : QR detection using the camera**
2) Step 2 : Angles control using the Lidar
3) Step 3 : Gate calibration using the camera
4) Step 4 : Gate crossed acknowledgement using the Lidar

Step 1 : QR Code detection using the camera

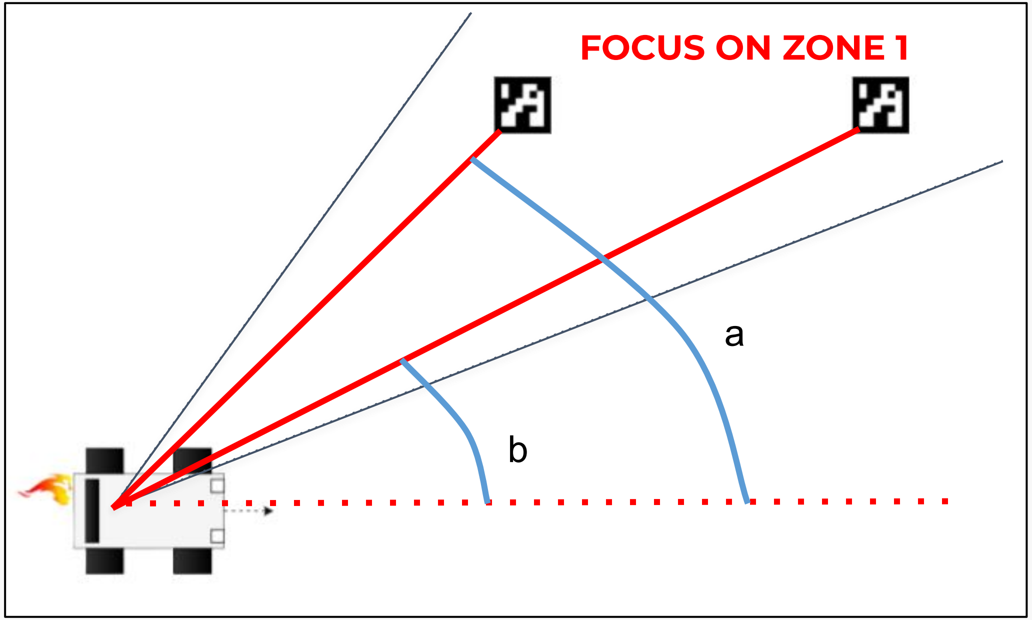


=> The zone of detection will be used for the next step

Theory of trajectory calculation

1) Step 1 : QR detection using the camera
2) **Step 2 : Angles control using the Lidar**
3) Step 3 : Gate calibration using the camera
4) Step 4 : Gate crossed acknowledgement using the Lidar

Step 2 : Angles control using the Lidar



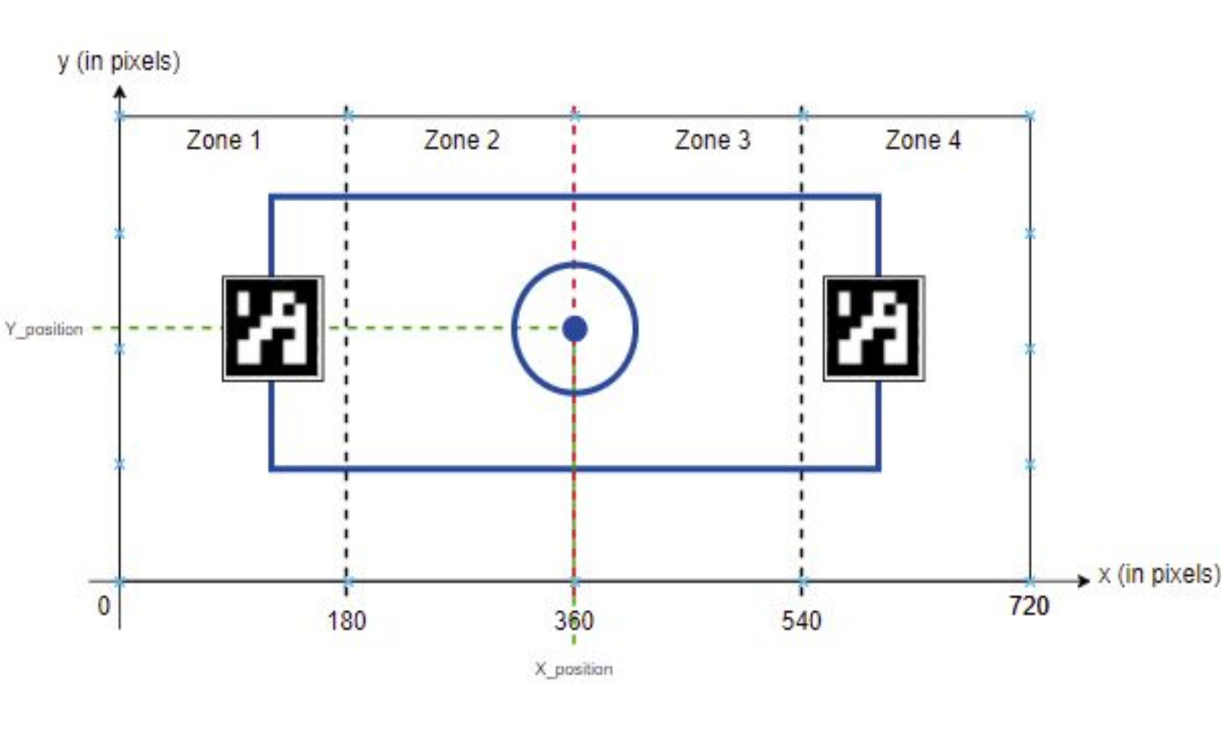
=> angles control : |a-b| ≤ 10° (ideal)

- use of a PID controller

Theory of trajectory calculation

1) Step 1 : QR detection using the camera
2) Step 2 : Angles control using the Lidar
3) **Step 3 : Gate calibration using the camera**
4) Step 4 : Gate crossed acknowledgement using the Lidar

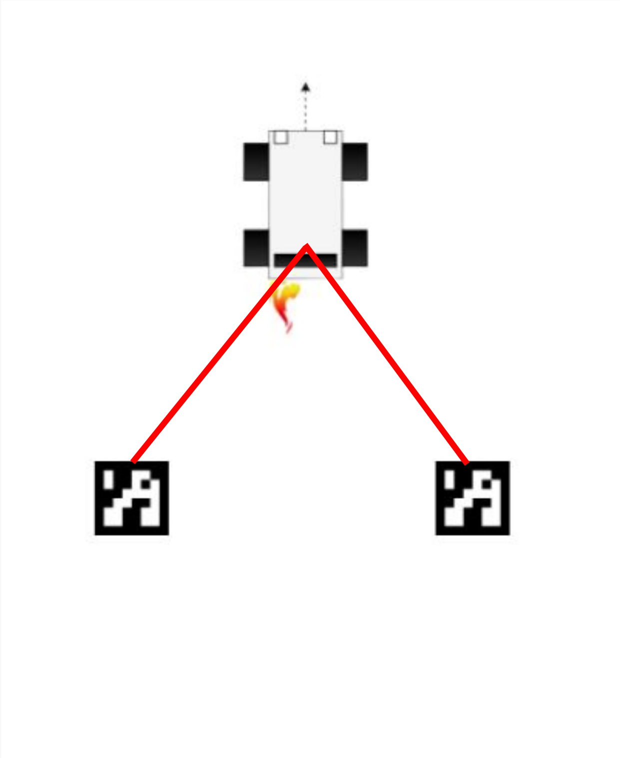
Step 3 : Gate calibration using the camera



Theory of trajectory calculation

1) Step 1 : QR detection using the camera
2) Step 2 : Angles control using the Lidar
3) Step 3 : Gate calibration using the camera
4) **Step 4 : Gate crossed acknowledgement using the Lidar**

Step 4 : Gate crossed acknowledgement using the Lidar



Sprint 4

Reminder of the sprint 4 's objectives

- The calculation of a simple path to a gate
- Set a priorities management between the autonomous state and the manual state
- Communication between the Jetson and the Raspberry

Project organization

- Gate position and distance detection using camera and communication between the Jetson and the Raspberry:
    - Asmae El Hachimi

- The autonomous state/manual state:
    - Asma Chouiya
    - Axel Marty

- Theory about the calculation of the path to a gate :
    - Nidishlall Burton
    - Maxime Ramiara
    - Nicolas Piques

- Tracking :
    - Nicolas Piques

Camera: QR position compared to the frame center

1) **Position of the car**

2) Demonstration and acceptance tests

3) PID Controller on the position’s error

Position of the frame center compared to the center of the gate :

- Draw a **vertical** line in the **center** of **frame** of the camera
- Compare the **position** of the center of the gate to the center of the frame
- Calculate **the distance between** the two centers.
- Calculate the **error**

Camera: QR position compared to the frame center

1) Position of the car

2) Demonstration and acceptance tests

3) PID Controller on the position's error

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
- Draw a **vertical line** in the center of the fram.
- Display the **distance** and the **error**

Acceptance tests :

- <u>Test 1:</u> show instantly the **vertical line** of the center of the frame
- <u>Test 2 :</u> Delay of the distance and error calculation **less than 500ms**

Camera: QR position compared to the frame center

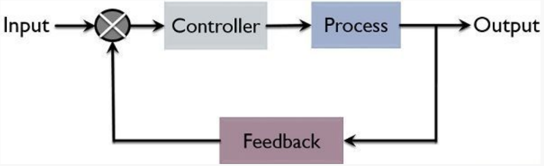
1) Position of the car

2) Demonstration and acceptance tests

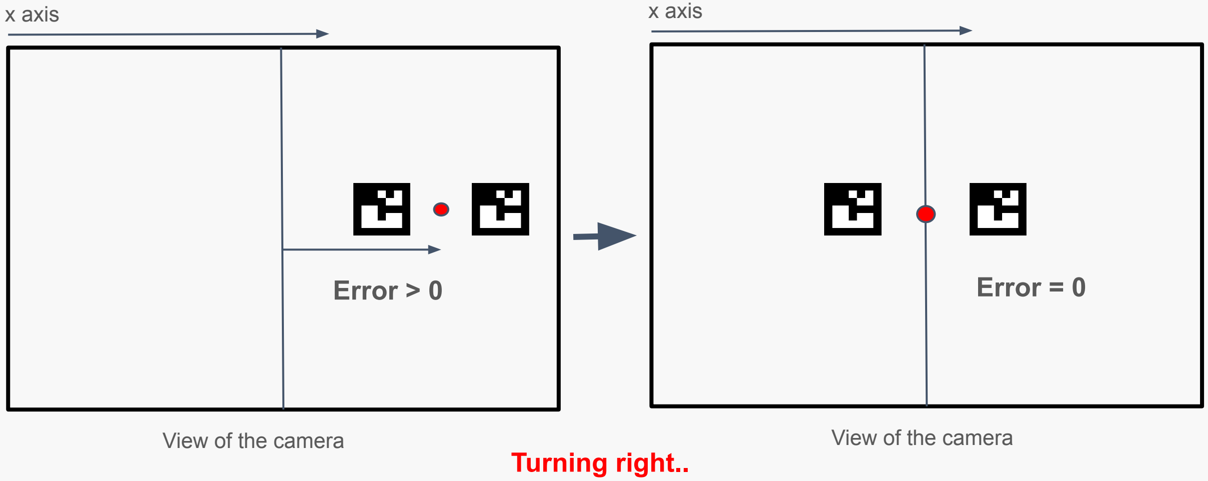
3) **PID Controller on the position’s error**

PID Controller on the position’s error:

- **Error** = Frame center - gate center
- **Objective :** Align the frame center and the gate center -> **PID Control loop**
- Only a **proportional** controller for now



Control processing:



Remote control mode

1) **Context and objectives**
2) GUI
3) Manual control
4) Tests
5) Demonstration

Context :

- control the car in emergency cases

Objectives :

- control the car via the GUI

Remote control mode

1) Context and objectives

2) GUI

3) Manual control

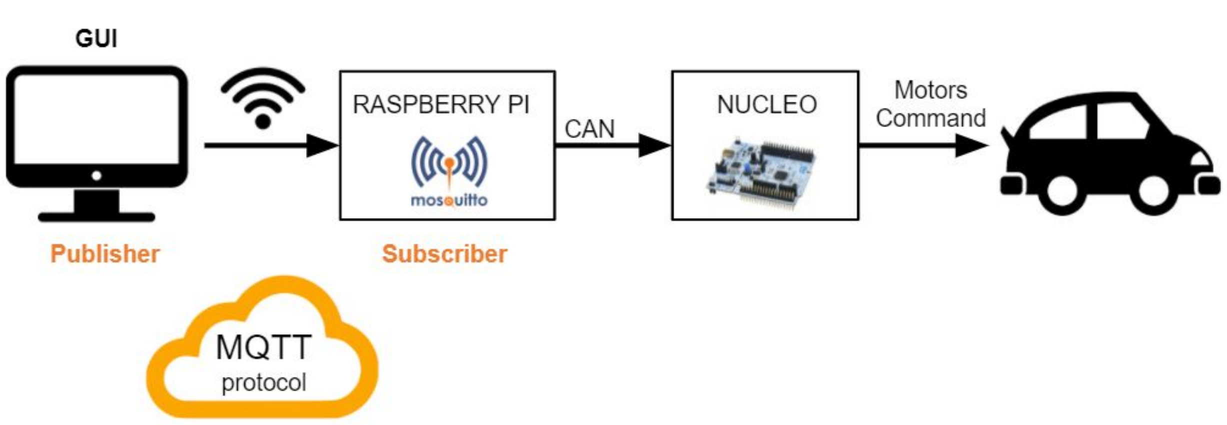
4) Tests

5) Demonstration

Graphical User Interface

Remote control mode

Communication between GUI and the car



Remote control mode

1) Context and objectives
2) GUI
3) Manual control
4) **Tests**
5) **Demonstration**

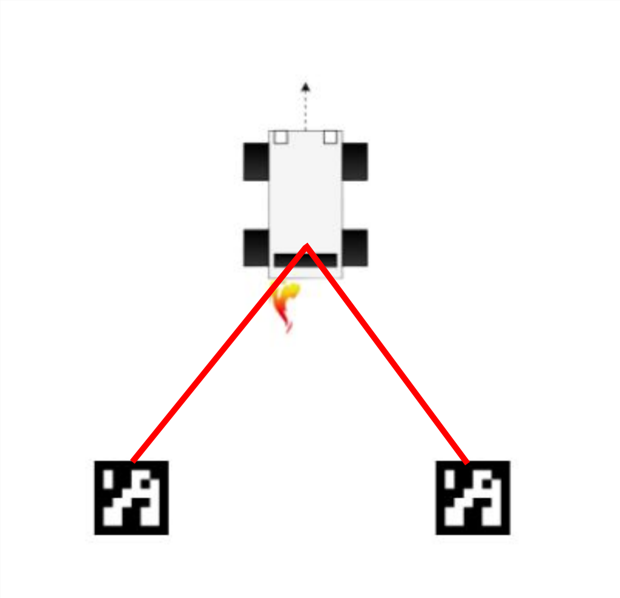
Acceptance tests :

- Test 1 : manual control with a range of **0 to 25 m**
- Test 2 : response time **< 1 s**

Tracking

1) **Objective**
2) Tracking the gate
3) Detecting the gate position
4) Tests
5) Demonstration

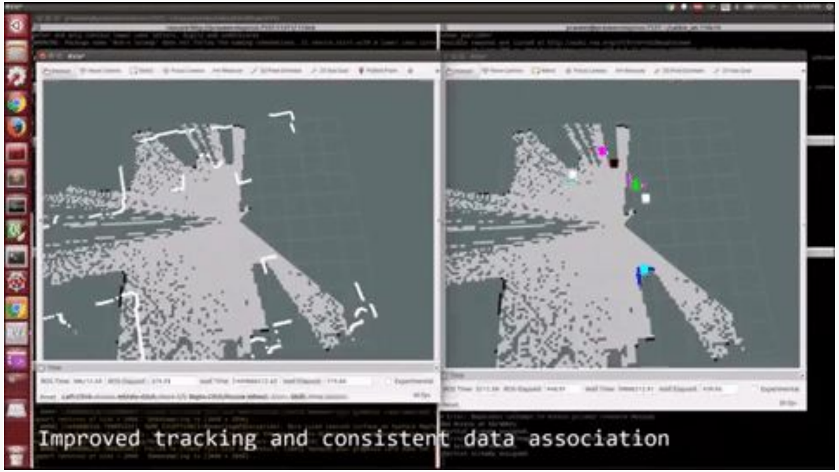
Objective : Stop the vehicle after crossing the gate



Step 4 : Gate crossed acknowledgement using the Lidar

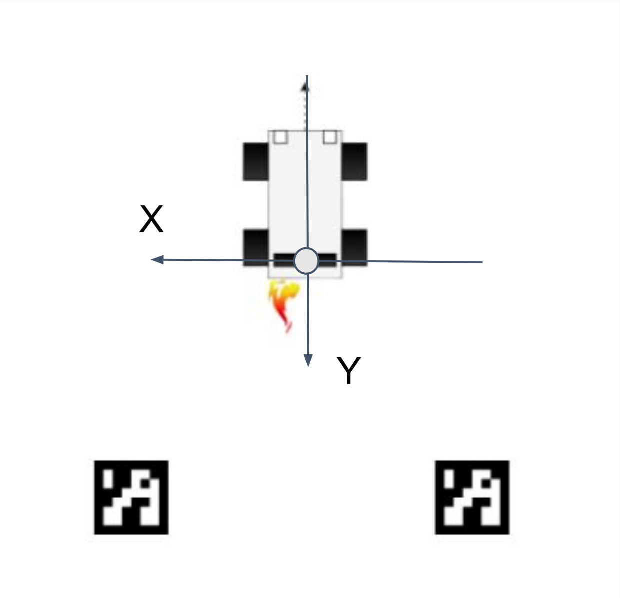
Tracking

Tracking the gate using the Lidar



- Know where the gate is using the camera
- Follow the gate using the Lidar
- Stop the vehicle when the gate is behind

Detecting the gate position



- Each object has a XY position
- The distance can be found (sqrt(X^2+Y^2))
- Stop the vehicle when : Y gate position > 0

Tracking

1) Objective
2) Tracking the gate
3) Detecting the gate position
4) **Tests**
5) **Demonstration**

Acceptance tests :

- Test 1 : Following an object at a distance **up to 2 m** from the Lidar

Sprint results

Sprint successes :

- Give direction to the car knowing the position of the gate
- Management in progress between manual and autonomous control mode
- Tracking of the gate

Sprint 5

Sprint 5

1) **Sprint 5 's objectives**
2) **The planned tasks**
3) Acceptance tests
4) Demonstrations planification

Objectives :

- **Priority 1 :** The calculation of the path to a gate at any location
- **Priority 2 :** Connect all the cards and exchange frames

Tasks :

- Merging data using ROS
  (Nidishlall Burton, Nicolas Piques, Axel Marty)
- Ethernet and hotspot communication
  (Asmae El Hachimi, Maxime Ramiara)
- Management between manual and autonomous control
  (Axel Marty, Asma Chouiya)
- PID controller
  (Maxime Ramiara, Nidishlall Burton)

Sprint 5

Test 1: Implement and test the trajectory tracking with **PID** corrector

- **5% of precision** from the center of the gate

Test 2: Manual control during an autonomous movement

- Recovery time **< 500 ms**

Sprint 5

1) Sprint 5 ’s objectives
2) The planned tasks
3) Acceptance tests
4) **Demonstrations planification**

Demonstrations :

- The car is able to cross a gate in a simple situation (step 3 and 4 will be showcased)
- The car can be manually controlled during an autonomous movement

The minute of the Scrum Master

Expected tasks

- **M** Tracking
- **S** Autonomous & manual state
- **L** Detect and move

In progress tasks

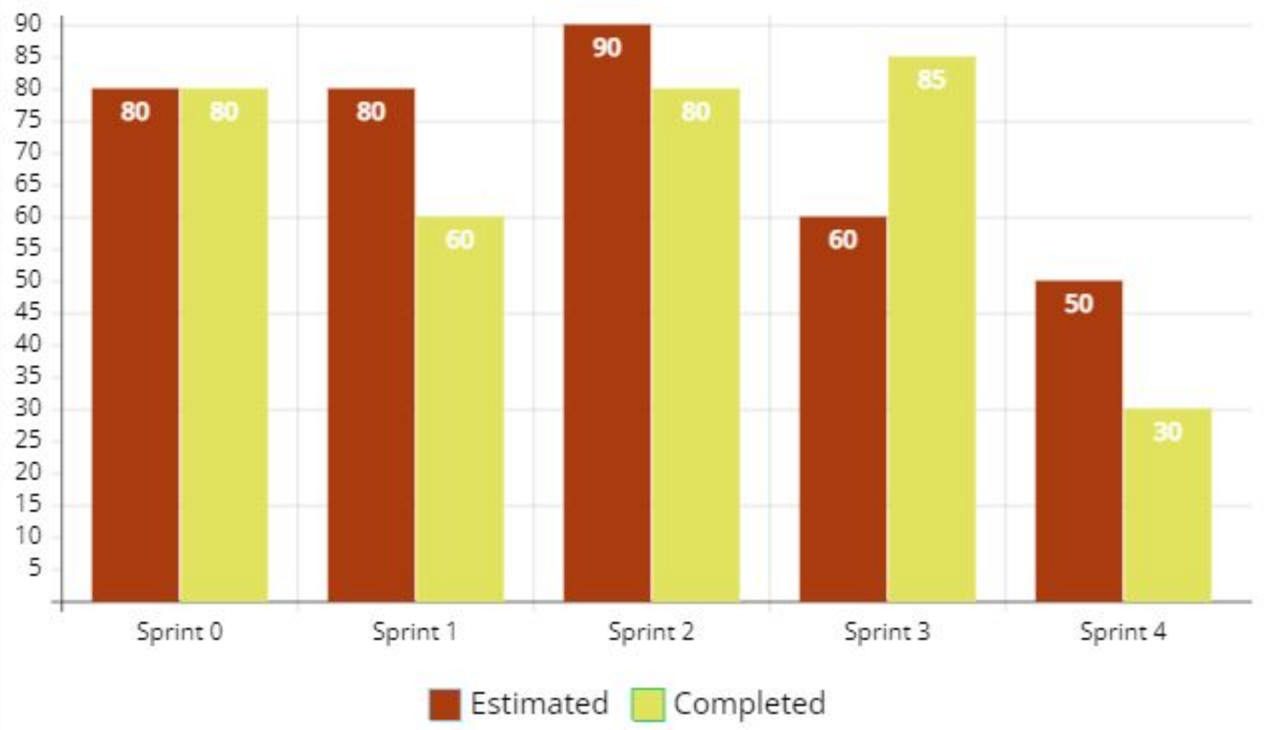
- **M** Ethernet connection
- **L** Path calculation
- **S** Autonomous & manual state

Performed tasks

- **M** Tracking
- **M** Remote control
- **S** Project review preparation

Team velocity

user stories(%)



Thanks !

Any Questions?