Wi-Fi Controlled Car

|  |  |
| --- | --- |
| Document title | Test Report |
| Document ID | WCC-TestReport |
| Authors | Ambroise Dhenain  Szymon Klepacz  Anatolii Shakhov  Alvaro Garcia  Pierre Le Texier |
| Supervisor | Torben Gregersen |
| No. pages | TODO |
| Audit |  |
| Date | 07-02-2013 |

Test report

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

Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Audit | Author | Description |
| 06-12-2013 | 0.1 | AS | Template + Information about successful tests. |
| 06-12-2013 | 1.0 | AD | Add software tests part. |
| 08-12-2013 | 1.1 | SzK | Changed test inforamtion. Corrected spelling |
| 09-12-2013 | 1.2 | AS | Updated User Acceptance part, also improved the first part with initial components tests |
| 10-12-2013 | 1.3 | AS | Updated Information about tests (according to supervisors remarks), removed conclusion. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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-TestReport |
| 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

[1.1. Purpose 4](#_Toc374396497)

[1.2. Reading instruction 4](#_Toc374396498)

[3.1. Hardware 6](#_Toc374396499)

[3.2. Software 7](#_Toc374396500)

[3.2.1. Android tests 7](#_Toc374396501)

[3.2.2. Arduino tests 7](#_Toc374396502)

[3.3. JUnit 7](#_Toc374396503)

[4.1. Hardware 8](#_Toc374396504)

[4.2. Software 8](#_Toc374396505)

[4.3. Overall system 9](#_Toc374396506)

[4.3.1. Stage 1 9](#_Toc374396507)

[4.3.2. Stage 2 9](#_Toc374396508)

[4.3.3. Stage 3 9](#_Toc374396509)

[6.1. User acceptance test: 11](#_Toc374396510)

[6.1.1. Version 1.0 11](#_Toc374396511)

[6.1.2. Version 1.1 11](#_Toc374396512)

[6.1.3. Version 1.2 12](#_Toc374396513)

[6.1.4. Version 2.0 12](#_Toc374396514)

[6.1.5. Version 2.1 12](#_Toc374396515)

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 a number of tests within the system environment.

We will show how the system was tested, which bugs and errors occurred during the process and how the system works in a real environment.

# Reading instruction

Chapter 2: Explanation about our way to test.

Chapter 3: Unit tests, one per functionality.

Chapter 4: Integration tests, using several functionality already tested by Unit test to check a bigger environement.

Chapter 5: Non-regression tests, about how we managed the functionalities evolutions.

Chapter 6: Final test, one per version of the application.

1. Test system

During the project we had to work with different hardware and software. The main idea was to assemble a car which can be remotely controlled by a smartphone. We needed to start initial tests before considering on the components. We didn’t know exactly the current and voltage level from the power supply and required by both engines. Each hardware component required special knowledge about its detailed schematics, special tools to provide a proper installation of the components. For the first tests we tried to understand how the different components work separately.

With the software tests situation was a bit different. To achieve the proper tests we had to use software together with the hardware. We could notice bugs and errors in the code from the IDE and we could test how the application works on the phone. The only concern was to test an application with the assembled hardware (the process of assembling took longer time due to some failed tests).

The last steps in testing the system would be create conditions to hold tests in communication between software and hardware in field Tests.

{TODO: Maybe improve or add some additional information}

1. Unit tests

# Hardware

To test the main board for the car we needed special software downloaded from the Arduino web site. With this software we could easily load any program on the board. One of the biggest benefits in it is the availability of test programs (they’re called sketches) which can be loaded on the board to test different functions.

During those initial tests we tried to understand how the board behaves and which instructions change something during the operation process.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Initialization | 07-10-2013 | SK/AS | **YES** |
| Basic test operations | 07-10-2013 | SK/AS | **YES** |
| Wireless Connection | 07-10-2013 | SK/AS | **YES** |

The initial tests passed successfully. We connected the Arduino board to PC, downloaded all the needed software for further development and started test sketches on the board. It ran all the programs successfully. When we turned on the board – we had a possibility to connect to wireless network created by Arduino. When connected we were able to log into the user interface and set up different settings.

The board was working properly out of the box. We didn’t have to add anything else to it – everything needed was to connect the board to the laptop using a USB-MicroUSB wire.

As with the board we had to test the functionality of the engines of the car. These operations required some additional tools from us: digital Multimeter, battery and our car with engines.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Movement/Rotation | 26-09-2013 | Group | **YES** |
| Voltage | 26-09-2013 | Group | **YES** |
| Battery | 26-09-2013 | Group | **YES** |

Those tests passed successfully as well, the battery worked perfectly and it could supply our board. The only problem during tests was the failure of multimeter (it didn’t show the précised values) so we had to change it for a working one model.

After the initial tests we understood the working principle of different parts of our future project. The engines needed a power supply of at least 5V, Arduino board used a mixture between C and C++ languages and we had to load sketches on it to implement different functions, the battery provided 9.7V.

After we disassembled the car, we could not find any information about the engine, but we had a successful voltage and operation tests. If we want the car to move forward, we should connect black and red wires (wires from 1st engine) in a straight way. If we want the car to move backward, we had to switch them. This test showed us the need in motor shield which can provide the functionality to send different voltage.

The initial tests in hardware part succeeded and we could move further for integration testing (deeper and thorough tests to provide information about components integration and inter operation.

# Software

The software tests part contains both Android and Arduino tests, because we can test the Android part separately from the Arduino part, but we have to test both at the same time to test the Arduino part.

All unit tests about Android and Arduino was validated once the functionality was completed, not before.

## Android tests

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Move | 16-11-13 | AD/SzK | **Yes** |
| Initialize car | 04-12-13 | AD/SzK | **Yes** |
| Change settings | 16-11-13 | AD/AG | **Yes** |
| Take picture | 17-11-13 | AD | **Yes** |
| Store picture |  | AG |  |
| Video stream |  | AG |  |
| Honk | 17-11-13 | AD/SzK | **Yes** |

The Android tests could be done without use the Arduino or the car, we needed only the IDE to check the values or display the results. Basically it was to check if the strings sent by socket was correct or not, sometimes it was also to check if the user interface was correctly updated.

## Arduino tests

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Move | 28-11-13 | AD/SzK | **Yes** |
| Initialize car | Not done. | AD/ |  |
| Change settings | Not done. | AD |  |
| Take picture | 01-12-13 | PLT | **Yes** |
| Video stream | 01-12-13 | PLT | **Yes** |
| Honk | 28-11-13 | AD/SzK | **Yes** |

The Arduino tests was the most difficult to do, because of the fact we destroyed two Arduinos, we couldn’t test our functionalities without a real Arduino. We lost almost three weeks to wait to validate Arduino tests. It was difficult because of the Serial connection, for instance we couldn’t display Integers in the console without some conversion, took some time to find out how to do it.

# JUnit

We didn’t use JUnit in our IDE (Android) to run the tests, we didn’t use any framework to do it, our application doesn’t need a test framework, and all tests were done manually by the developer in charge.

1. Integration tests

During these tests we tried to test an interaction between different components. Are they suitable for each other? Can they operate together to provide a proper result? Is the system solid and works as intended?

The new series of tests will show the working of the components and approval of their proper connection.

# Hardware

During the initial tests we tested components, how they operate and what principles are important for them.

At the beginning we soldered headers to our new Adafruit motor shield. Then we plugged it into an Arduino board. The system worked fine, the only one concern was a jumper to provide the voltage for the Arduino board from the shield. When we connect the shield to Arduino – we need to add a power supply to the shield and if we want that supply to give energy to Arduino also, we need to put a small jumper near the power socket.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Initialization | 13-11-2013 | AS/SzK | **YES** |
| Moving | 13-11-2013 | AS/SzK | **YES** |
| Rotation | 13-11-2013 | AS/SzK | **YES** |
| Speed change | 13-11-2013 | AS/SzK | **YES** |
| Moving + Rotation | 13-11-2013 | AS/SzK | **NO** |

First of all – we initialize the board and the whole system (connect wires, plug the power supply, assemble parts). The initialization is the first stage to understand that system turns on and functions. The second stage is to load the program on the board and see if the car starts operating as desired. The rotation tests – is to check how the engine responsible for rotation of the wheels functions. Another important part of it – is to check on how the speed of the car can be changed. At first, the car was moving too fast for us. But the speed can be controlled if the integer value inside the sketch is changed.

All the tests were successful except one. The rotation didn’t work together with the movements. We could start them separately but indeed the program needed some special functions and improvements to provide this important functionality.

The car moved and we needed to wait for the Software part to check the operation of the whole system.

# Software

The software integration tests took place at the same time than the **Arduino tests**, actually when we tested the Arduino we tested also the Android and it was an integration test and a unit test at the same time.

This was true for all tests **except** for the following tests because there were more unit tests about them:

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Take picture |  | AG |  |
| Video stream |  | AG |  |

# Overall system

These tests were held to test main functionality and stability of the whole system. Our car is connected with the phone using a Wi-Fi module and can be tested for its’ main operation. Not all of the test stages passed successfully.

The tests are held in 3 stages, because we had technical issues (burnt boards) during our tests. Each test has important steps inside which show information about different car functions:

Initialization – the first boot, after connecting all the components and plugging in power supply.

Connection – checking the physical connection between the components of the car.

Movement – moving car forward and backward, controlling it by the phone.

Rotation – turning of the wheels in both sides.

Response – the car does actions when user interacts with the Android application.

Wireless Connection – the test to see if the Wireless connection is active between the phone and the board.

Application + Car – test to check if all the functions of the application operate properly and the car executes desired commands.

Play a sound (Honk) – the car has a built-in speaker and can produce sound when user presses the button in the application.

## Stage 1

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Initialization | 14-11-2013 | SzK/AS/AD | **YES** |
| Connection | 14-11-2013 | SzK/AS/AD | **YES** |
| Movement | 14-11-2013 | SzK/AS/AD | **NO** |
| Rotation | 14-11-2013 | SzK/AS/AD | **NO** |
| Response | 14-11-2013 | SzK/AS/AD | **NO** |

We were able to start the system and even connect the phone to it. But the problem appeared when the battery discharged during tests (we thought it caused the problem). When we tried to load the program to test functionality of the board – battery died and Arduino switched off. We changed the battery but the board stopped working properly – all we could see is a single yellow LED flashing all the time. The built-in Wi-Fi module stopped responding to any commands and the hotspot didn’t work. Later on we tried to recover the board, but all attempts failed.

## Stage 2

In a couple of weeks we received a new Arduino and could set the tests again. We arranged a meeting and connected all the components. First tests were done with Arduino powered by a USB cable, and second part was with small jumper on the motor shield to provide a power from the battery.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Initialization | 26-11-2013 | SzK/AS/AD | **YES** |
| Connection | 26-11-2013 | SzK/AS/AD | **YES** |
| Movement | 26-11-2013 | SzK/AS/AD | **YES** |
| Rotation | 26-11-2013 | SzK/AS/AD | **YES** |
| Response | 26-11-2013 | SzK/AS/AD | **YES** |
| Wireless connection | 26-11-2013 | SzK/AS/AD | **NO** |

As with the previous Arduino, this board caught the same system failure – the Linino OS on the board was burnt. The failure occurred when we connected a jumper to the Adafruit shield – we wanted to test the car without wires. The battery should power both shield and the main board, but Arduino YUN was the only one board which does not have a Voltage Regulator. Without it – it may support only 5V and nothing more. The battery gives 9.7V and cannot be used together with this jumper.

## Stage 3

Successfully, our project manager took into account the possibility of another system failure and ordered the Arduino board on his name. It had arrived just a couple of days later after the second Arduino was burnt. The next series of stage started and finally we could test the system all together.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Date | Person | Result |
| Initialization | 28-11-2013 | AD/SzK | **YES** |
| Connection | 28-11-2013 | AD/SzK | **YES** |
| Movement | 28-11-2013 | AD/SzK | **YES** |
| Rotation | 28-11-2013 | AD/SzK | **YES** |
| Response | 28-11-2013 | AD/SzK | **YES** |
| Application + Car | 28-11-2013 | AD/SzK | **YES** |
| Wireless | 28-11-2013 | AD/SzK | ***NA*** |
| Play a sound (Honk) | 28-11-2013 | AD/SzK | **YES** |

During these tests we could successfully check the operation of the application together with the car. The only one trouble occurred was wrong commands (instead of moving back and forward the car was turning right/left and the same about turning). This problem was easily solved by changing a couple of lines in the code. The car was working and the application functioned well. The last step was to add a voltage regulator or another source of power to supply the main board (Stage 3 tests were held with Arduino connected to a laptop by wire).

1. Non-regression tests

Each time we added a new functionality, all previous tested functionalities have to be tested again to be sure the new functionality is not breaking something. These tests are called non-regression tests

We have not so many functionalities so it was easy to test all of these.

1. User acceptance tests

# User acceptance test:

Thes tests are final ones. They are held with the final customer to be sure everything works the way we wrote it on the “*Requirement Specification*” document. Those tests will be done for each version of the entire application.  
We chose our supervisor as tester because he knows well enough the application and the features from the user point of view.

## Version 1.0

This is the very basic version of the car. All the functionalities are cut to the basic movements. It can only move backward and forward. The supervisor has a chance to test it for the first time and to see the further improvements. This is a basic version and test for 1.0 will be used to compare how new functions operate and if the car is still able to move.

|  |  |  |  |
| --- | --- | --- | --- |
| Functions | Date | Tester | Result |
| Movement | 28-11-13 | Torben | **Yes** |
| Basic Funct. | 28-11-13 | Torben |  |
| Attractiveness | 28-11-13 | Torben |  |
| Integrity | 28-11-13 | Torben |  |

## Version 1.1

The second version of the car implements new functions into its’ behavior. Now the car is able to initialize the system, load special settings and run different functions (which user declares). Using settings user can control the operation of the car – regulate the max and minimum speed, rotation of the wheels and stop it when he needs to. Also, the car can be programmed to support executing pre-loaded instructions and execute them with the set up delays.

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Date | Tester | Result |
| Initialization | 28-11-13 | Torben | **Yes** |
| Movement |  | Torben |  |
| Control |  | Torben |  |
| Performance |  |  |  |

This version still has some bugs (Android application) and performance issues (Hardware) and needs to be improved.

## Version 1.2

During these tests we should test the added functionality and improvement of the existing behavior. The car is packed with a camera and should be able to take pictures (approximately every 1-2 seconds) and show them on the phone screen as a whole video stream.

## Version 2.0

This is a major release of the car. Now a lot of minor bugs and issues were fixed. We can test the car and it provides a good performance.

One of the key goals is achieved – we can start recording video and send a real-time video stream to the screen of the smartphone.

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Date | Tester | Result |
| Initialization | 28-11-13 | Torben | **Yes** |
| Movement |  | Torben |  |
| Control |  | Torben |  |
| Overall Performance |  | Torben |  |
| User interface |  | Torben |  |
| Video stream |  | Torben |  |
| Additional features |  | Torben |  |

## Version 2.1

The final pre-release version of the car. Finally it has a video stream on the screen of the phone, great performance and probably the proximity sensor to detect collisions.

The car now has a honk and can play a sound when user sends commands to it.