

Wi-Fi Controlled Car



Project report

This document is the project report of the group AAD1 summer 2013 about the application Wi-Fi Controlled Car.

Document title	Project report
Document ID	WCC-Project Report
Authors	Ambroise Dhenain Szymon Klepacz Anatolii Shakhov Alvaro Garcia Pierre Le Texier
Supervisor	Torben Gregersen
N°. pages	31
Date	30-08-2013

Document history

Date	Audit	Author	Description
02-10-2013	0.1	AD	Template only.
20-11-13	1.0	AD	Reorganization.
22-11-13	1.1	AD	Major revision of the entire document.
23-11-13	1.2	AD	Technologies and software.
24-11-13	1.3	AD	Packages diagram. Reorganization of the 4+1 view part. Sequence diagrams.
30-11-13	1.4	AS	Car design.
02-12-13	1.5	AD	Corrections after meeting with Torben. Design part removed and put in the Design document.
04-12-13	1.6	AD	Project management improved with pictures and details.
05-12-13	1.7	AD	Add appendix.
06-12-13	1.8	AD	Corrections after meeting with Torben. Add document summary.
07-12-13	1.9	AD	Conclusion and perspectives.
08-12-13	1.10	AD	Update reading instructions. Minors changes.
08-12-13	1.11	SzK	Correct spelling. Added links, information about hardware part and improved some point.
09-12-13	1.12	AS	Minor change in Arduino description + Correct spelling in HW.
09-12-13	1.13	AD	Corrections after meeting with Torben.
09-12-13	1.14	AS	Changed document title
12-12-13	2.0	AD	Final version. Class diagram, Gantt diagrams and appendix. Conclusion and future. Acknowledgments.

Approval

Author(s)	Ambroise Dhenain – 201301255 [AD] Szymon Klepacz – 201301204 [SzK] Anatolii Shakhov – 201301534 [AS] Alvaro Garcia – 201301220 [AG] Pierre Le Texier – 201301944 [PLT]
Project name	Wi-Fi controlled car
Document ID	WCC-ProjectReport
No. pages	

By signing this document both parties accept, that this is the requirements for the development of the desired system.

Place and date:

Authors

Supervisor

Contents

1. OPENING.....	6
1.1. PURPOSE.....	6
1.2. READING INSTRUCTION	6
2. MEMBERS	7
2.1. ROLES.....	7
2.1.1. <i>General</i>	7
2.1.2. <i>Android application</i>	7
2.1.3. <i>Car application</i>	7
3. GENERAL DESIGN	8
3.1. SYSTEM DESCRIPTION.....	8
3.2. CLASS DIAGRAM - OVERVIEW	9
3.3. TECHNOLOGIES & SOFTWARE	10
3.3.1. <i>Technologies</i>	10
3.3.1.a. Android	10
3.3.1.b. Sockets.....	10
3.3.1.c. Arduino YUN	10
3.3.1.d. Adafruit.....	11
3.3.2. <i>Software</i>	11
4. PROJECT MANAGEMENT.....	12
4.1. RISK MANAGEMENT.....	12
4.1.1. <i>Introduction</i>	12
4.1.2. <i>Dropbox</i>	12
4.1.3. <i>Git repository</i>	12
4.1.4. <i>SugarSync</i>	12
4.1.5. <i>Extra Arduino YUN</i>	13
4.2. SUPERVISOR CHECKPOINTS.....	13
4.3. GANTT DIAGRAMS.....	13
4.3.1. <i>Project management - Gantt</i>	14
4.3.2. <i>Android - Gantt</i>	15
4.3.3. <i>Car & Arduino - Gantt</i>	17
5. DOCUMENTS SUMMARY.....	19
5.1. PRE-ANALYSIS	19
5.2. REQUIREMENT SPECIFICATION	19
5.3. DESIGN	20
5.4. TESTS	20
6. CONCLUSION & FUTURE	21
6.1. PLANNING	21
6.1.1. <i>Planning resume</i>	22
6.2. RESULTS.....	22
6.3. DISCUSSION OF RESULTS.....	22
6.4. STRENGTHS AND WEAKNESSES	23
6.4.1. <i>Strengths:</i>	23

6.4.2. Weaknesses:..... 23

6.5. SUGGESTED IMPROVEMENTS 23

6.6. CONCLUSION..... 24

7. GLOSSARY25

8. REFERENCES.....25

9. APPENDIX26

Tables of figures

Figure 1 - Communication overview	8
Figure 2 - The car with the embedded Arduino	8
Figure 3 – Android class diagram – Overview	9
Figure 4 - Planning part #1 - Project management (Gantt).....	14
Figure 5 – Planning part #2 – Android (Gantt #1)	15
Figure 6 - Planning part #2 – Android (Gantt #2).....	16
Figure 7 - Planning part #3 – Car & Arduino (Gantt #1).....	17
Figure 8 - Planning part #3 – Car & Arduino (Gantt #2).....	18
Figure 9 – Planning overview	21
Figure 10 - Planning part #1 - Project management	26
Figure 11 – Planning part #2 - Android	28
Figure 12 - Planning part #3 – Car & Arduino	30

ACKNOWLEDGMENTS

In first place, we want to thanks the jury who accepted to assist at our project presentation and judge our work.

We want to thank our supervisor, Mr Torben Gregersen for his help during this whole project and all his advices. He provides us his help at any time and was really interested by our project. Without him the result would not be the same, for sure.

We also want to thank the Aarhus University that accepted to receive us as student for the summer 2013-2014 semester, for the infrastructure and for the hardware it provides to us, allowing us to work with a material high quality.

Finally, we want to thank the Denmark and the Danish people, because your country is awesome and the life here is really good. We really like this country and we are sad to go back in our native country soon.

We almost forgot to thank the Danish beer, without it this country would not be the same! Øl!

1. Opening

1.1. Purpose

The purpose of this document is to explain our project: *“Wi-Fi Controlled Car”*.

We will talk about the project itself, we talked about its purpose in the document *“2 - WCC - Requirements specification”*, we will now talk about our design, our planning, the project management including task's distribution and everything that was be done during this project.

1.2. Reading instruction

- Chapter 2: Contains information about the team, the sub-groups and the task distribution.
- Chapter 3: Describes the general design, the overview, the technologies and software we use.
- Chapter 4: Resumes the contents of the other documents such as Pre-analysis, Requirement specification, Design and Tests documents.
- Chapter 5: Describes the project management which contains the strategies and the planning using Gantt diagrams.
- Chapter 6: Final analysis about our project and our results, the planning, the reach of our goals and the future of our application.
- Chapter 7: Glossary which contains explanations about some words and abbreviations used in this document.
- Chapter 8: Lists the references we used in the project, such as source code, documentation and more.
- Chapter 9: Appendix that contains detailed planning, task by task.

2. Members

2.1. Roles

2.1.1. General

The official **project** manager is **Ambroise Dhenain**.

The official manager for the **Android** application is **Ambroise Dhenain**.

The official manager for the **Hardware & Arduino** application is **Szymon Klepacz**.

2.1.2. Android application

There are two members principally working on the Android application:

- Ambroise Dhenain
- Alvaro Garcia

2.1.3. Car application

There are three members principally working on the Car application:

- Szymon Klepacz
- Anatolli Shakhov
- Pierre Le Texier

3. General design

3.1. System description

The “Wi-Fi controlled car” is to be designed to be easy and user friendly to use with some extra functionality which doesn't exist on a simple car remote game like live video stream, pictures, collision warning and more.

The system will consist of two different parts, the car first, which is a simple car robot we borrowed from the University. The car can move forward, backward and turn left and right, as usual. The second part is about a phone application, actually we will only use Android for the moment. But it could be done for Windows or Apple phones. This application will display a video stream from the camera on board of the car and some buttons for control the car. (Turn, go forward, etc.)

The car will have its own battery for all the components on board (motors, camera and so on).



Figure 1 - Communication overview

The Wi-Fi network is generated by the Arduino YUN, he is able to acts as a hotspot and the Android phone has to connect to this hotspot to be able to communicate with the Arduino.

The user can use also a computer, instead of a smartphone, but we didn't develop any software on a computer and it's better to use a smartphone, because the user can move and follow the car, the application is designed to control everything on the car

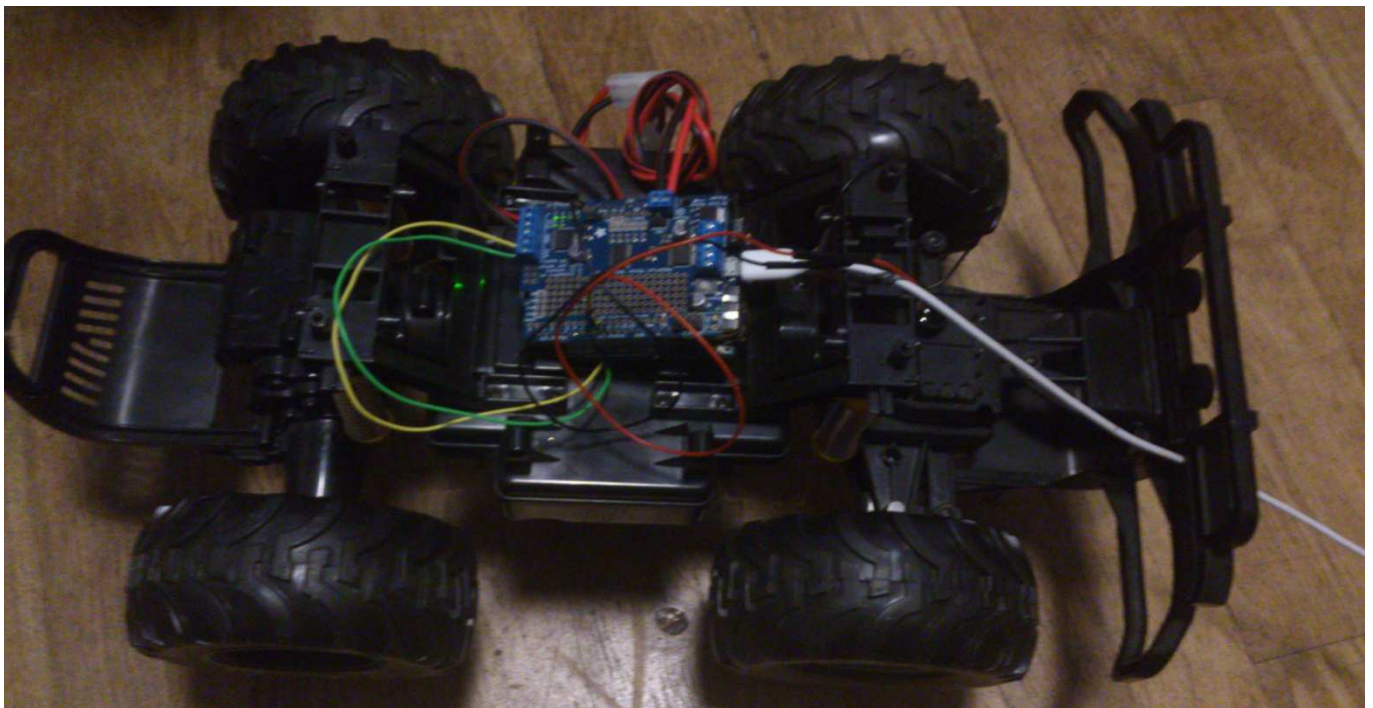


Figure 2 - The car with the embedded Arduino

3.2. Class diagram - Overview

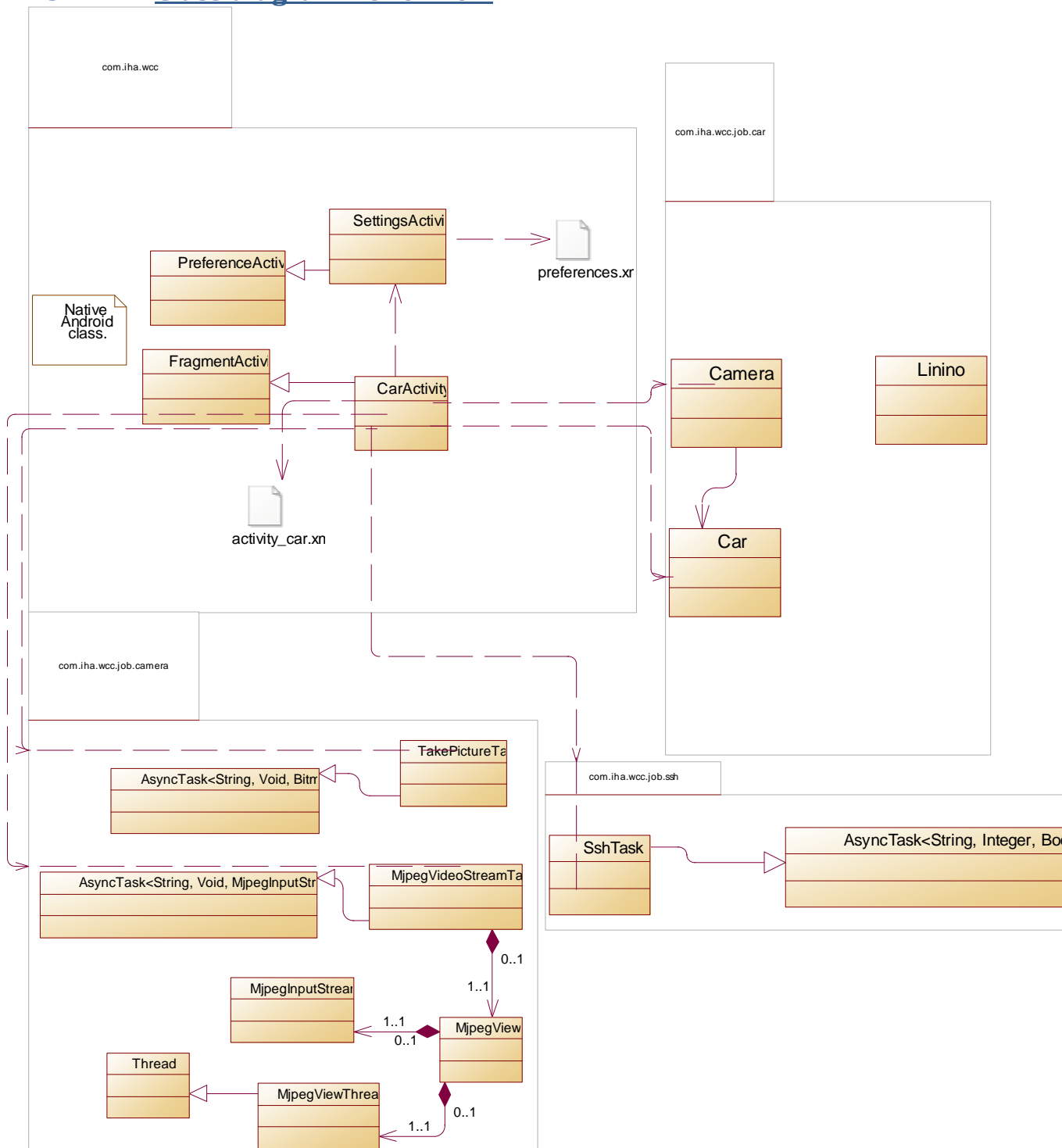


Figure 3 – Android class diagram – Overview

This diagram gives an overview about the interactions between classes and packages. A complete diagram is available in the “3 - WCC - Design” document.

3.3. Technologies & software

3.3.1. Technologies

3.3.1.a. Android

We used the Android Framework to develop the application. The application can be installed from smartphone that support the API 11 and upper.

Android uses widespread java language, Java SE7.

3.3.1.b. Sockets

We use sockets to communicate between the Android application and the Arduino application, it's the best way to communicate because it's really efficient, almost real time and really better than HTTP requests, we started by using simple HTTP requests but the response time was too important so we decided to change and use sockets instead.

3.3.1.c. Arduino YUN

Arduino board was chosen before because it provides great possibilities for development of small controlled systems. This small board can control sensors, motors, other small devices using a smart Atmel ATmega32U4 processor. It has a 32 KB of built-in Flash Memory which is enough for almost all tasks we need to accomplish in our application. The only problem was to load the program together with the video stream – we had some troubles because we didn't have enough space to achieve all the tasks in a small space. Later on we could use another way of video stream and memory size was right enough for our program.

Another outstanding feature of this board is a Wi-Fi module. It has its' own web interface and the user can interact with the board using either LAN connection or the wireless. For our project we will use a wireless connection to establish data exchange between the phone and the board.

Arduino Yun has its' own operating system called Linino. This is open source software combining the power of Linux OS with Arduino hardware. It controls the wireless web interface and the incoming ports (Ethernet and USBs).

As well as operating system, Arduino board had its' own programming language (the language combined from C and C++). It doesn't use classes like usual programs but we had to declare functions instead. This programming style simplifies the process and improves memory consumption.

With a slot for Micro SD card the board has much more possibilities. In our remote car control project we will use it to store temporary data from the camera, it's necessary to have a video stream.

Why Arduino?

The key point in the board decision was its' versatility. It has different types of connections available, the board already has a Wi-Fi module (no additional expenses), the price/profit level. From the beginning we had to decide on how we will establish connection between the board and the phone, how will we start motors and which features will be included in the project. All these choices played a huge role in making a final decision on the needed board.

3.3.1.d. Adafruit

Adafruit motor shield is the extension for Arduino board. We need it to control two separate motors (the front one is responsible for rotation (turn left/right) and the rear one is responsible for sense (backward/forward)). It has an Adafruit library which eases the process of creating commands for Arduino.

With usage of simple commands user can easily write down the needed code to control the rotation of the motors. Inside of it we call a variety of functions to accelerate, fully stop, turn, change the direction of movement/rotation, the speed and the delay timers. This shield allows us to gain a full control over the car movements. With its' help we were able to create an android application (see Android chapter) to increase and decrease the speed of the motor.

Out of the box it was a board without headers and any connectors. At the very first week after obtaining the shield we had to solder headers and terminals. We used an official guide (<http://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/install-headers>) to secure the assembly process and when the board was ready we connected it to an Arduino. We had a second version of shield which was not created for the Arduino Yun board (Yun has an Ethernet port) so we had to use more headers between the boards. The port cover was too high to apply the board to an Arduino Yun with standard headers, and we used another set to put the shield right above the port.

We are using an external library downloaded from Adafruit. Notice that by Android we control microcontroller on Adafruit board (by I2C), not motor chip. It let us to use all PWM pins on our Arduino. We are using only one I2C address.

3.3.2. Software

- **PowerAMC 15.0:** Used to design the application, UML diagrams basically.
- **Microsoft Project 2010:** Planning management software.
- **Word 2010:** Used to write the most important part of the documentation: the reports.
- **IntelliJ IDEA:** IDE used to develop the Android application. Better than the basics one: Eclipse with ADT plugin. From Google.
- **Arduino IDE:** IDE provided by Arduino vendor to develop Arduino application.
- **Atmel studio** - programming plugin that provides all of the features found in the **Arduino IDE**
- **Fluid UI:** Online and free software to create application mock-up.
- **Project viewer:** Online and free software to read and edit planning created by Microsoft Project. Useful to read or edit planning without have installed the software (not free) on the computer.

4. Project management

4.1. Risk management

4.1.1. Introduction

The risk management part is about “What does the group uses to avoid risks and issues?”

It’s basically up to the project manager to decide which strategies use to avoid risks in the future. Sometimes, some strategies are proposed by the team manager and sometimes by the group.

We will explain each strategy we used below.

4.1.2. Dropbox

Dropbox is famous software used by team to share documents. A Dropbox repository was created at the start to share files between team members.

The purpose of this strategy is to be always up-to-date with the shared files and be aware about updates from other people. It’s also a versioning system which allowed to restoring deleted files and manages parallel updates on the same files.

As project manager, I’m used to use Dropbox as file sharing and other members too. We tried Google Drive in first place but it was as efficient as Dropbox so we changed.

*Dropbox was a **group** strategy.*

4.1.3. Git repository

Git is famous by now, more than other versioning software such as SVN or Mercurial. It’s basically an online repository where we can store our files (mainly source code). We chose to use a public repository, that means everyone can see our project, download our source code and documentation and have rights to use it (depending on our privacy settings), but they have **read-only** access right on it. Only team members have rights to update the repository content.

Git is really a powerful tool to manage source code versioning and merging, because we are several to work on the same files at the same time, it’s really important to use this kind of tools to avoid problems with source code merging. Do that automatically using Git saved our time.

We can use Dropbox for such need, it’s not that powerful, it can’t automatically merge file content and the versioning is not enough powerful for development needs.

*Git was a **group** strategy.*

4.1.4. SugarSync

In order to avoid risks such as loss of file, an automatic synchronization between the **project manager’s computer** and external **file system storage** was made.

Especially useful for files that was not in the Git repository or in the Dropbox directory (documentation, planning and reports) but also because these two strategies require a specific user action such as **commit & push** (Git) or

copy & paste (Dropbox) to be useful, this strategy is automatic, it's triggered each time a file is updated, deleted or created. It has its own internal versioning system.

The software used is **SugarSync**. It can also synchronize files between several laptop and smartphones. This kind of strategy is particularly useful against computer issues such as viruses, broken/stolen hardware, unexpected computer shutdown and more.

*SugarSync was a **project manager** strategy.*

4.1.5. Extra Arduino YUN

When we broke the first Arduino YUN, we kindly asked to Torben to command a second one. The project manager decided to buy another one for himself in case of the Arduino commanded by the University was late, because the first one took almost three weeks between the time we commanded it and the time we got it.

Furthermore, we could also break the second Arduino even if we got it in time, because we didn't really understand how the first one was broken.

Finally, we broke the second Arduino and without the third one we would not be able to complete our project.

*The extra Arduino was a **project manager** strategy.*

4.2. Supervisor checkpoints

We decided with our supervisor, Torben Gregersen, to have a meeting each week to discuss about the advancement, the issues and the next steps.

It was also a moment where we were talking about the advancement of the different parts (Phone and Car) between us. We discussed to prevent issues and problems during these meetings and be sure about what we have to do for the next time, which is the next step and so on.

Due to these meeting, we were sure that the project was in the good direction, especially with Torben's advices.

4.3. Gantt diagrams

The purpose of our planning was not to plan what we will do but to write what we did every week and see the tasks advancement. We used Microsoft Project 2010 to make our planning. The planning was writing by the project manager, each member could see it at any time and ask for updates or update it themselves and send the new version to the project manager to be validated.

A planning is a good way to see the time left for the project, but also to see the time used for each tasks and to have an overview. It's useful to see the tasks advancement too.

Especially useful for the project manager, our planning is really precise and has a lot of information about our tasks, delays, issues and more.

We didn't do an estimated planning because there were too much unknown things about the project such as the time to get the components. We preferred do only one planning, to save our time.

Below, we will explain our planning and tasks, part by part, but you can also consult the **6.1**.

4.3.1. Project management - Gantt

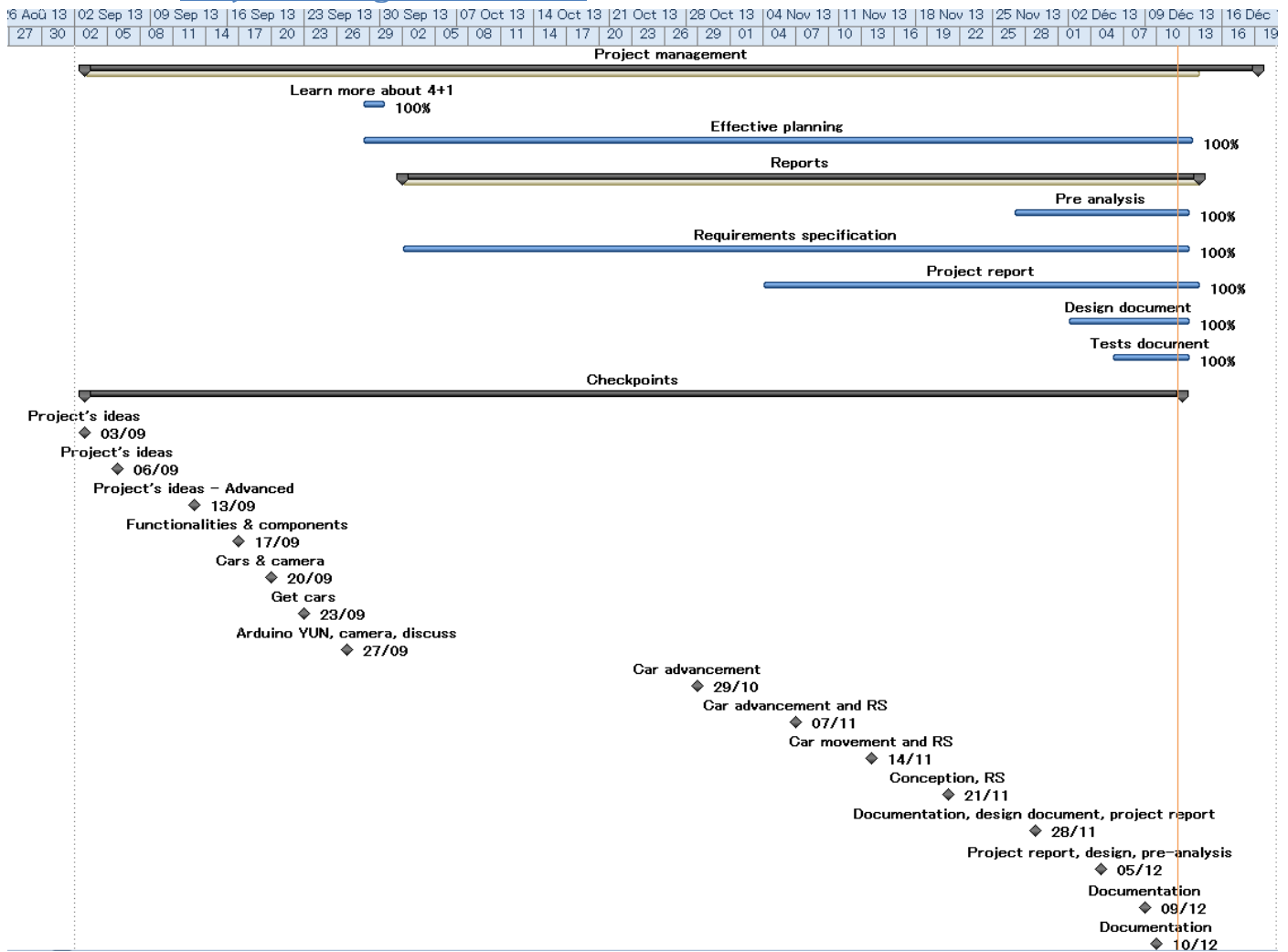


Figure 4 - Planning part #1 - Project management (Gantt)

This is a Gantt diagram, useful to have an overview of the tasks and see in a better way the tasks interactions.

These tasks are about project management such as write document, checkpoints, planning. Some tasks started at the same time than the project, such as the planning or the meetings because they are completed throughout the project. We did almost one checkpoint per week with our coordinator, Torben but at start we had a lot of checkpoints about the project ideas, some of them were only between us.

This diagram shows us a lot of information about the project management, when we started to write the different documents, we can clearly see that we didn't work from the end of September to the end of October, no checkpoints.

A detailed version of the planning is available in the **Appendix**.

4.3.2. Android - Gantt

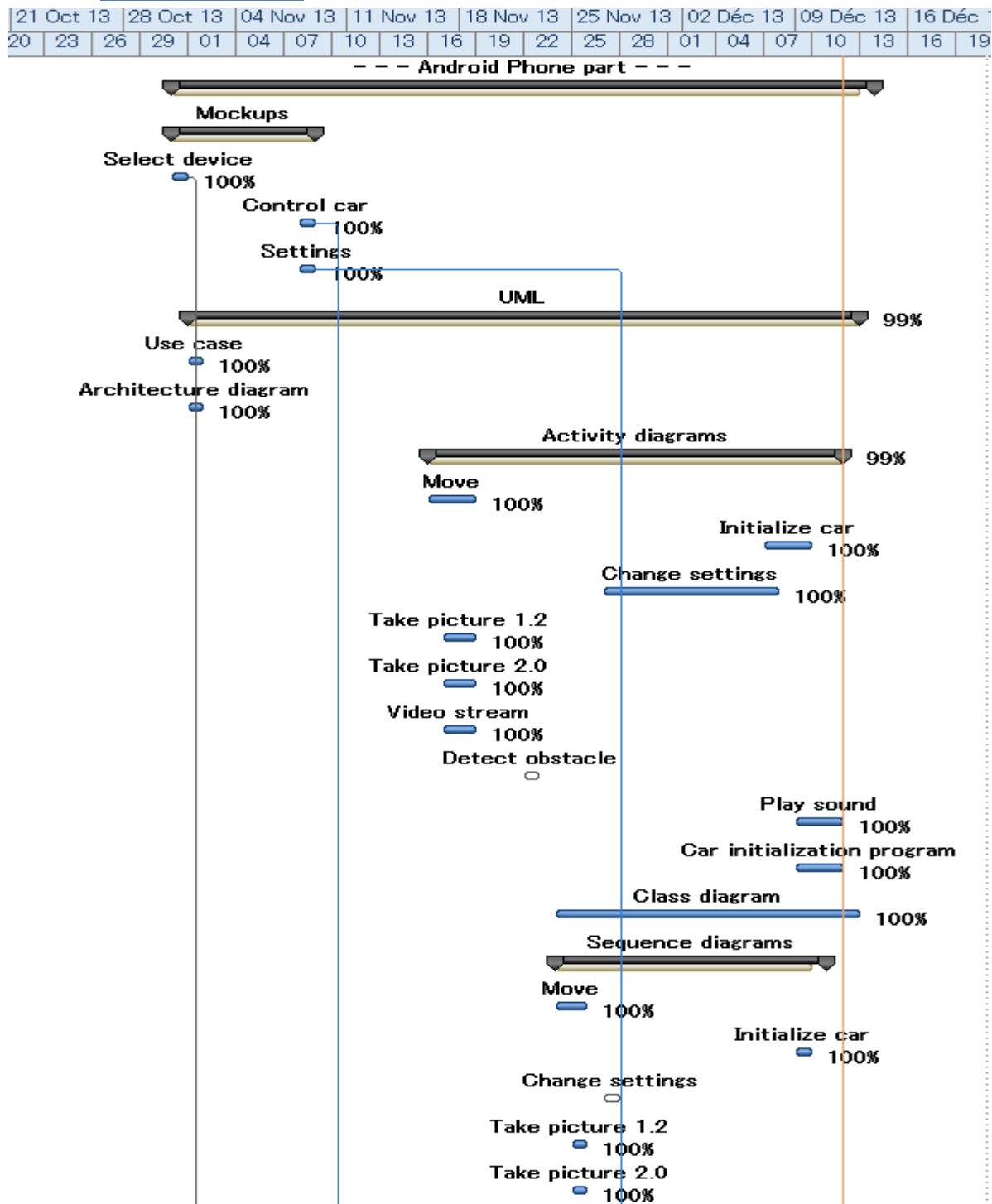


Figure 5 – Planning part #2 – Android (Gantt #1)

All the tasks described belong are about Android phone application part.

We started by create some mock ups to see how we wanted the user interface to be. At the same time we started the UML design with the Use case diagram, the most important one to have an overview of the system functionalities.

Based on this design we completed the mock ups. Then we started the Activity diagrams design to have an idea about how each functionality will work.

Activity diagrams were done twice, the first time they was not really precise, so we did them again once the functionalities were done.

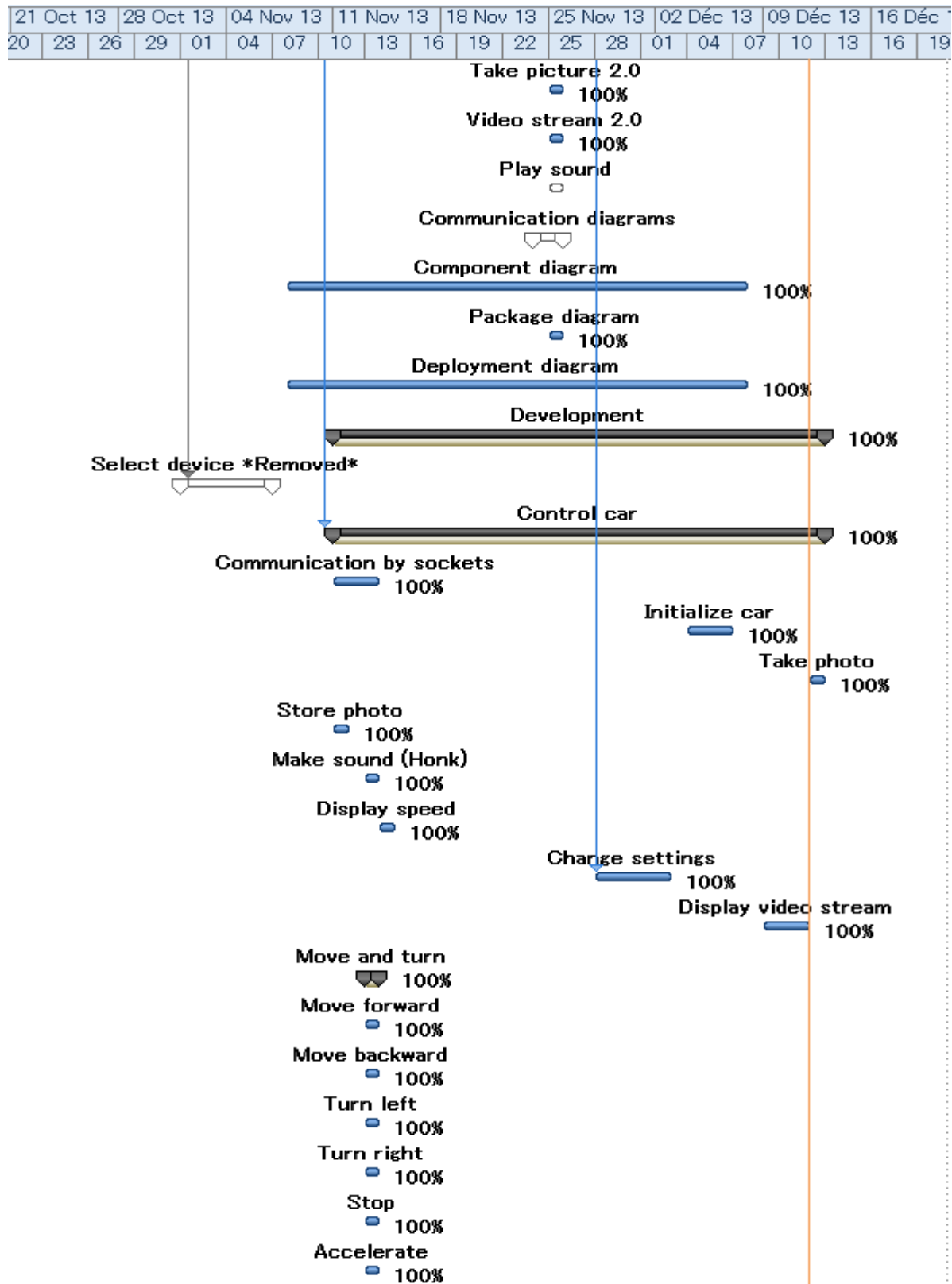


Figure 6 - Planning part #2 – Android (Gantt #2)

We started the Android development by some tasks we removed since, because they actually weren't useful, we did some mistakes about software design, because we didn't understand clearly the way the communication will work between the Android and the Arduino.

Diagrams such as "Move", "Initialize car" and "Change settings" were done twice. Some other diagrams such as "Component diagram" were updated a long time after they were done, following Torben advices.

4.3.3. Car & Arduino - Gantt

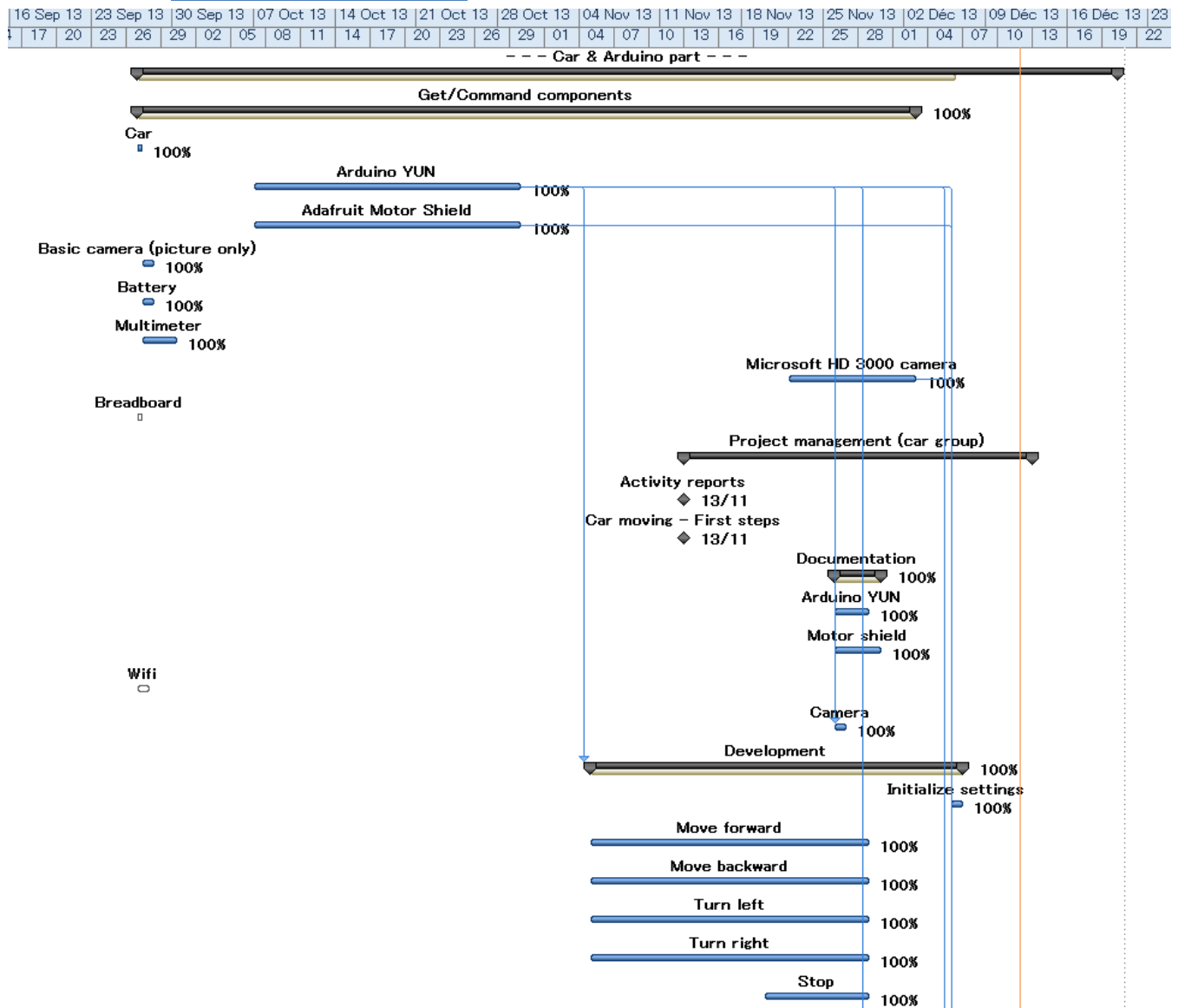


Figure 7 - Planning part #3 – Car & Arduino (Gantt #1)

The part about the Car and the Arduino started by find and buy components, we waited a long time for the first Arduino YUN (17 days).

The development started a few times after we got the Arduino, but we improved many times the program and because of some issues, we waited a long time before make it work.

Almost all the documentation about the Car and the Arduino started end of November.

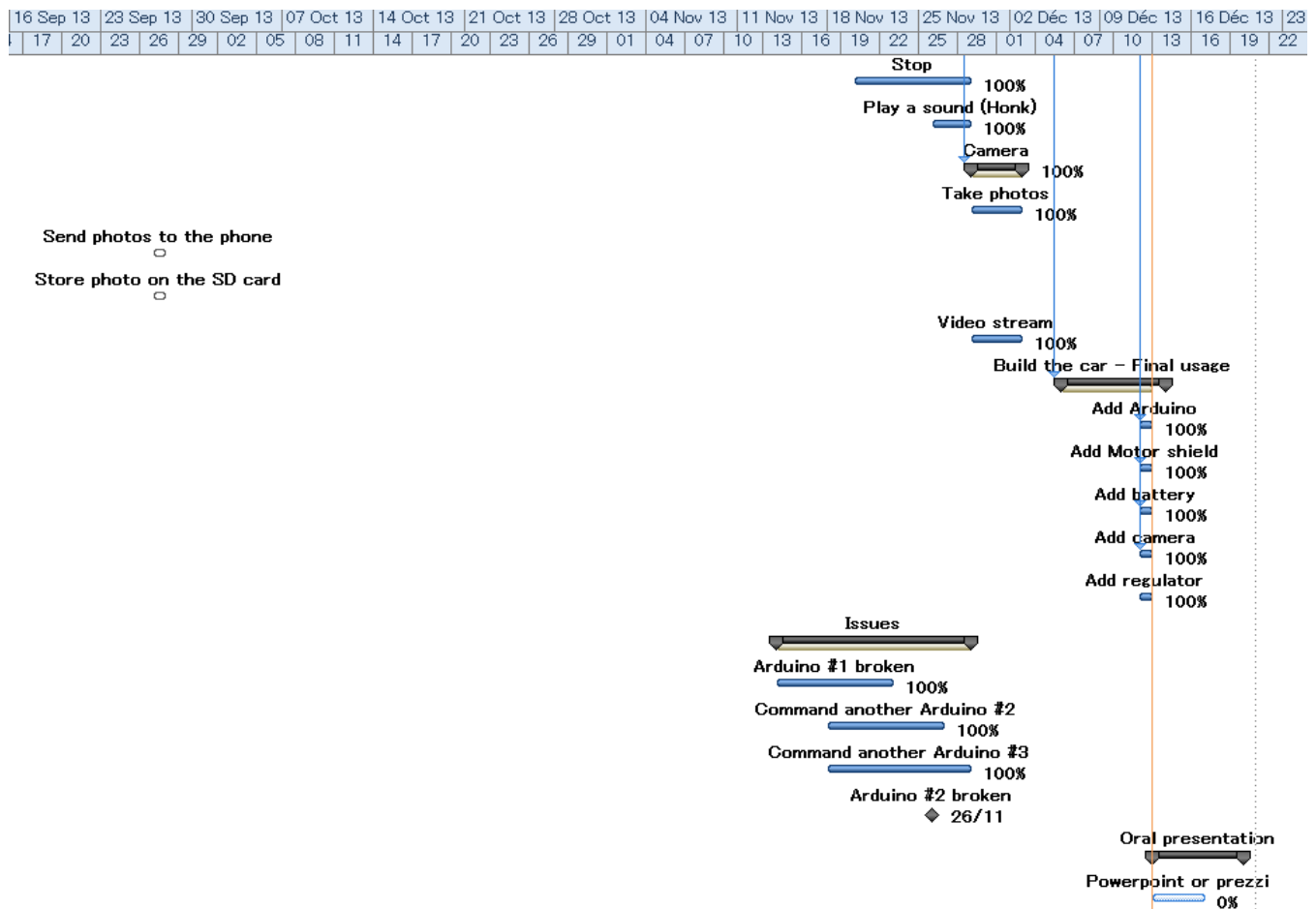


Figure 8 - Planning part #3 – Car & Arduino (Gantt #2)

We got some issues with the Arduino, two of them were broken, and we lost 14 days with these issues, during the meantime we started to write the documentation, we were not able to improve the Arduino program. These issues impacted also the development of the Android application because we weren't able to test it.

Finally, we got our third Arduino two days after broke the second and we were able to work further.

Some tasks such as Send and Store photos was removed, because it was not working the way we thought, it was to the Android application to manage the photos, not to the Arduino.

We planned to make some improvements on the car the last week before the examination

5. Documents summary

5.1. Pre-Analysis

The “Pre-Analysis” document is about our component possibilities and our component choices, how did we choose and with which criteria.

This document explains our choices about the communication, the main board, the motor shield, the camera, the car, the smartphone and the battery.

Finally, our pre-analysis made us choose these components:

1. Arduino YUN board with built-in Wi-Fi module to provide connection between the board and the phone
2. Adafruit motor shield to establish a connection between an Arduino and two motors (responsible for turning left/right and moving forward/backward)
3. Microsoft Lifecam HD-3000 to connect to Arduino board using USB port
4. A car skeleton from Univercity (including two engines, four wheels and a plastic frame with battery box)
5. An Android smartphone
6. Ni-Mh rechargeable battery and a charger for car
7. A set of wires and tools for assembling components
8. Voltage regulator for Arduino YUN
9. Alkaline battery for Arduino YUN

5.2. Requirement specification

The “Requirement specification” document describes the project requirements, it describes all the functionalities to implement for each version of the system and the improvements.

We decided to split the project into 5 different versions, from 1.0 to 2.1. Our minimal goal was the 1.0, we reached it at the end of November.

We decided to use the use case UML diagrams to present the functionalities and for each functionality to use an activity diagram to explain as close as possible how it works.

It also shows our user interfaces and some non-requirement specifications.

Finally, we decided to develop the following functionalities:

1. Move (including turn)
2. Initialize the car
3. Change the settings
4. Take pictures and store them
5. Get a video stream from the camera
6. Play a sound, such as honk, on the car.
7. Detect obstacles and play a sound on the phone.

But we decided later to abandon the last one because we had not the component either the time to implement it.

5.3. Design

The design document describes exactly how the system works. It explains both Android application and car application. We used the 4+1 View model to describe our design about the whole application. We explained in the document more information about 4+1 implementation.

We use **Class** diagram overview and **Class** diagram per package to *explain* each package and its *role*.

We show the most important **Use case** diagrams to *describe* the functionalities we want to build.

We show **Deployment** diagram and **Component** diagram to explain the *communication* between components.

We use **Sequence** diagrams to *explain* how work each functionality, we use this diagram only if it's *interesting* or *complicated*.

All these diagrams are explained using the 4+1 view model for the Android part. We use also **Class** diagram to represent the *Arduino* program even if it's not really a class but more functions linked to each other, we show some diagrams about how work the components we use too.

5.4. Tests

This document is about all the tests we have done during the development on both Android and Arduino applications. It explains every test we did and how we did them. It contains also the date of each tests, for each category such as Unit test, Integration test or User acceptance tests and if they was successful or not.

We use Unit test to test each single functionality while we use Integration tests to test these functionality working with other functionalities. We also use some non-regression test to be sure our new functionalities or updates didn't break them. Finally we use the User acceptance test to test if what we built respects what we designed to build.

6. Conclusion & future

6.1. Planning

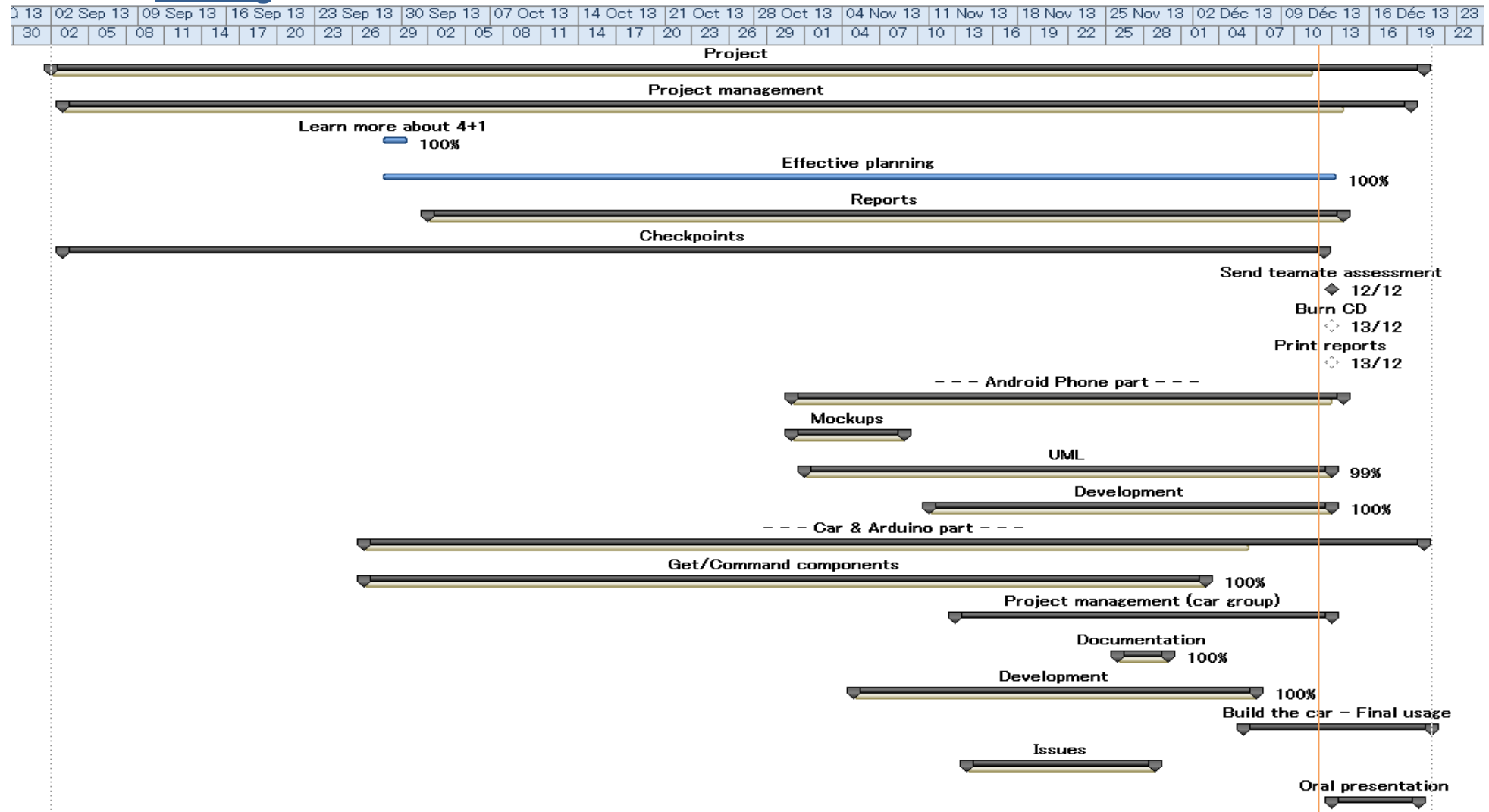


Figure 9 – Planning overview

6.1.1. Planning resume

The project started the 2^d of September and will be over at the 20th December 2013. We started by finding an idea about what we wanted to build during the three first weeks of September.

The project management took place during the entire project, starting with checkpoints then documentation, planning and more.

There is two big parts in the project, respectively Android application and Car/Arduino application. We started by the second with the choice of the components, we got the minimal hardware requirement in the middle of the project, about the end of October.

Once we received the basics components (Arduino especially) we started to work on both applications until the issues with the Arduino, we couldn't work while two-three weeks on the Arduino and we couldn't improve more the Android application without test it. We used this time to start to work on the documentation. Once the issue was fixed we worked again on the applications until the end of the project.

6.2. Results

Our objective was pretty clear, we wanted to reach at least the 1.0 version. We implemented the honk functionality that we planned for the 2.1 version for the 1.0 version, it was actually easier and quicker than we thought in a first place.

What we achieved at the end of the project is exactly the 2.0 version, we are able to move, turn, play a honk sound, take photo and have a video stream on the camera. All the car configuration about motor, video, photo can be customize by the user using the settings. We didn't reach the 2.1 because we didn't designed or buy proximity sensor components, maybe we could without our issues with Arduino, maybe not, mainly depending on the time required to buy and get the components.

6.3. Discussion of results

There are not so much variances between what we wanted to build and what we really built. We designed the system using Use case and Activity diagrams, some of them were wrong, for instance about the Use case diagrams we respected the defined versions, except the "Honk" functionality which was done in 1.0 instead of 2.1. There was more variances about the Activity diagrams because several things was not working at the same way we thought when we designed them such as the communication between Android and Arduino, the camera which includes the photos and the video stream.

We changed our design when we learned how it will really work.

We wasn't following a planning so we don't have planning variance even if the issues with the Arduino was definitively a problem, we could have done better without these issues.

When we take a look to the planning we can see that we lost one month and half waiting for component, this time is half the entire project, a lot of time lost waiting for something. The application could be better without all this time lost.

We reached the 1.0 version the 28th November. We reached the 1.1 version 6th December. We didn't reach the 1.2 version because we preferred have a real video stream instead of a fake stream using pictures. We reached the 2.0 version the 12th October. (The 1.1 version was actually reached before the 6th but we did the user acceptance test the 6th)

6.4. Strengths and weaknesses

6.4.1. Strengths:

- The project works, we are able to do what we wanted.
- We can do more than what we decided at the start, such as accelerate and decelerate.
- The source code for both Android and Arduino is commented, well explained and documented with external documents.
- All the source code and the documentation are available for future usages for other people and free. (Git)
- Good video stream and pictures.
- Android application customizable.
- Car customizable using Android settings with many options.
- Android application well designed with separated classes in different packages.
- Android application usable on different smartphones with different sizes, video stream width and height can be adapted to fill all screens sizes.
- Arduino program could be improved, there is still some space to develop new functionalities on it. (~20% space left)

6.4.2. Weaknesses:

- The product needs two separated power sources.
- Issues to fix on both Android and Arduino programs (threads ghosts, infinite loop, video freeze, etc.)
- Short Wi-Fi range (10-15 meters)
- Really short range with video stream. (~8 meters)
- Troubles to turn because of the Android framework limitations. (Unable to detect some events on the screen)

6.5. Suggested improvements

- New Android activity to show the list of Arduino networks (only Arduino network) to change the network used by the phone from this interface. Particularly useful if there are many networks detected by the phone.
- Refresh the car settings only if the car settings are updated, not each time.
- Use external antenna for greater range.
- Use only one source of power, so use the big battery used by the motor for both motor and Arduino using the regulator.
- Study and fix bugs on the applications.
- Improve the Android application to be able to get the video stream and the car control after lost them (out of range) automatically.
- Improve turn functionality to make it easier to use.
- Add sounds to warn the user on some events such as “out of range”, “video lost”, etc.
- Improve the application with new modules such as proximity detection using sensor.
- Improve Android application with a different way to control the car using sensors instead of buttons and propose the choice to the user between both.

6.6. Conclusion

As project manager, I am proud of what we built, even if we made mistakes and we could have done it better, the results are there and we completed this project with a really good documentation about an interesting application and finally our car works and we should be able to make a demonstration during our oral examination or show a video at least! (If something wrong happened in the meantime...)

Manage a group with people from four different countries is not easy, especially with so few hours together per week and several projects at the same time. We decided to have only one project manager, it's a lot of work, organization and responsibility for a single person, but it's what we decided and it's what we did.

We all learned many things in several parts, hardware or software, even project management, this project helped us, all of us to better understand the communication with hardware and software. We all are used to develop software and we are not used to deal with hardware, it's also really complicated many information and documentation to read, to be aware of. You cannot really break software, it doesn't happen often, but it's really easy to break hardware for instance, we discovered that.

Our supervisor helped us a lot with many advices, the view model 4+1 was his idea, we never used it before, but it's efficient and really useful to show a whole system dealing with hardware and software applications. We also learned many things about UML diagrams used by 4+1. We did a lot of checkpoints with Torben and it was really helpful, have someone available with his experience was really appreciable.

This project lasted long time but it was really interesting and everybody worked on it, the group was interested since the start and we choose quite quickly our subject, and everybody agreed on the subject and was interested by it and it's important, especially when a project lasts three months.

7. Glossary

WCC: Wi-Fi controlled car, the system as a whole.

Linino: Linux OS embedded on the Arduino board.

Arduino YUN: Arduino is the microcontroller used to control the car. We chose the YUN model.

8. References

- **Android:**
 - <http://android.serverbox.ch/?p=1039> (Sockets with Android and Arduino YUN)
 - <https://sites.google.com/site/androidhowto/how-to-1/display-a-web-page> (Video stream from website in Android)
 - <http://stackoverflow.com/questions/3205191/android-and-mjpeg> (Mjpeg format for Android)
 - <https://bitbucket.org/neuralassembly/simplemjpegview> (Source code sample to use Mjpeg view for Android)
 - <https://github.com/Vadorequest/AAD1> (Official Git repository)
- **Arduino:**
 - <http://arduino.cc/>
 - <http://forum.arduino.cc/>
 - <http://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/>

9. Appendix

	<i>Task name</i>	<i>Duration</i>	<i>Start</i>	<i>Finish</i>	<i>Resources</i>	<i>%</i>
2	Project management	78 days	Tue 9/3/13¹	Thu 12/19/13	Ambroise, Szymon	84%
3	Learn more about 4+1	1 day?	Sat 9/28/13	Mon 9/30/13	Ambroise	100%
4	Effective planning	50 days	Sat 9/28/13	Fri 12/6/13	Ambroise	100%
5	Reports	53 days	Wed 10/2/13	Fri 12/13/13		100%
6	Pre analysis	12 days	Wed 11/27/13	Thu 12/12/13	Szymon	100%
7	Requirements specification	52 days	Wed 10/2/13	Thu 12/12/13	Ambroise	100%
8	Project report	30 days?	Mon 11/4/13	Fri 12/13/13	Ambroise	100%
9	Design document	9 days	Mon 12/2/13	Thu 12/12/13	Ambroise	100%
10	Tests document	5 days	Fri 12/6/13	Thu 12/12/13	Anatolii	100%
11	Checkpoints	1 day	Tue 9/3/13	Tue 9/3/13		100%
12	Project's ideas	0 days	Tue 9/3/13	Tue 9/3/13	Torben	100%
13	Project's ideas	0 days	Fri 9/6/13	Fri 9/6/13	Torben	100%
14	Project's ideas - Advanced	0 days	Fri 9/13/13	Fri 9/13/13		100%
15	Functionalities & components	0 days	Tue 9/17/13	Tue 9/17/13		100%
16	Cars & camera	0 days	Fri 9/20/13	Fri 9/20/13	Torben	100%
17	Get cars	0 days	Mon 9/23/13	Mon 9/23/13	Torben	100%
18	Arduino YUN, camera, discuss	0 days	Fri 9/27/13	Fri 9/27/13	Torben	100%
19	Car advancement	0 days	Tue 10/29/13	Tue 10/29/13	Torben	100%
20	Car advancement and RS	0 days	Thu 11/7/13	Thu 11/7/13	Torben	100%
21	Car movement and RS	0 days	Thu 11/14/13	Thu 11/14/13	Torben	100%
22	Conception, RS	0 days	Thu 11/21/13	Thu 11/21/13	Torben	100%
23	Documentation, design document, project report	0 days	Thu 11/28/13	Thu 11/28/13	Torben	100%
24	Project report, design, pre-analysis	0 days	Thu 12/5/13	Thu 12/5/13	Torben	100%
25	Documentation	0 days	Mon 12/9/13	Mon 12/9/13	Torben	100%
26	Documentation	0 days	Tue 12/10/13	Tue 12/10/13	Torben	100%
27	Meeting	0 days	Thu 12/12/13	Thu 12/12/13	Torben	0%
28	Send teammate assessment	0 days	Thu 12/12/13	Thu 12/12/13		
29	Burn CD ²	0 days	Fri 12/13/13	Fri 12/13/13		
30	Print reports	0 days	Fri 12/13/13	Fri 12/13/13		

Figure 10 - Planning part #1 - Project management

¹ MM/dd/YY² Task canceled

	<i>Task name</i>	<i>Duration</i>	<i>Start</i>	<i>Finish</i>	<i>P³</i>	<i>Resources</i>	<i>%</i>
31	Android Phone part	32 days	Thu 10/31/13	Fri 12/13/13			99%
32	Mockups	7 days	Thu 10/31/13	Fri 11/8/13		Ambroise	100%
33	Select device	1 day?	Thu 10/31/13	Thu 10/31/13		Ambroise	100%
34	Control car	1 day?	Fri 11/8/13	Fri 11/8/13		Ambroise	100%
35	Settings	1 day?	Fri 11/8/13	Fri 11/8/13		Ambroise	100%
36	UML	30 days?	Fri 11/1/13	Thu 12/12/13		Alvarro, Ambroise	99%
37	Use case	1 day?	Fri 11/1/13	Fri 11/1/13		Ambroise	100%
38	Architecture diagram	1 day?	Fri 11/1/13	Fri 11/1/13		Ambroise	100%
39	Activity diagrams	18 days?	Sat 11/16/13	Wed 12/11/13		Alvarro, Ambroise, Pierre	99%
40	Move	1 day?	Sat 11/16/13	Mon 11/18/13		Ambroise	100%
41	Initialize car	1 day	Sat 12/7/13	Mon 12/9/13		Ambroise	100%
42	Change settings	8 days	Wed 11/27/13	Sat 12/7/13		Ambroise	100%
43	Take picture 1.2	1 day?	Sun 11/17/13	Mon 11/18/13		Pierre	100%
44	Take picture 2.0	1 day?	Sun 11/17/13	Mon 11/18/13		Pierre	100%
45	Video stream	1 day?	Sun 11/17/13	Mon 11/18/13		Pierre	100%
46	Detect obstacle	1 day?	Fri 11/22/13	Fri 11/22/13			0%
47	Play sound	3 days?	Mon 12/9/13	Wed 12/11/13		Ambroise	100%
48	Car initialization program	3 days?	Mon 12/9/13	Wed 12/11/13		Ambroise	100%
49	Class diagram	14 days?	Sun 11/24/13	Thu 12/12/13		Ambroise	100%
50	Sequence diagrams	13 days	Sun 11/24/13	Tue 12/10/13		Alvarro, Ambroise	99%
51	Move	1 day	Sun 11/24/13	Mon 11/25/13		Ambroise	100%
52	Initialize car	1 day?	Mon 12/9/13	Mon 12/9/13		Ambroise	100%
53	Change settings	1 day?	Wed 11/27/13	Wed 11/27/13		Alvarro	0%
54	Take picture 1.2	1 day?	Mon 11/25/13	Mon 11/25/13		Pierre	100%
55	Take picture 2.0	1 day?	Mon 11/25/13	Mon 11/25/13		Pierre	100%
56	Video stream 2.0	1 day?	Mon 11/25/13	Mon 11/25/13		Pierre, Alvarro, Ambroise	100%
57	Play sound	1 day?	Mon 11/25/13	Mon 11/25/13		Anatolii	0%
58	Communication diagrams	2 days	Sun 11/24/13	Mon 11/25/13		Alvarro, Ambroise	0%
59	Move	1 day?	Mon 11/25/13	Mon 11/25/13		Ambroise	0%
60	Initialize car	1 day?	Mon 11/25/13	Mon 11/25/13		Ambroise	0%
61	Change settings	1 day?	Mon 11/25/13	Mon 11/25/13		Alvarro	0%
62	Component diagram	21 days	Fri 11/8/13	Sat 12/7/13		Ambroise	100%
63	Package diagram	1 day?	Mon 11/25/13	Mon 11/25/13		Ambroise	100%
64	Deployment diagram	21 days?	Fri 11/8/13	Sat 12/7/13		Ambroise	100%
65	Development	24 days?	Mon 11/11/13	Thu 12/12/13			100%
66	Select device *Removed*	4 days?	Fri 11/1/13	Wed 11/6/13	33		0%
67	Enable/Disable	1 day?	Fri 11/1/13	Fri 11/1/13		Ambroise	0%

³ Predecessors, tasks required to be **completed before**, to start.

	WIFI						
68	Get IP of the current network	3 days	Fri 11/1/13	Tue 11/5/13		Ambroise	0%
69	Refresh current network	4 days	Fri 11/1/13	Wed 11/6/13		Ambroise	0%
70	Control car	24 days?	Mon 11/11/13	Thu 12/12/13	34		100%
71	Communication by sockets	3 days	Mon 11/11/13	Wed 11/13/13		Ambroise	100%
72	Initialize car	3 days	Wed 12/4/13	Fri 12/6/13		Ambroise	100%
73	Take photo	1 day?	Thu 12/12/13	Thu 12/12/13		Alvarro	100%
74	Store photo	1 day?	Mon 11/11/13	Mon 11/11/13		Alvarro	100%
75	Make sound (Honk)	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
76	Display speed	1 day?	Thu 11/14/13	Thu 11/14/13		Ambroise	100%
77	Change settings	3 days	Thu 11/28/13	Mon 12/2/13	35	Alvarro	100%
78	Display video stream	3 days	Mon 12/9/13	Wed 12/11/13		Alvarro,Ambroise	100%
79	Move and turn	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
80	Move forward	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
81	Move backward	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
82	Turn left	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
83	Turn right	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
84	Stop	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
85	Accelerate	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%
86	Decelerate	1 day?	Wed 11/13/13	Wed 11/13/13		Ambroise	100%

Figure 11 – Planning part #2 - Android

	<i>Task name</i>	<i>Duration</i>	<i>Start</i>	<i>Finish</i>	<i>P⁴</i>	<i>Resources</i>	<i>%</i>
87	--- Car & Arduino part ---	61 days	Fri 9/27/13	Fri 12/20/13		Szymon,Anatolii,Pierre	75%
88	Get/Command components	47.88 days?	Fri 9/27/13	Mon 12/2/13			100%
89	Car	0.1 days	Fri 9/27/13	Fri 9/27/13			100%
90	Arduino YUN	17 days	Mon 10/7/13	Tue 10/29/13			100%
91	Adafruit Motor Shield	17 days	Mon 10/7/13	Tue 10/29/13			100%
92	Basic camera (picture only)	1 day?	Fri 9/27/13	Sat 9/28/13			100%
93	Battery	1 day?	Fri 9/27/13	Sat 9/28/13			100%
94	Multimeter	2 days	Fri 9/27/13	Mon 9/30/13			100%
95	Microsoft HD 3000 camera	7 days	Fri 11/22/13	Mon 12/2/13			100%
96	Breadboard	0.1 days	Fri 9/27/13	Fri 9/27/13			0%
97	Project management (car group)	22 days	Wed 11/13/13	Thu 12/12/13		Szymon	100%
98	Activity reports	0 days	Wed 11/13/13	Wed 11/13/13		Szymon	100%
99	Car moving - First steps	0 days	Wed 11/13/13	Wed 11/13/13		Anatolii	100%
100	Documentation	4 days?	Tue 11/26/13	Fri 11/29/13			100%
101	Arduino YUN	3 days	Tue 11/26/13	Thu 11/28/13		Anatolii	100%
102	Motor shield	4 days	Tue 11/26/13	Fri 11/29/13		Anatolii	100%
103	Wifi	1 day?	Fri 9/27/13	Fri 9/27/13			0%
104	Camera	1 day?	Tue 11/26/13	Tue 11/26/13	90	Pierre	100%
105	Development	24 days?	Tue 11/5/13	Fri 12/6/13	90		100%
106	Initialize settings	1 day?	Fri 12/6/13	Fri 12/6/13		Ambroise	100%
107	Move forward	18 days	Tue 11/5/13	Thu 11/28/13		Szymon,Ambroise	100%
108	Move backward	18 days	Tue 11/5/13	Thu 11/28/13		Szymon,Ambroise	100%
109	Turn left	18 days	Tue 11/5/13	Thu 11/28/13		Szymon,Ambroise	100%
110	Turn right	18 days	Tue 11/5/13	Thu 11/28/13		Szymon,Ambroise	100%
111	Stop	7 days	Wed 11/20/13	Thu 11/28/13		Szymon,Ambroise	100%

⁴ Predecessors, tasks required to be **completed before**, to start.

112	Play a sound (Honk)	3 days	Tue 11/26/13	Thu 11/28/13		Szymon	100%
113	Camera	2 days	Fri 11/29/13	Mon 12/2/13	90		100%
114	Take photos	2 days	Fri 11/29/13	Mon 12/2/13		Pierre	100%
115	Send photos to the phone	1 day?	Fri 9/27/13	Fri 9/27/13			0%
116	Store photo on the SD card	1 day	Fri 9/27/13	Fri 9/27/13			0%
117	Video stream	2 days	Fri 11/29/13	Mon 12/2/13		Pierre	100%
118	Build the car - Final usage	11 days	Fri 12/6/13	Fri 12/20/13	90		100%
119	Add Arduino	1 days	Tue 12/12/13	Tue 12/12/13	90	Ambroise	100%
120	Add Motor shield	1 days	Tue 12/12/13	Tue 12/12/13	91	Ambroise	100%
121	Add battery	1 days	Tue 12/12/13	Tue 12/12/13	93	Ambroise	100%
122	Add camera	1 days	Tue 12/12/13	Tue 12/12/13	95	Ambroise	100%
123	Add regulator	1 days	Tue 12/12/13	Tue 12/12/133		Ambroise	100%
124	Issues	11 days	Thu 11/14/13	Thu 11/28/13			100%
125	Arduino #1 broken	7 days	Thu 11/14/13	Fri 11/22/13		Szymon	100%
126	Command another Arduino #2	7 days	Mon 11/18/13	Tue 11/26/13		Torben	100%
127	Command another Arduino #3	9 days	Mon 11/18/13	Thu 11/28/13		Ambroise	100%
128	Arduino #2 broken	0 days	Tue 11/26/13	Tue 11/26/13		Szymon	100%
129	Oral presentation	5 days	Fri 12/13/13	Thu 12/19/13			0%
130	Powerpoint or prezzi	2 day	Fri 12/13/13	Mon 12/16/13			0%

Figure 12 - Planning part #3 – Car & Arduino