MOBILE MAPPING CAR

Requirement specification

Supervisor: MSc. Torben Gregersen

Team: AAD1

Editors:

- Guillaume Trousseau
- Jaime Escuer
- Iñaki Abadia
- Roger Prat

AARHUS UNIVERSITY

Applied App Development
(ITIPRJ)

Content

1 Opening

1.1 Purpose

In this document we want to describe the requirements of a mobile mapping car. The main purpose of the project is to create maps of the rooms where we are with our car. We will make a smartphone app, with this smartphone app we will be able to control our car. We'll make a connection between the smartphone and the microcontroller of our car, also between this smartphone and another one in order to receive its live stream video. Some ultrasonic sensors will be installed to the car, they will collect data from the room and send it through the microcontroller to the smartphone. The smartphone will plot different points based on the data received that will approach the outline of the room.

1.2 Reading instructions

1.2.1 Chapter 2: General description

Contains the general description of the system.

1.2.2 Chapter 3: Use cases

Contains information about the functional requirements or use cases of the system.

1.2.3 Chapter 4: Non-functional requirements

Contains information about the non-functional requirements of the system.

1.2.4 Chapter 5: External Interface requirements

Contains information about the graphical user interface.

2 General description

2.1 System description

The mobile mapping car is an interesting project regarding its capabilities of mapping rooms which can be difficult for human access. These rooms can even be without any source of light. The system will mainly consist of a small robot car, an Arduino microcontroller board, a Wi-Fi shield, some ultrasonic sensors, a smartphone with camera and an "App" to control everything via another Smartphone.

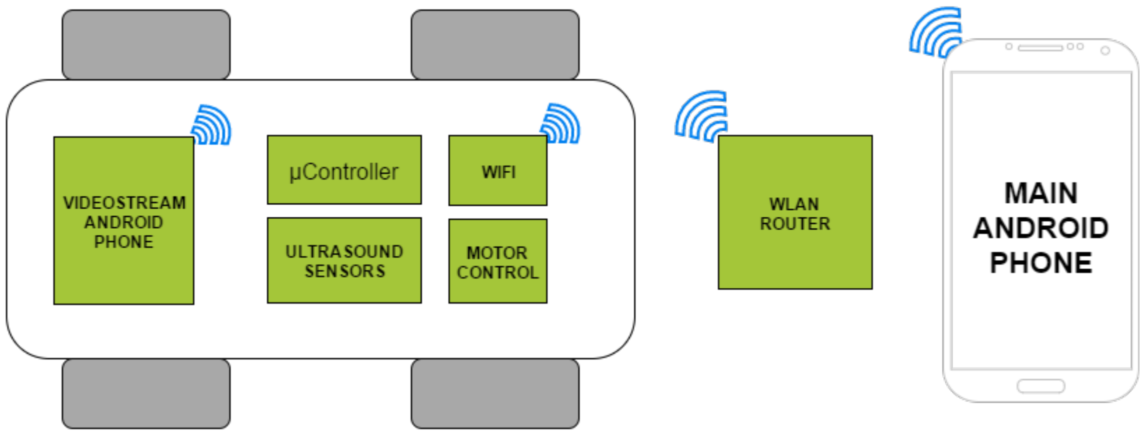
The user interface is going to make the user able to control the direction of the car (front, stopped or back), the steering (right, middle or left) and when to start recording data and when to stop. To record this data the car must not move so, the user will not be able to move the car while it is recording. However, when the user decides to stop recording data, he will be able to control the car again. The user will also be capable of watching a live stream of what the car will be seeing (if it is not a really dark room) while he is controlling it. After the desired data has been recorded, the plotted points should show an approximate view of the layout of the room.

The Smartphone used to control the project will communicate via Wi-Fi with the microcontroller and with the other smartphone.

On one hand, the other smartphone is needed because, with our microcontroller, we are not able to stream video so, we need it to stream the live video.

On the other hand, the uController board will communicate through its Wi-Fi shield with the controller smartphone, the uController will send the orders to the car's engines to move and to turn (with PWM signals), it will collect data from the sensors and it will also send the data collected to the smartphone application.

2.1.1 System Overview

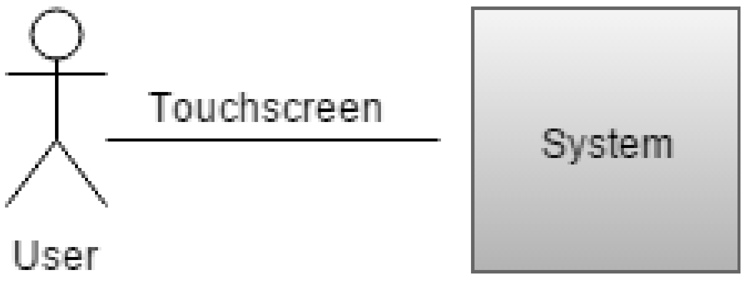


In figure 1 the main components of our project are shown. We can differentiate two main parts, the car and the main Smartphone. The car is composed by:

- Another smartphone, in charge of the video streaming.
- Motor control.
- Ultrasound sensors.
- The Arduino MEGA and WIFI shields, steering the car and sending back the sensors data.

2.1.2 Context Diagram

Our system doesn't interact with outer devices or services. Only the user interacts with the system via the main smartphone's touchscreen.



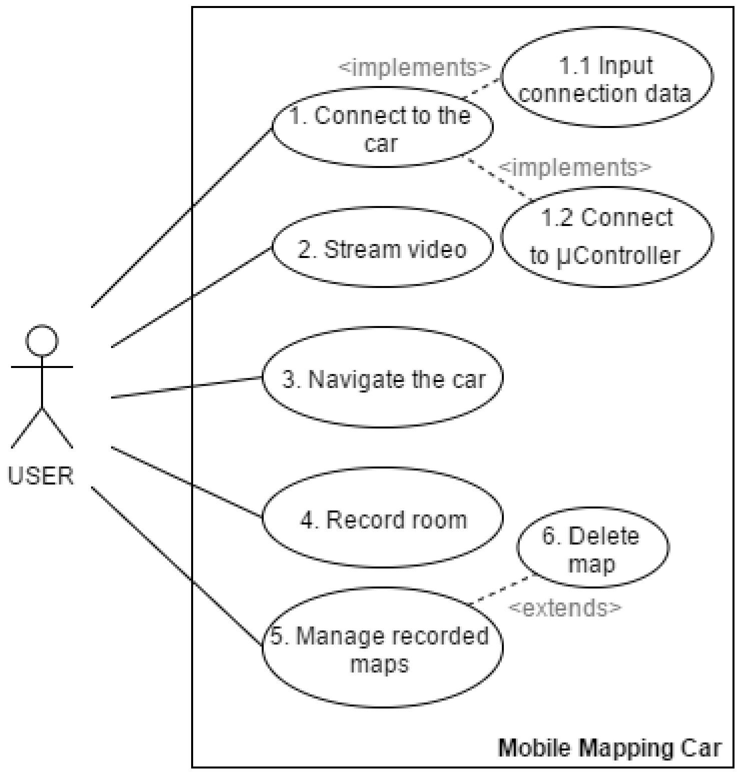
AARHUS UNIVERSITY

2.1.3 Actor description

Actor name	User
Type	Primary
Description	The user is the main actor. He steers the car and decides when the system starts recording data from the sensors via the smartphone application
Number of simultaneous actors	One

2.2 Functional requirements

The functional requirements are described with the use case diagram. The figure shows the diagram for the Mobile Tracking Car. The aim of this diagram is to show an overview of features which can be performed.



3 Use cases

3.1 Use case 1.1: Input connection data

This use case describes how the user can input the necessary data for initializing the connection with the rest of the system

Initiator: The user.

Pre-condition: The user is facing the Start Screen.

Post-condition: The user's smartphone and the car's system connection data are ready.

Scenario:

1. The user pushes the connect button on the "start screen".
2. The "connection screen" is displayed.
3. The user enters the IP and port of the car's microcontroller.
   [Exception 1]
4. The user enters the IP and port of the car's smartphone/IPCam.
   [Exception 1]
5. The data is ready to use for the connection

Exception:

- Exception 1: The IP or port provided by the user is either invalid or missing. The user is informed that one of both is incorrect and he can retry to enter them.

3.2 Use case 1.2: Connect to uController

This use case describes how the user can connect his smartphone with the car's system.

Initiator: The user.

Pre-condition: The car's microcontroller is powered and secondary smartphone listening for connection requests. The user has already entered the data for the connection.

Post-condition: The user's smartphone and the car's system are connected.

Scenario:

1. The user pushes the start recording button on the "start screen".
2. The smartphone tries to connect to the Arduino.
   [Exception 1]
3. The car's system and the user's smartphone are connected.

Exception:

- Exception 1: The IP or port provided by the user is invalid. The user is informed that one of both is incorrect and he can retry to enter them.

3.3 Use case 2: Stream video

This use case describes how the smartphone on the car streams a video to the user's smartphone. The camera of the car's smartphone records a video and transmits a stream to the user's smartphone. The video is shown on the main screen of the user's smartphone.

Initiator: The car's smartphone.

Pre-condition: The car's smartphone and the user's smartphone must be connected using the Wi-Fi network.

Scenario:

1. The car's smartphone captures a video.
2. The car's smartphone sends the video stream to the user's smartphone. [Exception 1]
3. The user's smartphone displays the video on the interface.

Exception:

- Exception 1: The user's smartphone is not correctly connected with the car's smartphone. The user is informed that the connection failed.

3.4 Use case 3: Navigate the car

This use case describes how the user moves the car with his smartphone. The user can move the car forward and backward or make it turn. Only one speed is available.

Initiator: The User

Pre-condition: The car and the user's smartphone must be connected using the Wi-Fi network.

Scenario:

1. The user's smartphone provides an interface allowing the user to move the car forward / backward and left/right.
2. The user commands the car to move.
3. The user's smartphone transmits the user's input to the car. [Exception 1]
4. The car moves forward / backward.

Exception:

- Exception 1: The user's smartphone is not correctly connected with the car. The user is informed that the connection failed.

3.5 Use case 4: Record room

This use case describes how the user can launch the process of collecting data. The user can launch a scan of the area using the ultrasonic sensor. The results of the scan will be used to create a map of the room.

Initiator: the user.

Pre-condition:

- The car and the user's smartphone must be connected using the Wi-Fi network.
- The car is not moving.

Scenario:

1. The user's smartphone provides an interface allowing the user to launch the scan.
2. The user launches the scan.
3. The user's smartphone transmits the user's input to the car.
   [Exception 1]
4. The car's ultrasonic sensors scan the area / room.
5. The car transmits the results of the scan back to the user's smartphone.
   [Exception 1]
6. The platform where the sensors are mounted turns by 10º.
7. Steps 4 to 6 are repeated until the user stops the scan or the sensors have collected data from the complete spin (360º).
8. The user's smartphone builds a map using the data received.

Exception:

- Exception 1: The user's smartphone is not correctly connected with the car. The user is informed that the connection failed.

3.6 Use case 5: Manage recorded maps.

This use case describes the user's access to the list of saved maps. From the "control screen" and from the "start screen" the user can access the map list. The maps list consists of a list of the maps created by the user's smartphone from the car's results. The interface is separated in two parts: the list and a preview of the currently selected map.

Initiator: The user.

Pre-condition: At least one map must be saved.

Scenario:

1. The user pushes the "map button" from one of the two screens.
   [Exception 1]
2. The maps list is displayed on the left; the preview of the map at the beginning of the list is displayed on the right.

3. The user selects a map from the list.
4. The preview of the map selected is displayed.
5. The user selects the preview.
6. The selected map is displayed in full screen mode.

Exception:

- Exception 1: No map is saved in the maps list, an error message is displayed and the user is redirected on the previous screen.

3.7 Use case 6: Delete map

This use case describes how the user can delete a map from the maps list. When a map is selected from the map list, a button is provided by the interface which allows the user to delete the selected map.

Initiator: The user.

Pre-condition:

- The user is accessing the maps list.
- At least one map is saved.

Scenario:

1. The user selects a map from the maps list.
2. The user pushes the “delete button” on the selected map preview.
3. The map is deleted.

4 Non-functional requirements:

4.1 Distance:

Our project is based on mapping a room with ultrasonic sensors, and these sensors must have a Ranging Distance between 2cm – 400 cm.

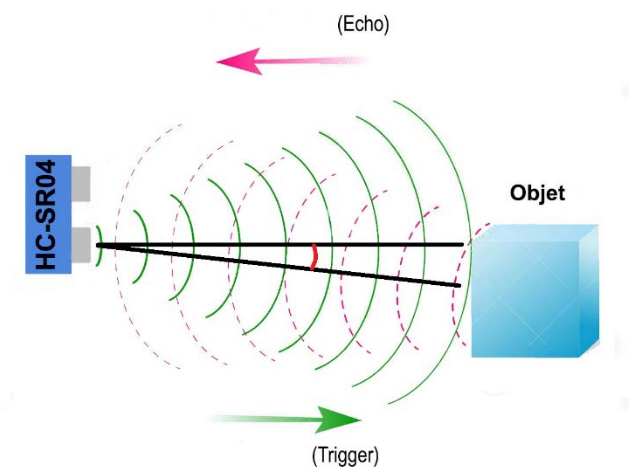
In lower and higher measures outside this range, our car can’t get correct values.

We also have a limitation with the distance between our user interface (smartphone) and the car with Arduino WIFI shield incorporated. This distance outdoors is at max 150m, but even though we’ll use it indoors and this range will be reduced due to walls and other obstacles, it’s more than enough for our purposes.

4.2 Angle measurement:

When the sensors make the original wave (Trigger) the object which receives the wave has to be sufficiently faced.

The effectual angle of the sensors is 15°-30° approximately, in case that the wall or the object has a higher angle the measure will not be correct.



http://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04

4.3 Small Objects:

In the same case than the angle measurement, our car will find circular and narrow objects. In this situation the waves' bounce will not correct.

This can be problematic because these objects can be the reason of getting some wrong measurements.

4.4 Noise:

This problem is usual when we when we implemented ultrasonic devices. This sensors produce waves with frequencies higher than the upper audible limit of human hearing.

As the car will be stopped while measuring, its engines will not produce any kind of noise that can interfere with the sensors.

4.5 Surface:

For our car the surface of the floor must be as flat as possible. It is necessary because the flatter the surface is, the better the obtained results will be.

4.6 Extreme conditions:

Like all electronic components our devices have humidity and temperature restrictions.

Searching in forums about functional temperature of Arduino we found that it is in the range of -40°C to 85°C. The temperature will not be a problem in our project, but the humidity is our main external problem.

Our car does not have bodywork, and all the devices like the microprocessor and sensors are exposed. We have to be careful with rain or areas that can make our car wet.

http://forum.arduino.cc/index.php?topic=179077.0

5 External Interface requirements:

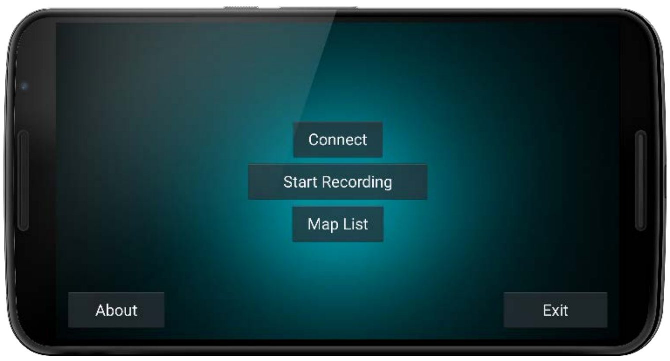
5.1 User Interface:

In this section we are going to explain our smartphone app's UI. This app will be responsible for carrying out the movements of our car and all the control of the external devices that we are going to implement.

5.2 GUI (Graphical User Interface)

5.2.1 Start Screen View:

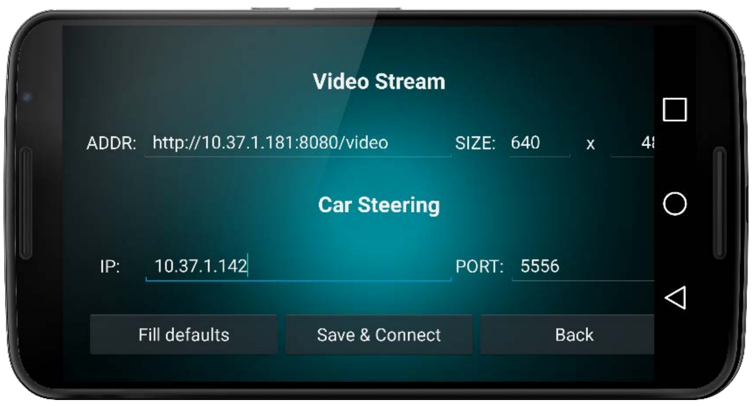
This is the main window when we open the app. We can see some buttons to choose the different windows and begin the function desire.



5.2.2

5.2.3 Connect:

We use this window to connect our car and the smartphone via WIFI. We only have to fill up the gaps with the stream video server IP address and the car's Arduino one, or we can push the default data button which will fill them with preset values.


Video Stream

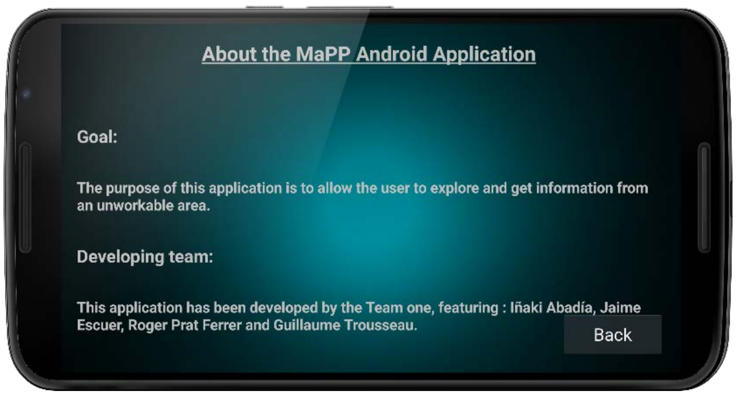
ADDR: http://10.37.1.181:8080/video SIZE: 640 x 4

Car Steering

IP: 10.37.1.142 PORT: 5556

5.2.4 About Screen:

This activity contains some information about the project and the developing team (Us).



About the MaPP Android Application

Goal:

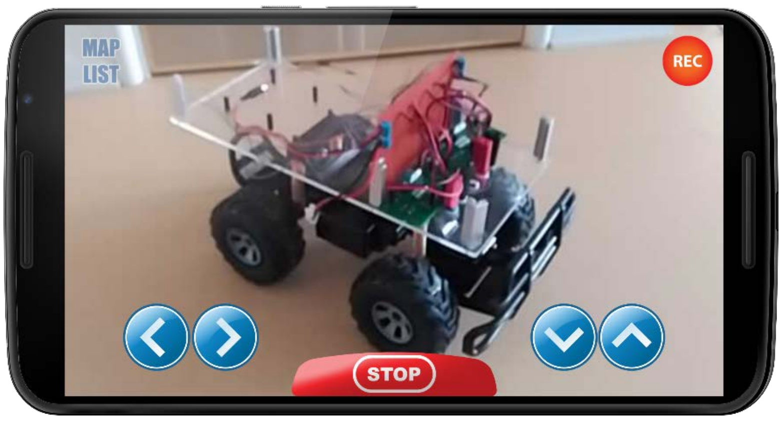
The purpose of this application is to allow the user to explore and get information from an unworkable area.

Developing team:

This application has been developed by the Team one, featuring : Iñaki Abadía, Jaime Escuer, Roger Prat Ferrer and Guillaume Trousseau.

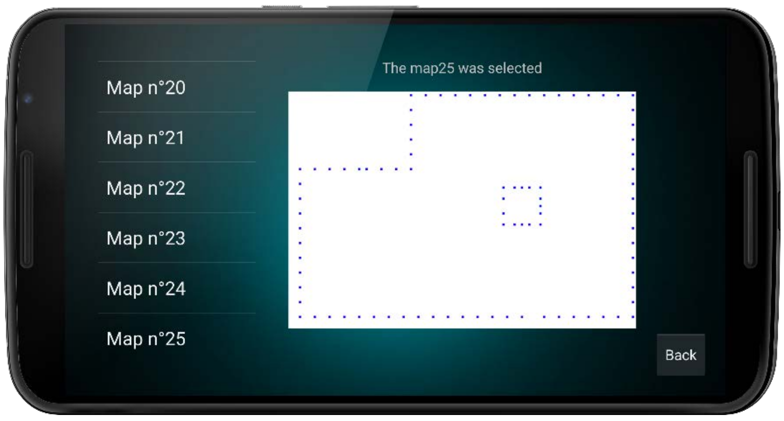
5.2.5 Control View:

In this window we have the control of our car. In the background, at the same time that we move the car, we have a live video where we can see what the car has in front of it. We can start the recording of data with the sensors with a button located in the top right of the screen.

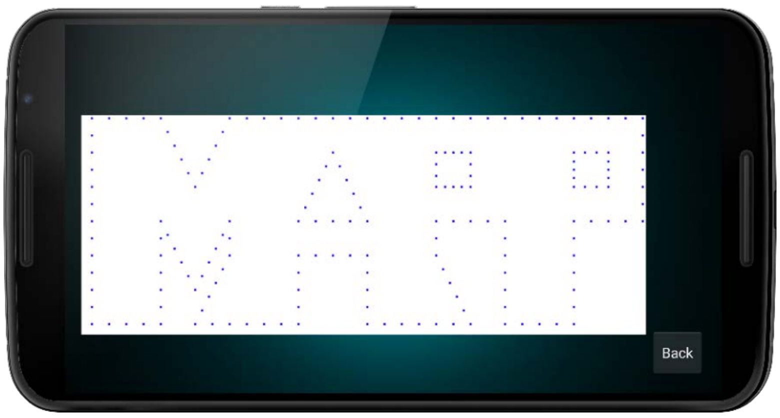


5.2.6 Map List View

This is the last window of our app, it has a little list with all the measures that the car has done. We can compare the different stored maps by clicking at their names and seeing their previews.



We can also select the preview of one of them to expand it and have a more detailed view. (In this case both maps seen are different just to show that we can have many maps stored but, in reality, when you click one preview, you get the larger view of the same map).



AARHUS UNIVERSITY