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 |
| No. pages | TODO |
| Audit |  |
| Date | 30-08-2013 |

Project report

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

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 |

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 5](#_Toc374271467)

[1.1. Purpose 5](#_Toc374271468)

[1.2. Reading instruction 5](#_Toc374271469)

[2. Members 6](#_Toc374271470)

[2.1. Roles 6](#_Toc374271471)

[2.1.1. General 6](#_Toc374271472)

[2.1.2. Android application 6](#_Toc374271473)

[2.1.3. Car application 6](#_Toc374271474)

[3. General design 7](#_Toc374271475)

[3.1. System description 7](#_Toc374271476)

[3.2. Class diagram - Overview 8](#_Toc374271477)

[3.3. Technologies & software 9](#_Toc374271478)

[3.3.1. Technologies 9](#_Toc374271479)

[3.3.1.a. Android 9](#_Toc374271480)

[3.3.1.b. Sockets 9](#_Toc374271481)

[3.3.1.c. Arduino YUN 9](#_Toc374271482)

[3.3.1.d. Adafruit 10](#_Toc374271483)

[3.3.2. Software 10](#_Toc374271484)

[4. Documents summary 11](#_Toc374271485)

[4.1. Pre-Analysis 11](#_Toc374271486)

[4.2. Requirement specification 11](#_Toc374271487)

[4.3. Design 11](#_Toc374271488)

[4.4. Tests 11](#_Toc374271489)

[5. Project management 12](#_Toc374271490)

[5.1. Risk management 12](#_Toc374271491)

[5.1.1. Introduction 12](#_Toc374271492)

[5.1.2. Dropbox 12](#_Toc374271493)

[5.1.3. Git repository 12](#_Toc374271494)

[5.1.4. SugarSync 12](#_Toc374271495)

[5.2. Supervisor checkpoints 13](#_Toc374271496)

[5.3. Gantt diagrams 13](#_Toc374271497)

[5.3.1. Project management - Gantt 14](#_Toc374271498)

[5.3.2. Android - Gantt 15](#_Toc374271499)

[5.3.3. Car & Arduino - Gantt 17](#_Toc374271500)

[6. Conclusion & perspectives 19](#_Toc374271501)

[6.1. Planning 19](#_Toc374271502)

[6.1.1. Planning resume 20](#_Toc374271503)

[6.2. Achievment of objectives 20](#_Toc374271504)

[6.3. Analysis of variances 20](#_Toc374271505)

[6.4. Strengths and weaknesses 20](#_Toc374271506)

[6.4.1. Strengths: 20](#_Toc374271507)

[6.4.2. Weaknesses: 21](#_Toc374271508)

[6.5. Suggested improvements 21](#_Toc374271509)

[6.6. Conclusion 21](#_Toc374271510)

[7. Glossary 22](#_Toc374271511)

[8. References 22](#_Toc374271512)

[9. Appendix 23](#_Toc374271513)

Tables of figures

[Figure 1 - Communication overview 7](#_Toc374271455)

[Figure 2 - The car with the embedded Arduino 7](#_Toc374271456)

[Figure 3 – Android class diagram – Overview 8](#_Toc374271457)

[Figure 4 - Planning part #1 - Project management (Gantt) 14](#_Toc374271458)

[Figure 5 – Planning part #2 – Android (Gantt #1) 15](#_Toc374271459)

[Figure 6 - Planning part #2 – Android (Gantt #2) 16](#_Toc374271460)

[Figure 7 - Planning part #3 – Car & Arduino (Gantt #1) 17](#_Toc374271461)

[Figure 8 - Planning part #3 – Car & Arduino (Gantt #2) 18](#_Toc374271462)

[Figure 9 – Planning overview 19](#_Toc374271463)

[Figure 10 - Planning part #1 - Project management 23](#_Toc374271464)

[Figure 11 – Planning part #2 - Android 24](#_Toc374271465)

[Figure 12 - Planning part #3 – Car & Arduino 26](#_Toc374271466)

TODO

*This page is about the tasks to do, some of these can be done only after a long time so have a page for list them is useful. If you think that some tasks to do are missing, just add them.*

* Check pagination.
* Check spelling by 3 members of the team minimum.
* Check internal “TODO”. (On others pages)
* Check PDF generation. (Document well created, all figures readable, not truncated, etc.)
* Delete this page before generate PDF version or send the final release.
* Regenerate pagination.

***This document belongs to the project manager. If you want to change it, you have to tell to him and explain your changes in the document history part. Be sure to update the last version of this document.***

1. Opening

# 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.

# 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.

1. Members

# Roles

## 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**.

## Android application

There are two members principally working on the Android application:

* Ambroise Dhenain
* Alvaro Garcia

## Car application

There are three members principally working on the Car application:

* Szymon Klepacz
* Anatolli Shakhov
* Pierre Le Texier

1. General design

# 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 exists 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

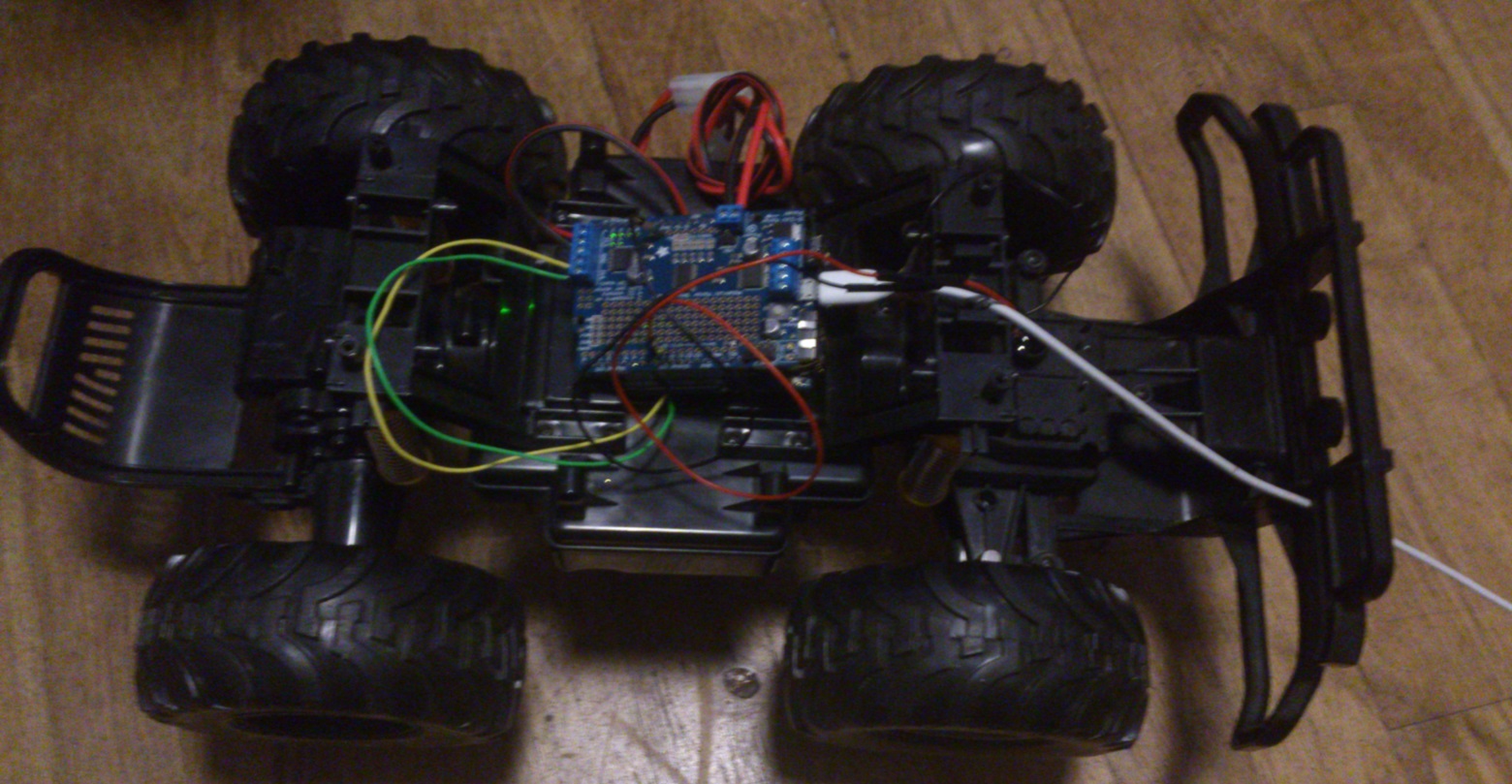


Figure 2 - The car with the embedded Arduino

# 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. TODO Update it.

# Technologies & software

## Technologies

### Android

We use the Android Framework to develop the application. From the beginning, minimum API version was 11 because it was supposed to create an application without any additional work with support libraries. This is why we use only default support library and the framework only – not to overload the code.

When the application was created right in time we were able to add a support for lower level API (now the minimum API level is 4).

Android uses widespread java language, Java SE7.

TODO:{More Info Here? Nope? => That’s wrong, change explanation.}

### 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.

### 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 user data and even some images/videos created with a camera.

#### 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.

### 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.

## 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 U**I: 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.

1. Project management

# Risk management

## 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.

## 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.*

## 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.*

## 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.*

# 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.

# 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 **planning overview**.

## Project management - Gantt

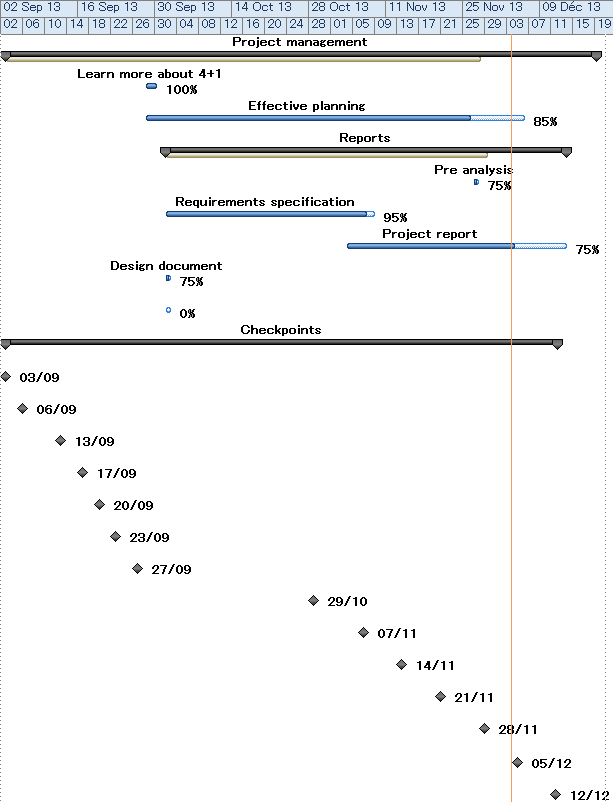


Figure 4 - Planning part #1 - Project management (Gantt)

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.

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

## Android - Gantt

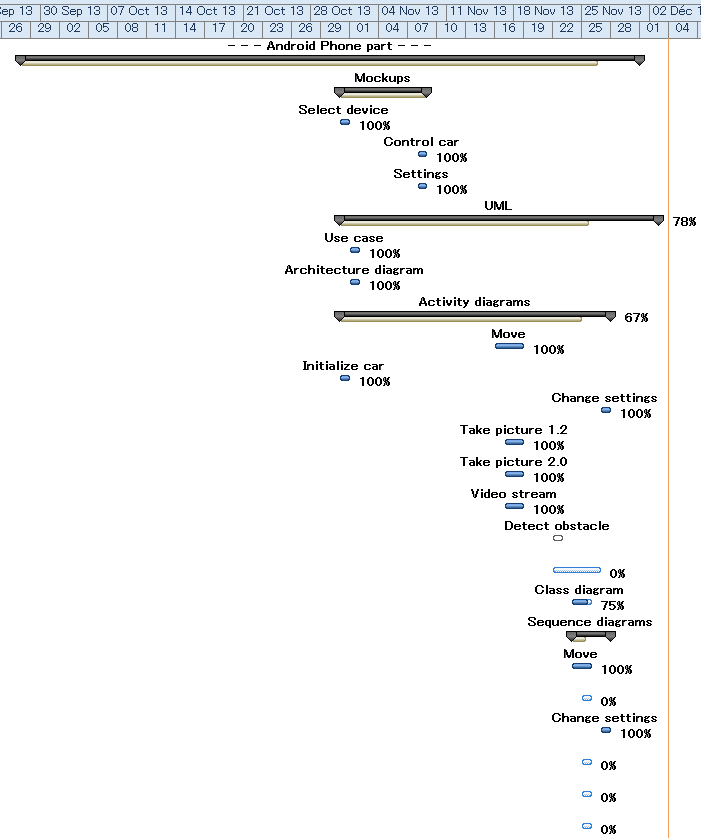


Figure 5 – Planning part #2 – Android (Gantt #1)

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.

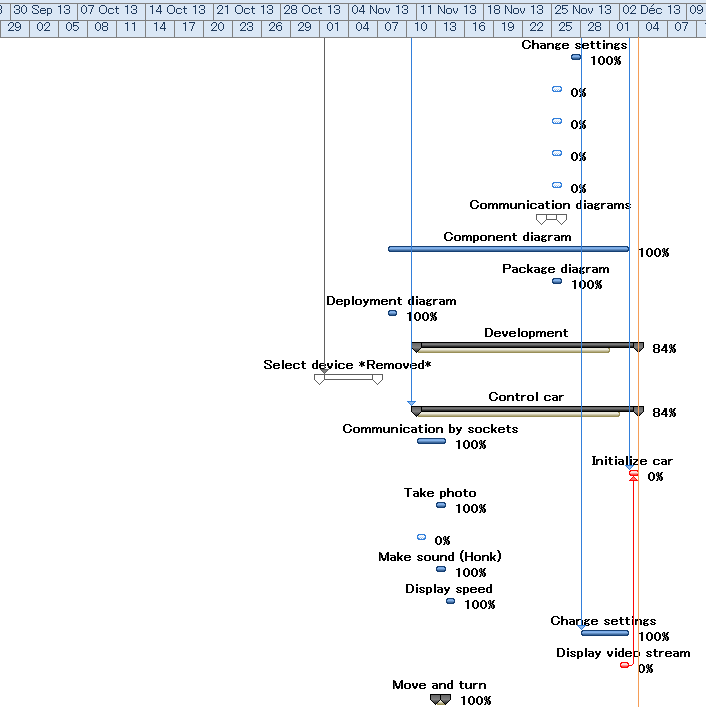


Figure 6 - Planning part #2 – Android (Gantt #2)

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.

## Car & Arduino - Gantt

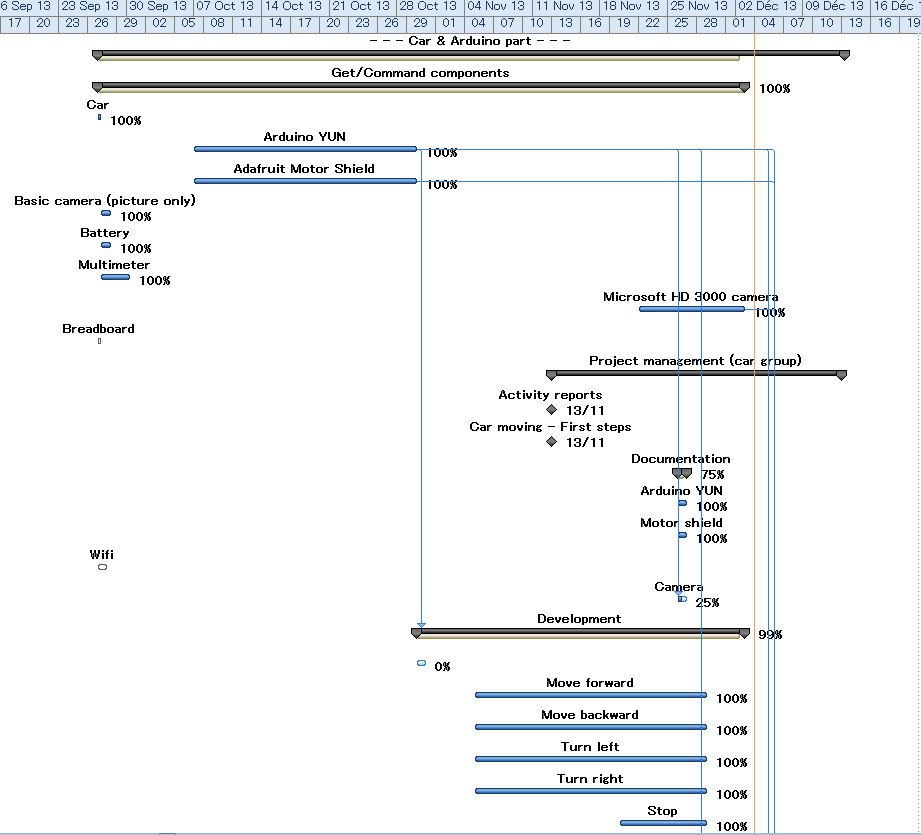


Figure 7 - Planning part #3 – Car & Arduino (Gantt #1)

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.

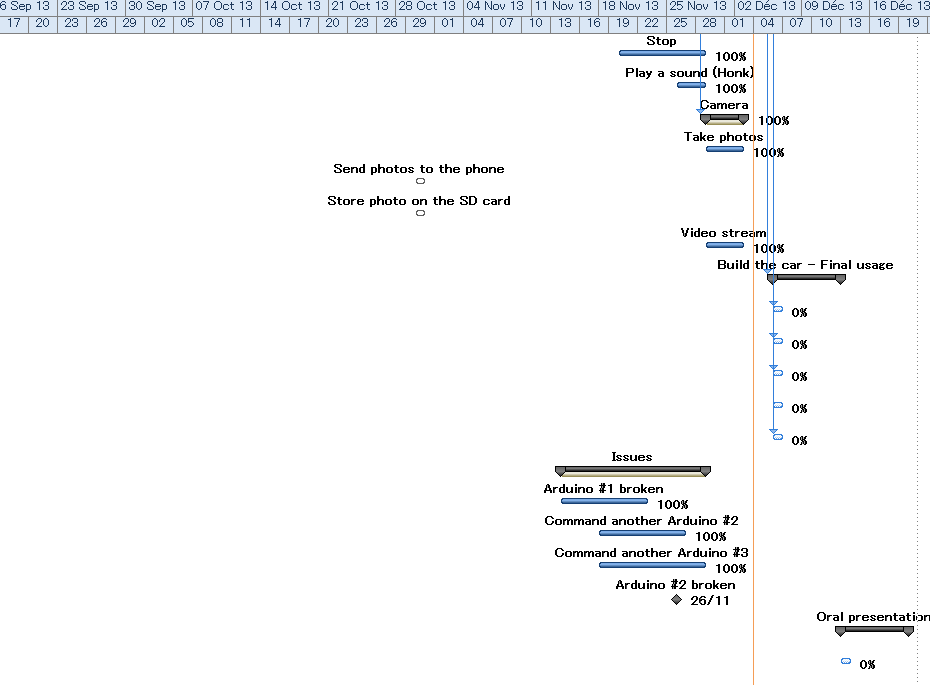


Figure 8 - Planning part #3 – Car & Arduino (Gantt #2)

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.

1. Documents summary

# 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

# 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.

# 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.

# 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.

1. Conclusion & perspectives

# Planning

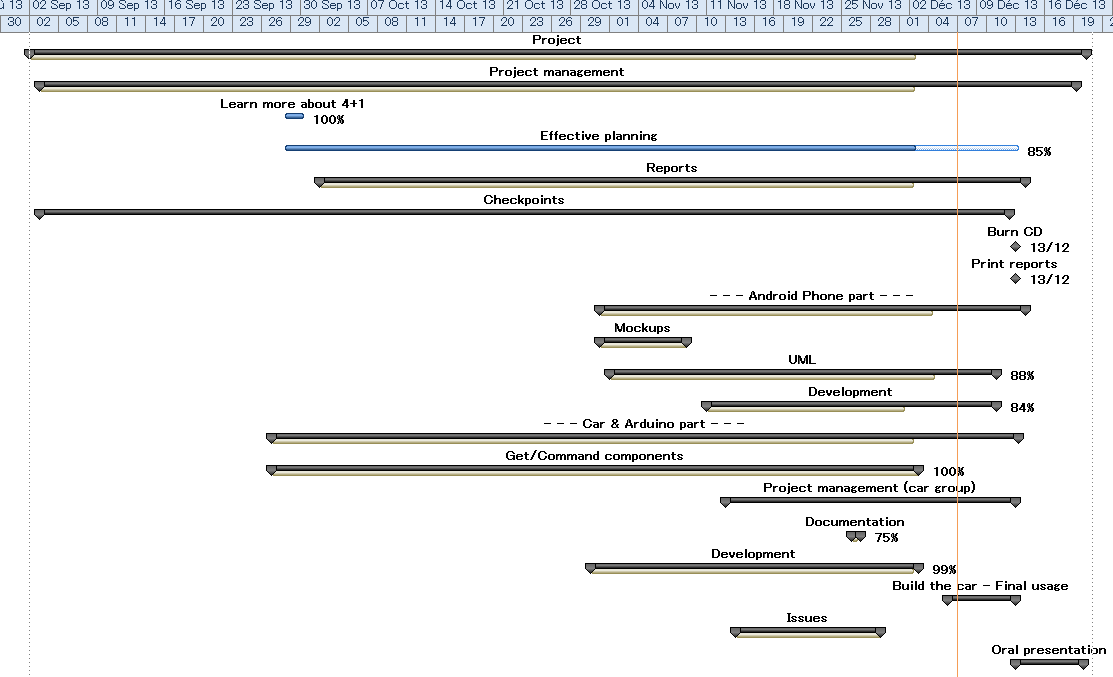


Figure 9 – Planning overview

## Planning resume

The project started the 2d 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.

# Results

Our objective was pretty clear, we wanted to reach at least the 1.0 version. We reached it the 28th November. We also implemented at the same time (1.0) the honk functionality which was planned for the 2.1 version but actually easier and quicker than we thought in a first place.

TODO: Continue.

# 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.

# Strengths and weaknesses

## 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)

## Weaknesses:

* The product needs two separated power sources.

# 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.

# Conclusion

1. 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.

1. 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)
  + <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/>

1. Appendix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Task name** | **Duration** | **Start** | **Finish** | **Resources** | **%** |
| 2 | **Project management** | **78 days** | **Tue 9/3/13[[1]](#footnote-1)** | **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 | 85% |
| 5 | **Reports** | **53 days** | **Wed 10/2/13** | **Fri 12/13/13** |  | **83%** |
| 6 | Pre analysis | 1 day | Wed 11/27/13 | Wed 11/27/13 | Szymon | 75% |
| 7 | Requirements specification | 28 days | Wed 10/2/13 | Fri 11/8/13 | Ambroise | 95% |
| 8 | Project report | 30 days? | Mon 11/4/13 | Fri 12/13/13 | Ambroise | 75% |
| 9 | Design document | 1 day | Wed 10/2/13 | Wed 10/2/13 | Ambroise | 75% |
| 10 | Tests document | 1 day | Wed 10/2/13 | Wed 10/2/13 | Anatolii | 0% |
| 11 | **Checkpoints** | **1 day** | **Tue 9/3/13** | **Tue 9/3/13** |  | **0%** |
| 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 | Meeting | 0 days | Thu 12/5/13 | Thu 12/5/13 | Torben | 0% |
| 25 | Meeting | 0 days | Thu 12/12/13 | Thu 12/12/13 | Torben | 0% |

Figure 10 - Planning part #1 - Project management

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Task name** | **Duration** | **Start** | **Finish** | **P[[2]](#footnote-2)** | **Resources** | **%** |
| 26 | **Android Phone part** | **47 days** | **Sat 9/28/13** | **Sat 11/30/13** |  |  | **81%** |
| 27 | **Mockups** | **7 days** | **Thu 10/31/13** | **Fri 11/8/13** |  | **Ambroise** | **99%** |
| 28 | Select device | 1 day? | Thu 10/31/13 | Thu 10/31/13 |  | Ambroise | 100% |
| 29 | Control car | 1 day? | Fri 11/8/13 | Fri 11/8/13 |  | Ambroise | 100% |
| 30 | Settings | 1 day? | Fri 11/8/13 | Fri 11/8/13 |  | Ambroise | 100% |
| 31 | **UML** | **23 days?** | **Thu 10/31/13** | **Mon 12/2/13** |  | **Alvarro,**  **Ambroise** | **78%** |
| 32 | Use case | 1 day? | Fri 11/1/13 | Fri 11/1/13 |  | Ambroise | 100% |
| 33 | Architecture diagram | 1 day? | Fri 11/1/13 | Fri 11/1/13 |  | Ambroise | 100% |
| 34 | **Activity diagrams** | **20 days?** | **Thu 10/31/13** | **Wed 11/27/13** |  | **Alvarro, Pierre, Ambroise** | **67%** |
| 35 | Move | 1 day? | Sat 11/16/13 | Mon 11/18/13 |  | Ambroise | 100% |
| 36 | Initialize car | 1 day? | Thu 10/31/13 | Thu 10/31/13 |  | Ambroise | 100% |
| 37 | Change settings | 1 day? | Wed 11/27/13 | Wed 11/27/13 |  | Alvarro | 100% |
| 38 | Take picture 1.2 | 1 day? | Sun 11/17/13 | Mon 11/18/13 |  | Pierre | 100% |
| 39 | Take picture 2.0 | 1 day? | Sun 11/17/13 | Mon 11/18/13 |  | Pierre | 100% |
| 40 | Video stream | 1 day? | Sun 11/17/13 | Mon 11/18/13 |  | Pierre | 100% |
| ~~41~~ | ~~Detect obstacle~~[[3]](#footnote-3) | ~~1 day?~~ | ~~Fri 11/22/13~~ | ~~Fri 11/22/13~~ |  |  | ~~0%~~ |
| 42 | Play sound | 3 days? | Fri 11/22/13 | Tue 11/26/13 |  | Anatolii | 0% |
| 43 | Class diagram | 1 day? | Sun 11/24/13 | Mon 11/25/13 |  | Ambroise | 75% |
| 44 | **Sequence diagrams** | **4 days** | **Sun 11/24/13** | **Wed 11/27/13** |  | **Alvarro,**  **Ambroise** | **29%** |
| 45 | Move | 1 day | Sun 11/24/13 | Mon 11/25/13 |  | Ambroise | 100% |
| 46 | Initialize car | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Ambroise | 0% |
| 47 | Change settings | 1 day? | Wed 11/27/13 | Wed 11/27/13 |  | Alvarro | 100% |
| 48 | Take picture 1.2 | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Pierre | 0% |
| 49 | Take picture 2.0 | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Pierre | 0% |
| 50 | Video stream 2.0 | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Pierre | 0% |
| 51 | Play sound | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Anatolii | 0% |
| ~~52~~ | ~~Communication diagrams~~ | ~~2 days~~ | ~~Sun 11/24/13~~ | ~~Mon 11/25/13~~ |  | ~~Alvarro,~~  ~~Ambroise~~ | ~~0%~~ |
| 56 | Component diagram | 17 days? | Fri 11/8/13 | Mon 12/2/13 |  | Ambroise | 100% |
| 57 | Package diagram | 1 day? | Mon 11/25/13 | Mon 11/25/13 |  | Ambroise | 100% |
| 58 | Deployment diagram | 1 day? | Fri 11/8/13 | Fri 11/8/13 |  | Ambroise | 100% |
| 59 | **Development** | **17 days?** | **Mon 11/11/13** | **Tue 12/3/13** |  |  | **84%** |
| ~~60~~ | ~~Select device \*Removed\*~~ | ~~4 days?~~ | ~~Fri 11/1/13~~ | ~~Wed 11/6/13~~ | ~~28~~ |  | ~~0%~~ |
| 64 | **Control car** | **17 days?** | **Mon 11/11/13** | **Tue 12/3/13** | **29** |  | **84%** |
| 65 | Communication by sockets | 3 days | Mon 11/11/13 | Wed 11/13/13 |  | Ambroise | 100% |
| 66 | Initialize car | 1 day? | Tue 12/3/13 | Tue 12/3/13 | 72,36,46 | Ambroise | 0% |
| 67 | Take photo | 1 day? | Wed 11/13/13 | Wed 11/13/13 |  | Ambroise | 100% |
| 68 | Store photo | 1 day? | Mon 11/11/13 | Mon 11/11/13 |  |  | 0% |
| 69 | Make sound (Honk) | 1 day? | Wed 11/13/13 | Wed 11/13/13 |  | Ambroise | 100% |
| 70 | Display speed | 1 day? | Thu 11/14/13 | Thu 11/14/13 |  | Ambroise | 100% |
| 71 | Change settings | 3 days | Thu 11/28/13 | Mon 12/2/13 | 30 | Alvarro | 100% |
| 72 | Display video stream | 1 day | Mon 12/2/13 | Mon 12/2/13 |  | Alvarro | 0% |
| 73 | **Move and turn** | **1 day?** | **Wed 11/13/13** | **Wed 11/13/13** |  | **Ambroise** | **100%** |

Figure 11 – Planning part #2 - Android

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Task name** | **Duration** | **Start** | **Finish** | **P[[4]](#footnote-4)** | **Resources** | **%** |
| 81 | **Car part** | **75 days** | **Mon 9/2/13** | **Thu 12/12/13** |  | **Szymon,**  **Anatolii,**  **Pierre** | **97%** |
| 82 | **Get/Command components** | **48 days?** | **Fri 9/27/13** | **Mon 12/2/13** |  |  | **100%** |
| 83 | Car | 0.1 days | Fri 9/27/13 | Fri 9/27/13 |  | Szymon | 100% |
| 84 | Arduino YUN | 17 days | Mon 10/7/13 | Tue 10/29/13 |  | Szymon | 100% |
| 85 | Adafruit Motor Shield | 17 days | Mon 10/7/13 | Tue 10/29/13 |  | Szymon | 100% |
| 86 | Basic camera (picture only) | 1 day? | Fri 9/27/13 | Sat 9/28/13 |  |  | 100% |
| 87 | Battery | 1 day? | Fri 9/27/13 | Sat 9/28/13 |  | Szymon | 100% |
| 88 | Multimeter | 2 days | Fri 9/27/13 | Mon 9/30/13 |  | Szymon | 100% |
| 89 | Microsoft HD 3000 camera | 14 days | Fri 9/27/13 | Mon 12/2/13 |  |  | 100% |
| 90 | Breadboard | 0.1 days | Fri 9/27/13 | Fri 9/27/13 |  | Szymon | 100% |
| 91 | **Project management (car group)** | **53 days** | **Mon 9/2/13** | **Wed 11/13/13** |  | **Szymon** | **99%** |
| 92 | **Activity reports (weekly)** | **53 days** | **Mon 9/2/13** | **Wed 11/13/13** |  | **Szymon** | **99%** |
| 93 | Waiting for components | 46 days | Mon 9/2/13 | Fri 11/1/13 |  |  | 100% |
| 94 | Car moving - First steps | 0 days | Wed 11/13/13 | Wed 11/13/13 |  | Szymon, Anatolii | 100% |
| 95 | **Documentation** | **1 day?** | **Tue 11/26/13** | **Tue 11/26/13** |  |  | **75%** |
| 96 | Arduino YUN | 1 day? | Tue 11/26/13 | Tue 11/26/13 |  | Anatolii, Szymon | 100% |
| 97 | Motor shield | 1 day? | Tue 11/26/13 | Tue 11/26/13 |  | Szymon, Anatolii | 100% |
| ~~98~~ | ~~Wifi~~ | ~~1 day?~~ | ~~Mon 9/2/13~~ | ~~Mon 9/2/13~~ |  |  | ~~0%~~ |
| 99 | Camera | 1 day? | Tue 11/26/13 | Tue 11/26/13 | 84 | Pierre | 25% |
| 100 | **Development** | **24 days?** | **Wed 10/30/13** | **Mon 12/2/13** | **84** |  | **99%** |
| 101 | Initialize settings | 1 day? | Wed 10/30/13 | Wed 10/30/13 |  | Ambroise | 0% |
| 102 | Move forward | 18 days | Tue 11/5/13 | Thu 11/28/13 |  | Szymon,  Ambroise | 100% |
| 103 | Move backward | 18 days | Tue 11/5/13 | Thu 11/28/13 |  | Szymon,  Ambroise | 100% |
| 104 | Turn left | 18 days | Tue 11/5/13 | Thu 11/28/13 |  | Szymon,  Ambroise | 100% |
| 105 | Turn right | 18 days | Tue 11/5/13 | Thu 11/28/13 |  | Szymon,  Ambroise | 100% |
| 106 | Stop | 7 days | Wed 11/20/13 | Thu 11/28/13 |  | Szymon,  Ambroise | 100% |
| 107 | Play a sound (Honk) | 3 days | Tue 11/26/13 | Thu 11/28/13 |  | Szymon | 100% |
| 108 | **Camera** | **2 days** | **Fri 11/29/13** | **Mon 12/2/13** | **84** |  | **100%** |
| 109 | Take photos | 2 hours | Ven 29/11/13 | Lun 02/12/13 |  | Pierre | 100% |
| 110 | ~~Send photos to the phone~~ | ~~1 jour?~~ | ~~Mer 30/10/13~~ | ~~Mer 30/10/13~~ |  |  | ~~0%~~ |
| 111 | ~~Store photo on the SD card~~ | ~~1 jour?~~ | ~~Mer 30/10/13~~ | ~~Mer 30/10/13~~ |  |  | ~~0%~~ |
| 112 | Video stream | 2 hours | Ven 29/11/13 | Lun 02/12/13 |  | Pierre | 100% |
| 113 | **Build the car - Final usage** | **5 days** | **Fri 12/6/13** | **Thu 12/12/13** | **84** |  | **0%** |
| 114 | Add Arduino | 1 day? | Fri 12/6/13 | Fri 12/6/13 | 84 |  | 0% |
| 115 | Add Motor shield | 1 day? | Fri 12/6/13 | Fri 12/6/13 | 85 |  | 0% |
| 116 | Add battery | 1 day? | Fri 12/6/13 | Fri 12/6/13 | 84 |  | 0% |
| 117 | Add camera | 1 day? | Fri 12/6/13 | Fri 12/6/13 |  |  | 0% |
| 118 | Add regulator | 1 day? | Fri 12/6/13 | Fri 12/6/13 | 89 |  | 0% |
| 119 | **Issues** | **11 days** | **Thu 11/14/13** | **Thu 11/28/13** |  |  | **100%** |
| 120 | Arduino #1 broken | 7 days | Thu 11/14/13 | Fri 11/22/13 |  | Szymon | 100% |
| 121 | Command another Arduino #2 | 7 days | Mon 11/18/13 | Tue 11/26/13 |  | Torben | 100% |
| 122 | Command another Arduino #3 | 9 days | Mon 11/18/13 | Thu 11/28/13 |  | Ambroise | 100% |
| 123 | Arduino #2 broken | 0 days | Tue 11/26/13 | Tue 11/26/13 |  | Szymon | 100% |

Figure 12 - Planning part #3 – Car & Arduino

1. MM/dd/YY [↑](#footnote-ref-1)
2. Predecessors, tasks required to be **completed before**, to start. [↑](#footnote-ref-2)
3. “~~Task name~~”: Tasks removed from the project. [↑](#footnote-ref-3)
4. Predecessors, tasks required to be **completed before**, to start. [↑](#footnote-ref-4)