REAL TIME INFORMATICS

S9 CSI

ENIT

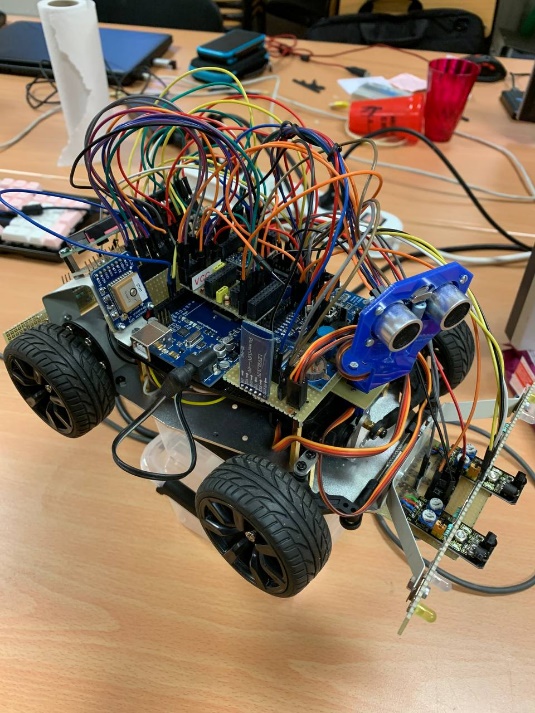
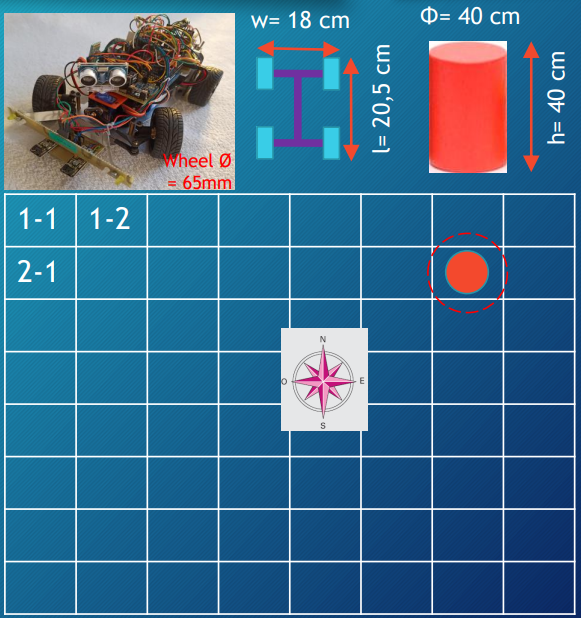
Rémi TISSEUR – Antonin CATUSSE – Nathan DEL PUPPO

# RTI Report:

Introduction:

What is a real time informatic system?

It’s an embedded system (application or set of applications) able to adapt itself to the dynamic changes of its environment.

Throughout this project we used RTOS which is a free, open source RTI system, that can be implemented on an Arduino board, a mega 2560 Arduino board.

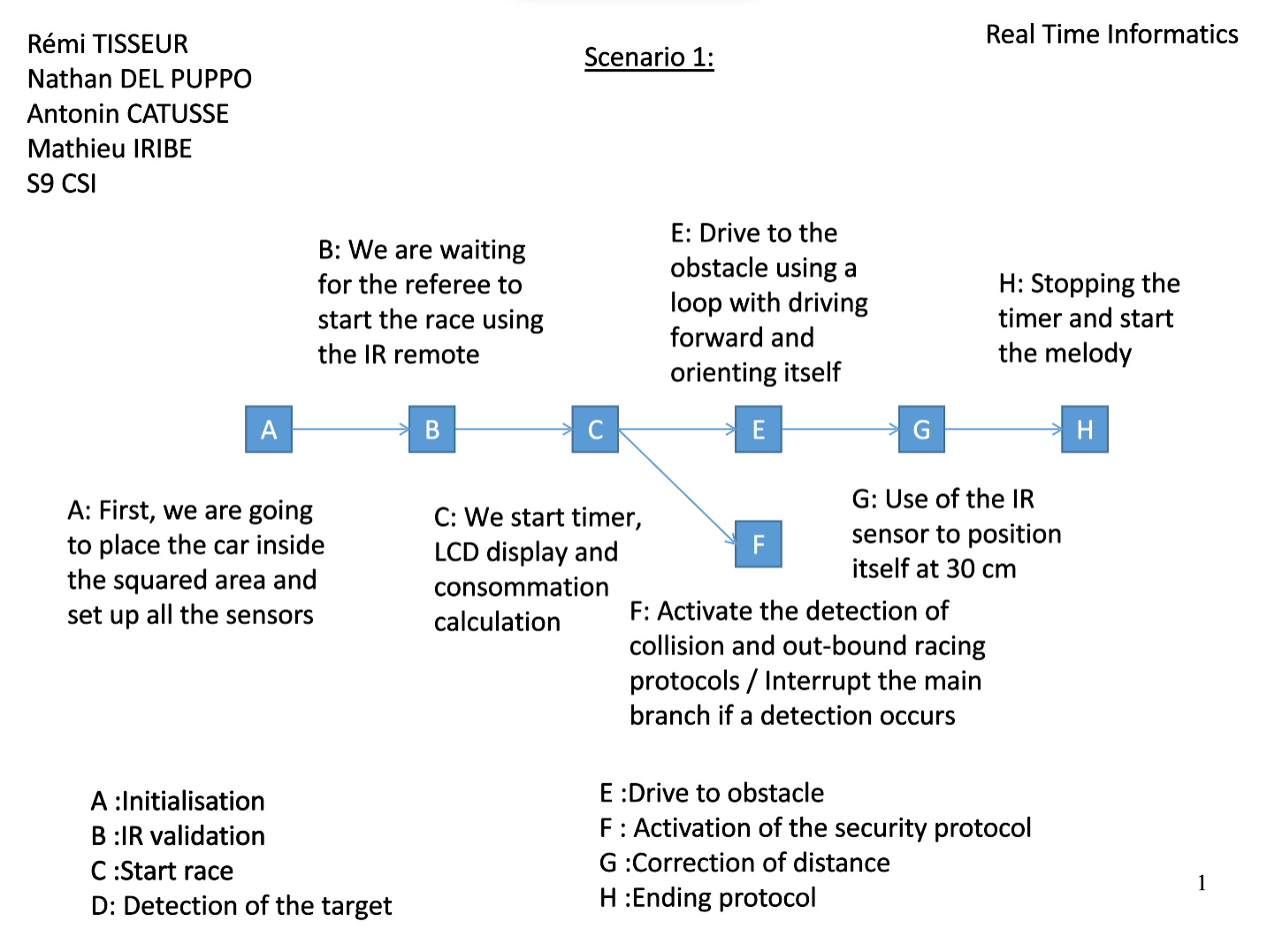
The project was to program an autonomous car in order for it to find an obstacle randomly set on a four-meter by four-meter area. And to park at 30cm from it.

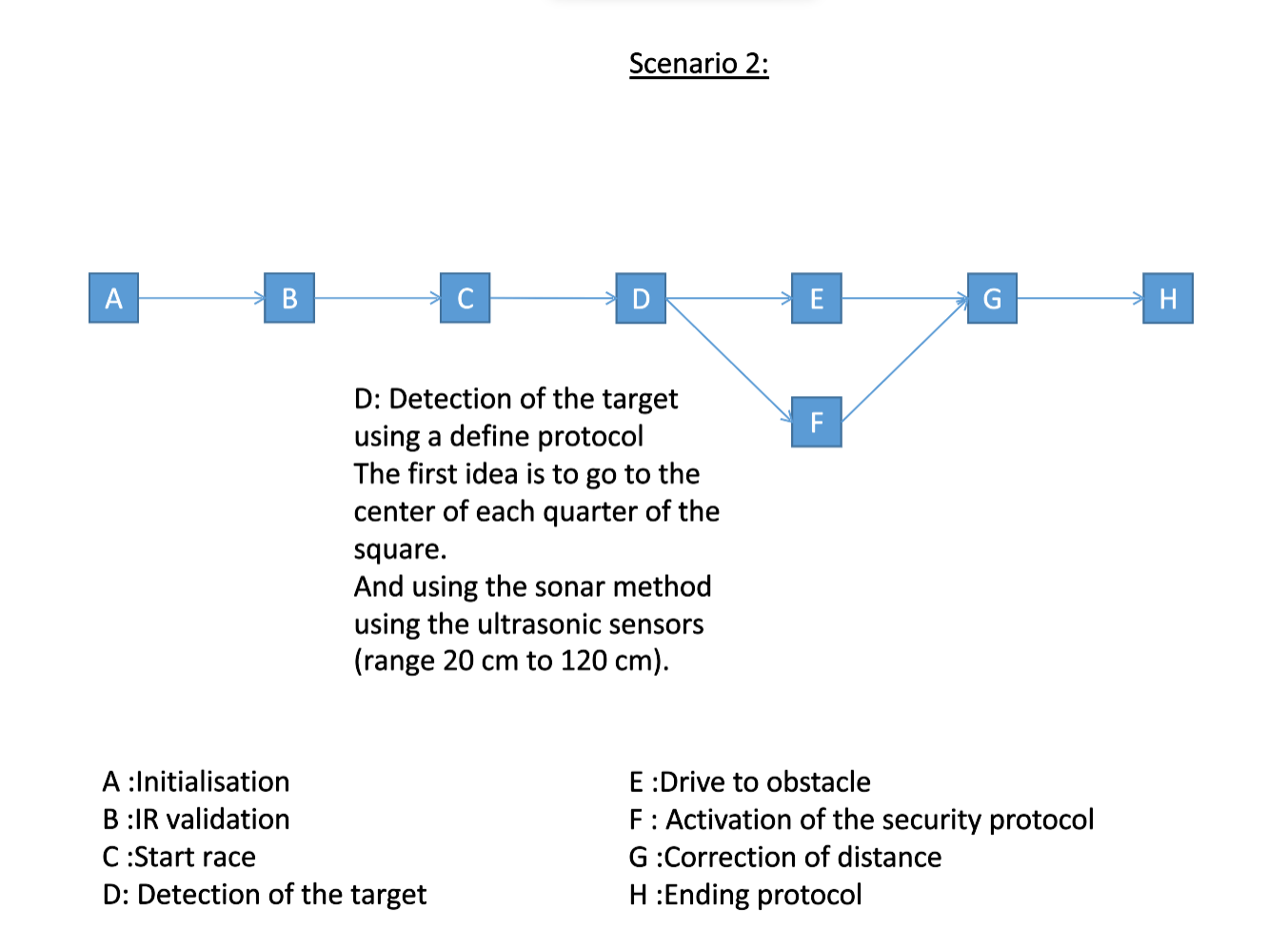
Strategy:

We started by doing a logical diagram for both scenarios in order to have an idea of all the sensors, the actuators to wire and program and get more insights on the moving strategy.

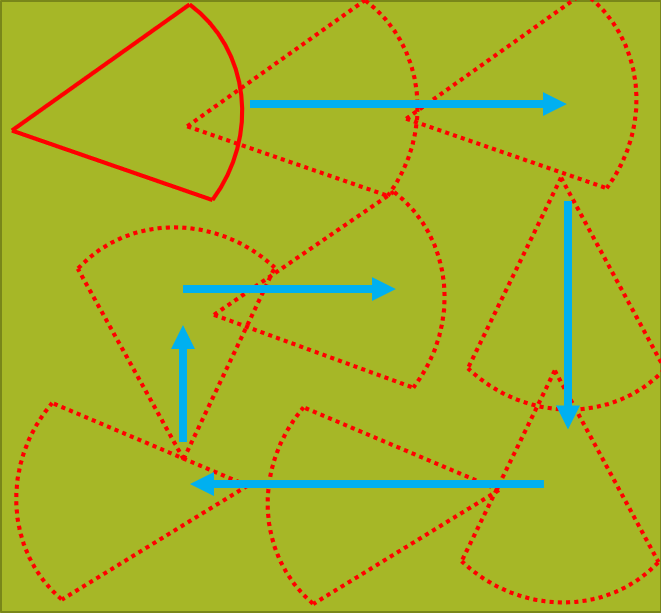
Here are our 2 logical diagrams for the 2 scenarios:

One scenario in which we know the position of the obstacle we want to find with the car, one other scenario in which we don’t know the position of the obstacle so we first have to detect it then go in his direction.





Strategy trajectory:



We wanted to use a strategy that was easy to implement because it will be easier to code afterwards.

The car follow the edges and go into the corners then into the middle to cover the whole area 4mx4m to find the obstacle.

Difficulties:

We coped with some difficulties throughout this project, in particular concerning the hardware part, such as the batteries, ultrasonic sensors and odometric sensor.

Compass

Our compass wasn’t working even when we tried to search online and implement code found online. After being wired and initialized, we have tried to compare the north given by the onboard compass with the north given by our smartphone. And the results were the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Smartphone | N | W | S | E |
| Arduino | NW | W | NWN | N |

As you can see, we didn’t get any South or East direction. We tried some modifications including derive calculation or calibration but none of them were working

Servomotor

Another problem we had is that our Servomotors didn’t work in the tasks, they were working individually but not inside the tasks. The reason of this could be, according to another group, the use of the timers for the odometrical sensor.

LCD

We also add a trouble with the LCD display screen, in fact, one of the solders was faulty. When we realise that, we ask the technician to redo these solders.

Code explanation:

Concerning the distance driven, we tried at first to use the odometrical sensor.

Odometry is the use of data from motion sensors to estimate change in position over time.

We have some issue with it because it had to use a timer that was exactly the same as the servo motor.

Overall, we had to choose between the use of the odometrical sensor and the servomotor for ultrasonic sensor and the direction.

We chose the Ultrasonic sensor because we had to scan the area in order to find the obstacle.

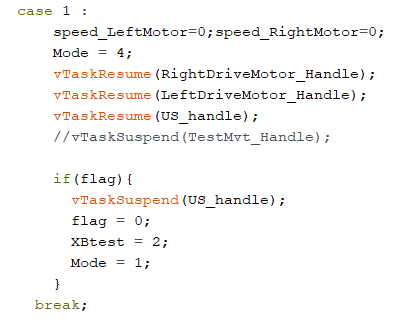
Moreover, we were able to replace the odometrical sensor by a Task delay in order to get the car moving and knowing the distance it has driven.



In the code above, it’s a Task taking care of the movement of the car.

There are 3 switch cases, each one represents a movement of the car (whether it’s forward, backward or rotating).

There is another case corresponding of the scan’s task, which consists in stopping the car, rotating the ultrasonic sensor thanks to a servo motor and display the distance of the obstacle in front of the car.



We can notice right here that the speed of the motor is set to 0 which means that the car won’t move and the task corresponding to the scanning is resuming “vTaskResume(US\_handle);”

The Ultrasonic sensor task is described right below, the servo motor will rotate the Ultrasonic Sensor by 3 units while scanning. (After doing several tests we found out that this rotating speed was the best in our opinion).

Furthermore, this task will display the distance calculated by the ultrasonic sensor and display it on the LCD screen present on the car. This function aims not to be definitive but only for debug propose.



At the end, we must display the stopwatch, the consumption of energy used by the car and the total distance.

To get the consumption of the car we used the power supply measurement and converted the value in Wh. For the distance we used the odometric sensor located in the motors. Finally, for the stopwatch, we used the timer of the arduino board.

Sadly, we didn’t have time to build this in real time task

Conclusion:

It was a great experience for us to work on this project, we have discovered RTOS, however, due to a lack of time we weren't able to finish the project. We also had issue with the battery which delayed us on the project as we got to deal with it.

This project motivated us to learn more about real time informatics system and how they work in our daily lifes.

If we had to do this project again, we would change our approach by delegating more our work throughout the members of the group and test all the components first before doing any code.