

UNIVERSITÄT ILMENAU

CAR - ANDROID - ARDUINO

Pflichtenheft Carduinodroid

Bearbeiter:

Behlau, Benjamin
Haueisen, Sven
Lewandowski, Benjamin
Lewandowski, Felix
Schulze, Christian
Strenger, Robin
Thorwirth, Paul
Vogel, Lars

Betreuer:

Dietrich, Thomas

zuletzt geändert am 24. April 2012

Inhaltsverzeichnis

1	Zielbestimmungen	3
1.1	Muss-Kriterien	3
1.1.1	Android-Applikation	3
1.1.2	Java-Programm	3
1.2	Wunsch-Kriterien	4
1.2.1	Android-Applikation	4
1.2.2	Java-Programm	4
1.2.3	Energiemanagement	4
1.2.4	Kostenmanagement	5
1.3	Abgrenzungskriterien	5
1.3.1	Arduino-Controller	5
1.3.2	Verbindung	5
1.3.3	Android-Applikation	5
2	Produkteinsatz	6
2.1	Anwendungsbereiche	6
2.2	Zielgruppen	6
2.3	Betriebsbedingungen	6
3	Produktumgebung	7
3.1	Software	7
3.2	Hardware	7
3.3	Orgware	7
4	Produktfunktion	8
4.1	Android-Funktionen	8
4.2	Javafunktionen	9
4.3	Oberfläche	9
5	Produktleistungen	11
5.1	Benutzbarkeit	11
5.2	Verbindungs-/Bildqualitätsanpassung	11
5.3	Abbruchssicherheit	11
5.4	Portabilität (Desktop/Applikation)	11
5.5	Effizienz	11
5.6	Änderbarkeit und Wartbarkeit	11
5.7	Informationsmeldungen (Klassifiziert)	12
6	Benutzeroberfläche	13
6.1	Android-Applikation	13
6.2	Desktop	13
7	Qualitätszielbestimmungen	15
8	Testszenarien und Testfälle	16
9	Entwicklungsumgebung	18
9.1	Software	18
9.2	Hardware	18

1 Zielbestimmungen

1.1 Muss-Kriterien

1.1.1 Android-Applikation

- Stellt dem Java-Programm alle Funktionen zur Steuerung der Drohne zur Verfügung
- Schnittstelle zwischen dem Computer und den einzelnen Smartphonefunktionen (Ermittlung/ Auslesen und Weiterleitung der GPS-, Verbindungs-, Akku-, Bewegungs-, Kamera und Lautsprecherdaten des Mobiltelefons)
- Logging aller Steuerbefehle, gesendeter Daten sowie Informationen zu Verbindungsvorgängen in eine Datei auf dem Speicher des Android-Mobiltelefons, inklusive einstellbarem Loglevel
- Leichte Erweiterbarkeit (durch einheitliche und modulare Schnittstellen)
- Kompatibilität mit allen Android-fähigen Geräten ab Version 2.3.3
- Verbinden mit dem Computer-Programm über ein lokales Netz
- Steuerung der Motoren (Lenkung, Bewegung, Geschwindigkeitsregelung)
- Kamerabilddaten (Vorder- oder Rückseite) zur Verfügung stellen
- Bei Verbindungsabbruch oder sonstigen steuerungseinschränkenden Fehlern erfolgt automatischer „Systemstopp“ mit anschließenden Wiederverbindungsversuchen
- Automatischer Wechsel bei Netz-/Verbindungsverlust zwischen mobilem Internet und WLAN
- Auslesen der Verbindungsart und -qualität
- Wiedergabe eines Signaltones (z.B. Hupen, Sirene) über die Lautsprecher des Smartphones
- Bereitstellung der Applikation im Google-Playstore Open-Source

1.1.2 Java-Programm

Oberfläche

- Jederzeit zugängliches Hauptmenü mit wichtigen, zentralen Funktionen
- Anzeige der Verbindungsart und -qualität
- Permanente Anzeige der Videoübertragung (Vorder- oder Rückkamera)
- Anzeige wichtiger Informationen (GPS-Daten, Verbindungsart, aktive Kamera, Tastenfeedback etc.)
- Anzeige von Steuerelementen
- Dauerhafte Ausgabe des Kommunikations-Logs

Programmfähigkeit

- Herstellung der Verbindung mit der Android-Applikation
- Logging aller Steuerbefehle, gesendeter Daten sowie Informationen zu Verbindungsvorgängen in eine Datei auf dem Speicher des Computers, inklusive einstellbarem Loglevel (Kommunikations-Log)
- Verarbeitung der Eingabedaten
- Übertragung der Steuerdaten an die Android-Applikation
- Empfangen und Darstellen der Kamerabildraten
- Informationsmeldungen bei beispielsweise Verbindungsabbrüchen oder Bildqualitätsänderungen
- Empfangen aller Statusdaten (GPS, Akku etc.)
- Manuelle Regulierung der Bildauflösung/-qualität
- Steuerung der Tonausgabe am Smartphone
- Bereitstellung des Programms Online als Open-Source

1.2 Wunsch-Kriterien

1.2.1 Android-Applikation

- Automatische Regulierung der Bildauflösung/-qualität anhand der Verbindungsqualität
- Erweiterung des Funktionsumfanges (bspw. Sensorik/ Aktorik Mobiltelefon: Lage, Mikrofon, Zoom)
- Bereitstellung des Blitzes als Beleuchtung

1.2.2 Java-Programm

Oberfläche

- Möglichkeit der Verarbeitung von GPS-Daten zur Kartenansicht
- Anzeige der Geschwindigkeit anhand der GPS-Daten
- Anzeige der verbleibenden Akkulaufzeit

Programmfähigkeit

- Leichte Erweiterbarkeit des Funktionsumfanges
- Speicherung der GPS-Daten (Wegstrecke)

1.2.3 Energiemanagement

- Schonung der Akkuleistung durch sinnvolle, effiziente Prozessoptimierung und intelligentes Batteriemanagement (Display abschalten)

1.2.4 Kostenmanagement

- Problematik zwischen Übertragungsqualität und Volumenverbrauch (mobiles Internet) optimieren

1.3 Abgrenzungskriterien

1.3.1 Arduino-Controller

- Der Arduino-Controller wird vom Projektbetreuer mit allen benötigten Steuerbefehlen bereitgestellt
- Eine Verbindung zwischen Arduino-Controller und ferngesteuertem Modellfahrzeug ist bereits gegeben

1.3.2 Verbindung

- Der Verbindungsaufbau muss ausschließlich funktionieren, wenn sich sowohl Computer als auch Smartphone im gleichen Netzwerk befinden

1.3.3 Android-Applikation

- Bei der Entwicklung einer Bedienoberfläche muss maximal die Beenden-Funktion implementiert sein (Energieeffizienz)

2 Produkteinsatz

2.1 Anwendungsbereiche

Carduinodroid dient der Fernsteuerung eines handelsüblichen Modellautos mittels eines Android-fähigen Mobiltelefons. Erkundungen und Informationsbeschaffungen sollen aus der Sicht einer Drohne ermöglicht werden. Besonders der Einsatz in schwer zugänglichem und unbekanntem Terrain aus sicherer Entfernung kann auf diese Weise realisiert werden.

2.2 Zielgruppen

Aufgrund einer relativ einfachen Handhabung ist Carduinodroid für jeden Benutzer geeignet, der Bedarf und Freude an drohnenähnlichen Vehikeln hat. Dies schließt neben Einrichtungen für Forschung und Lehre auch Privatanwender aller Altersgruppen ein.

2.3 Betriebsbedingungen

Die Betriebsdauer wird rein durch die Akkuleistung und Verschleißteile der Hardware (Fahrzeug, Mikrocontroller und Mobiltelefon) bestimmt. Sowohl das Computerprogramm, als auch die Android-Applikation sind wartungsfrei.

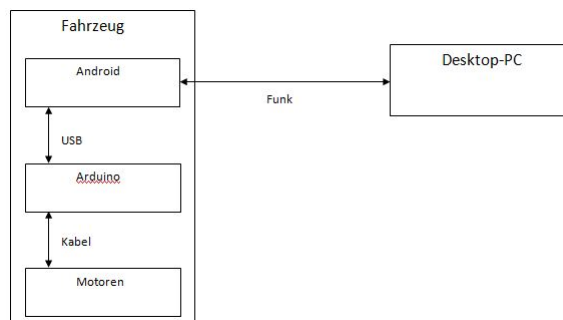


Abbildung 1: Komponentenaufbau

3 Produktumgebung

Das Produkt ist weitgehend unabhängig vom Betriebssystem, sofern folgende Produktumgebung vorhanden ist.

3.1 Software

Android-Applikation

- Android Version 2.3.3 (Gingerbread) und höher

Java-Programm

- Java 32Bit 1.7

3.2 Hardware

Android-Applikation

- Internet-/netzwerkfähiges Handy

Java-Programm

- Internetfähiger Rechner
- Rechner muss die Kriterien der Software erfüllen
- Ausreichend Rechen- und Festplattenkapazität

3.3 Orgware

- Gewährleistung einer permanenten Internetanbindung
- Vorhandensein einer stehenden Verbindung zwischen Mobiltelefon und Computer

4 Produktfunktion

Die Steuerung der Drohne steht in direkten Zusammenhang mit einer stehenden Netzwerkverbindung zwischen dem Android-fähigen Mobiltelefon und Computer.

4.1 Android-Funktionen

- /A01/ *Verbinden*: Das Smartphone muss in der Lage sein sich entweder über das WLAN oder über das mobile Internet zu verbinden um so die Kommunikation mit dem Computer herzustellen.
- /A02/ *Übermitteln*: Das Smartphone soll zur Steuerung des Fahrzeuges über den USB-Port vorgegebene Befehle an den Arduino-Mikrocontroller senden können. Der Mikrocontroller wandelt diese Befehle in elektrische Impulse um und spricht so die Motoren des Autos an. Die Applikation muss dazu Signale, die sie vom Computer erhält, in entsprechende Steuerungsbefehle umwandeln können.
- /A03/ *Verbindungsqualität ermitteln*: Die derzeitige Verbindungsart wird ausgelesen und an den Computer übermittelt.
- /A04/ *Systemstopp*: Beim Verlust der Verbindung zum Computer oder der Kamerafunktion müssen die Motoren des Autos gestoppt werden.
- /A05/ *Zugreifen*: Der Zugriff auf verschiedene Androidfunktionen und anschließendes Übermitteln an den Desktop-PC muss zu jeder Zeit gewährleistet sein.
- /A051/ *GPS*: Die GPS-Daten sind zu ermitteln.
- /A052/ *Kamera*: Die Applikation muss Zugriff auf die Kameras des Mobiltelefons besitzen und je nach Einstellung am Java-Programm entweder die Bilder der Front- oder der Rückkamera auslesen. Außerdem müssen unterschiedliche Qualitätsstufen zu wählen sein. Dazu zählt auch Einzelbildübertragung (in Zeitabständen Fotos erstellen). Die Wahl der Qualitätsstufe soll entweder am Java-Programm eingestellt werden oder von der Verbindungsgeschwindigkeit zum Internet abhängen. Die aufgenommenen Bilder sollen permanent live an den Computer übertragen werden.
 - /A0521/ *Qualität*: Die Bildqualität muss je nach Verbindungsqualität variiert werden können.
 - /A0522/ *Umschalten*: Das Umschalten zwischen beiden Kameras soll möglich sein.
- /A053/ *Lautsprecher*: Um einen eingespeicherten Signalton (z.B. Hupe, Sirene) wiederzugeben, muss die Applikation Zugriff auf die Lautsprecher des Mobiltelefons haben.
- /A054/ *Beleuchtung*: Ein Zugriff auf das Blitzlicht, falls vorhanden, soll ermöglicht werden. (optional)

/A055/ *Mikrofon*: Falls am Java-Programm eingestellt, soll die Applikation das Mikrofon des Mobiltelefons einschalten und Töne aufnehmen können. Diese werden dann, wie die Bilder, live übertragen. Fällt der Ton aus oder ist er abgeschaltet, kann das Fahrzeug weiterhin fahren. (optional)

/A0551/ *Qualität*: Die Soundqualität kann angepasst werden.

4.2 Javafunktionen

Dieser Bereich umfasst die Bereitstellung eines gewissen Funktionsumfangs, der den korrekten Ablauf aller nicht sichtbaren Aufgaben mittels Java gewährleistet.

/J01/ *Verbinden*: Das Programm muss das Android-Mobiltelefon über WLAN oder mobiles Internet finden.

/J02/ *Empfangen*: Nachdem eine funktionstüchtige Internetverbindung hergestellt wurde, muss das Programm erfolgreich die Daten, die via Android übermittelt werden, empfangen.

/J03/ *Verarbeiten*: Die empfangenen Daten sollen ausgewertet werden, d.h. Bild- und Ton- sowie GPS-Daten stehen dem Benutzer zur Verfügung.

/J04/ *Übermitteln*: Die eingegebenen Daten, wie Steuerung, werden an die Android-Applikation übermittelt.

/J05/ *Wechseln*: Nach entsprechender Eingabe wird das Bild zwischen Vorder- und Rückkamera gewechselt.

/J06/ *Informationsmeldungen*: Bei auftretenden Fehlern oder sonstigen Ereignissen werden detaillierte Hinweise angezeigt.

4.3 Oberfläche

/O01/ *Menü*: Dem User wird ein übersichtliches Menü zur Verfügung gestellt.

/O011/ *Beenden*: Das Programm wird bei Anforderung geschlossen.

/O012/ *Verbinden*: Eine Verbindung zum Mobiltelefon wird durch Eingabe der lokalen IP des Smartphones hergestellt.

/O013/ *Einstellung*: Wichtige Einstellungen bezüglich der Bildqualität sowie weitere Änderungen können hier vorgenommen werden.

/O02/ *GPS-Position anzeigen*: Zeigt dem Nutzer die GPS-Position des Autos auf einer Landkarte an. (optional)

/O03/ *Bild anzeigen*: Das Java-Programm muss permanent die empfangenen Kamerabilder darstellen können. Dazu dient ein Schalter mit dem man zwischen Front- und Rückkamera wechseln kann. Weiterhin soll es Optionen geben um die Qualität der Bilder einstellen zu können. Ist die Kamera ausgeschaltet oder die Internetverbindung unterbrochen, soll eine Warnmeldung auf dem Bildschirm ausgegeben und, sofern noch möglich, der Befehl zum „Systemstopp“ an das Smartphone gesendet werden.

- /O031/ *Verbindung verloren*: Es wird eine Fehlermeldung ausgegeben und Wiederverbindungsversuche werden unternommen.
- /O032/ *Licht*: Ist das Bild durch eine dunkle Umgebung nicht nutzbar, soll ein Schalter das Blitzlicht des Mobiltelefons anschalten und (dauerhaft) als Beleuchtung dienen. (optional)
- /O04/ *Geschwindigkeit*: Der Nutzer kann die Geschwindigkeit des Fahrzeuges regulieren.
- /O05/ *Richtung*: Die Richtung des Fahrzeugs kann durch Betätigen der Pfeiltasten geändert werden.
- /O06/ *Informationsbox*: Die Informationsbox setzt den User über GPS-Koordinaten, Verbindungsart, verbleibende Akkulaufzeit, verwendete Kamera und Tastenfeedback in Kenntnis.
- /O07/ *Signalton*: Ein Schalter zur Ausgabe eines Signaltons steht dem Anwender zur Verfügung.

5 Produktleistungen

5.1 Benutzbarkeit

Das Produkt richtet sich an alle Personen, die Besitzer eines Android-Smartphones sind und sich für ferngesteuerte Modellautos/ Drohnen interessieren. Die Benutzerfreundlichkeit wird durch eine grafische Oberfläche, auf der alle Funktionen und Einstellungen auf einem Blick bzw. mit einem Klick einsehbar sind (auch beenden etc.), gewährleistet. Ein Hilfesystem wird nicht nötig sein, da die Oberfläche selbsterklärend ist. Die Drohnensteuerung soll intuitiv erfolgen.

5.2 Verbindungs-/Bildqualitätsanpassung

Die nahtlose Anpassung bei Änderung des Up-/Downstreams soll besonders beim Wechsel zwischen mobilem Internet und WLAN beachtet werden. Die Verbindung wird grundsätzlich über ein lokales Netzwerk aufgebaut, in dem Zusammenhang besteht keine Beachtung der Änderung der IP-Adresse. Zudem muss der Neuaufbau der Übertragung in kürzester Zeit gewährleistet sein. Weiterhin muss die Möglichkeit bestehen die Qualität der Kamerabilder manuell am Computer anzupassen (Auflösung etc.).

5.3 Abbruchssicherheit

Eine Verbindungssteuerung, die für Technik und Gesellschaft ungefährlich ist, muss gewährleistet sein. Besonders bei Verbindungsabbruch muss die Drohne entsprechend mit einem „Systemstopp“ reagieren.

5.4 Portabilität (Desktop/Applikation)

Das Computerprogramm soll auf allen Geräten mit Java Runtime Environment und Internetanbindung, die Smartphone Applikation auf allen Geräten mit Android 2.3.3 Gingerbread oder höher lauffähig sein.

5.5 Effizienz

Das Energiemanagement bei der heutigen Generation von Smartphones stellt die wichtigste Aufgabe in Mobilität und Ausdauer dar. Besonders die Bildschirmregulierung muss in jeglicher Nutzungsphase durchdacht arbeiten, schlussfolgernd bis hin zu deren Abschaltung während des Betriebes. Die Umschaltung zwischen WLAN und mobilem Internet erfolgt intern im Betriebssystem, wobei eine Bevorzugung des drahtlosen Netzwerks vorhanden ist.

5.6 Änderbarkeit und Wartbarkeit

Voraussetzung für eine gute Wartung und Änderbarkeit ist das Vorhandensein einer guten Dokumentation sowie das objektorientierte Programmierparadigma (Klassenaufbau). Dazu wird der Quelltext beispielsweise mit javadoc oder ähnlichem dokumentiert. Eine Beschreibung zum Vorgehen beim Erweitern um neue Funktionen wird dem Produkt beigelegt.

5.7 Informationsmeldungen (Klassifiziert)

Anhand typisch auftretender Probleme, beispielsweise in der Kommunikation der verbundenen Systeme sowie bei Änderung der Verbindungsart, folglich auch Bildqualität, sollen Warnhinweise den Benutzer über Funktionseinschränkungen oder -fehler in Kenntnis setzen.

6 Benutzeroberfläche

6.1 Android-Applikation

- Im besten Fall soll die Android-Oberfläche aus Gründen der Energieeffizienz nicht vorhanden sein. Lediglich ein Warnhinweis, der den Verbindungsabbruch signalisiert, sowie ein Beenden-Button sollen integriert sein.

6.2 Desktop

- Die Anpassung der Oberfläche findet anhand der Kamerabildverhältnisse (hochkant) statt. Eine Menüleiste mit den Punkten Verbindungsaufbau sowie -abbruch, Einstellungen und Beenden stellen die Grundfunktionen zur Verfügung. Die Oberfläche wird zum großen Teil durch das Kamerabild ausgefüllt. Die freien Flächen werden durch Informationen über GPS-Daten, Verbindungs- und Bildqualität sowie einem Richtungstastenfeedback vervollständigt.

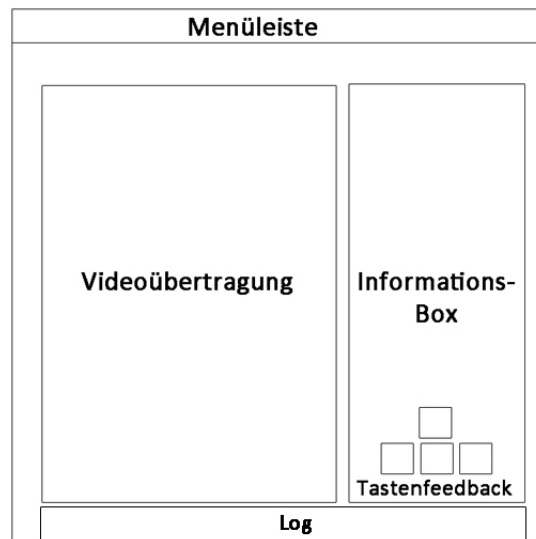


Abbildung 2: Möglicher Oberflächenaufbau

- Eine dynamische Anpassung der Oberfläche ist vorzufinden, wobei eine Mindestgröße festgestellt und -gelegt werden muss, um eine optimale Steuerung sowie Erkennung auf dem empfangenen Bildmaterial zu gewährleisten.
- Eine Möglichkeit zur manuellen Einstellung der Bildqualität muss gegeben sein.
- Die Integration der Muss-/Wunschkriterien Blitzlicht als Leuchtquelle, Kamerawechsel, Ein-/Ausschalter für Tonaufnahmen, verbleibende Akkulaufzeit, Möglichkeiten der Geschwindigkeitsregulierung, Ausgabe eines Signaltones sowie die Positionsbestimmung per GPS-Daten auf Kartenansicht müssen per Schaltfläche anhand der festgelegten Aufteilung in

die Reihe der Informationen/Steuerung aufgenommen werden. Wobei eine abschließende, intuitive Anordnung über die Gewohnheiten erst mit der Oberflächenerstellung erfolgt.

7 Qualitätszielbestimmungen

	Sehr wichtig	Wichtig	Weniger wichtig
Benutzerfreundlichkeit		x	
Effizienz		x	
Fehlertoleranz	x		
Kompatibilität		x	
Korrektheit	x		
Portierbarkeit			x
Robustheit		x	
Sicherheit	x		
Zuverlässigkeit	x		

8 Testszenarien und Testfälle

- /TA01/+ /TJ01/+ /TO012/** Das Java Programm wird gestartet. Über das Menü wird eine Verbindung mit dem Mobiltelefon hergestellt.
- /TA03/** Es wird im Javaprogramm überprüft, ob die Anzeige für die Verbindungsqualität funktioniert. Dazu wird das Carduinodroid durch mehrere Gebiete verschiedener Netzarten navigiert.
- /TA041/+ /TO031/** Der Computer wird vom Internet getrennt. Das Carduinodroid hat mit einem „Systemstopp“ zu reagieren.
- /TA042/** Der Verlust der Kamerafunktion wird dadurch simuliert, dass die Android-Applikation minimiert und dann eine andere Anwendung gestartet wird, die ebenfalls die Kamera benutzt. Das Carduinodroid hat mit einem „Systemstopp“ zu reagieren. Während dieses Tests wird das Auto in der Luft gehalten.
- /TA05/** In der Java-Software werden auf verschiedene Funktionen der Android-Applikation zugegriffen und deren Reaktion überprüft.
- /TA051/+ /TO02/** Am PC werden in der Informationsbox die GPS-Daten ausgelesen und auf Aktualisierung überprüft. Weiterhin wird der Button zur Anzeige der GPS-Koordinaten auf einer Karte betätigt. Danach wird die Kartenanzeige überprüft.
- /TA052/+ /TJ05/** Durch Betätigen des dafür vorgesehenen Buttons wird zwischen Front- und Rückkamera umgeschaltet.
- /TA0521/+ /TO013/+ /TO03/** Im Computerprogramm wird die Bildqualität geändert. Es wird kontrolliert, ob sich jene entsprechend den Vorgaben anpasst.
- /TA0522/+ /TO03/** Im Computerprogramm wird zwischen den beiden Kameras gewechselt. Die Änderung des wiedergegebenen Bildes wird überprüft.
- /TA053/+ /TO07/** Durch Betätigen des „Signalton“-Buttons wird ein Ton vom Mobiltelefon über dessen Lautsprecher abgespielt.
- /TA054/+ /TO03/** Nachdem das Carduinodroid in einem Bereich mit geringer Lichtstärke platziert wurde, wird die Beleuchtung der Kamera aktiviert.
- /TA055/** Mittels der Computersoftware wird das Mikrofon aktiviert. Es wird überprüft, ob akustische Signale von Android zum PC übertragen werden.
- /TA0551/** Am Computer wird die Reaktion bei Änderung der Soundqualität getestet.
- /TJ02/+ /TJ03/+ /TO06/** Es werden im Computerprogramm das Kamerabild, der Umgebungssound sowie die Informationsbox überprüft. Dabei wird die Drohne in unterschiedliche Gebiete gefahren und auf Vorhandensein der Daten sowie deren korrekte Aktualisierung geachtet.

/TJ06/+TO03/ Die Informations-/ Fehlermeldungen können beispielweise mit Trennen der Internetverbindung, Fahren in Umgebungen mit sehr schwachem Netz (mobiles Internet) oder Öffnen der Android-Kamera-Applikation im Vordergrund überprüft werden. Nach Trennen der Internetverbindung, wird kontrolliert, ob die Drohne zum Stehen kommt.

/TO04/+TA02/+TJ04/ Mittels der Pfeil-Hoch-Taste wird das Carduinodroid beschleunigt. Dies wird jeweils mit verschiedenen Geschwindigkeitseinstellungen wiederholt. Mittels der Pfeil-Runter-Taste wird das Carduinodroid abgebremst. Nach Loslassen aller Tasten hat das Carduinodroid wieder stillzustehen.

/TO05/+TA02/+TJ04/ Mittels der Pfeil-Rechts-Taste und der Pfeil-Links-Taste wird das Carduinodroid, sofern wie in /TO04/ beschreiben beschleunigt, nach rechts oder links gelenkt. Nach Loslassen Pfeil-Rechts-Taste oder der Pfeil-Links-Taste hat das Carduinodroid wieder geradeaus zu fahren.

/TO01/ Das Menü wird geöffnet.

/TO03/ Es wird während einer gesamten Testfahrt überprüft, ob permanent und mit welcher Verzögerung ein Bild der Kamera zu sehen ist.

/TO011/ Im Anschluss an alle aufgelisteten Testszenarien wird das Java- Programm über das Menü beendet.

9 Entwicklungsumgebung

9.1 Software

Plattform:

- Java 32Bit 1.7
- Android Version 2.3.3 (Gingerbread) und höher

Java 32Bit 1.7 Tools:

- MiKTeX 2.9
- Eclipse
 - Android SDK Manager
 - ADT 18.0.0
 - SDK Tools, Revision 19
 - TeXlipse

9.2 Hardware

- Internet-/netzwerkfähiges Mobiltelefon sowie Rechner

10 Glossar

Begriff	Erklärung
Aktorik	bezeichnet im Allgemeinen das Erzeugen einer Bewegung.
Android	ist ein Betriebssystem für Smartphones.
Applikation	ist ein Computerprogramm, das eine für den Anwender nützliche Funktion ausführt.
Arduino	ist eine aus Hard- und Software bestehende Mikrocontroller-Umgebung.
Arduino-Controller	ist ein Microcontroller, der benötigt wird um die Motoren des Fahrzeuges anzusteuern.
Bildqualitätsänderung	bedeutet Änderung der Bildrate und/oder der Bildauflösung.
Carduinodroid	ist die Zusammensetzung der Wörter Car, Arduino und Android zu einem Wort.
Drohne	bezeichnet ein unbemanntes Fahrzeug.
EDGE	ist ein Übertragungsart von Informationen über das Handynetz.
GPS	ist ein globales Navigationssatellitensystem zur Positionsbestimmung und Zeitmessung.
HSDPA	ist ein Datenübertragungsverfahren des Mobilfunkstandards UMTS. Das Verfahren ermöglicht DSL-ähnliche Datenübertragungsraten im Mobilfunknetz.
Java	ist eine objektorientierte Programmiersprache.

Mobiles Internet	bezeichnet die verschiedenen Übertragungsarten, wie zum Beispiel: EDGE, HSDPA et cetera.
Orgware	beschreibt Rahmenbedingungen bei IT-Projekten.
Sensorik	bezeichnet die Gesamtheit an Messeinrichtungen zur Erfassung und Kontrolle von Veränderungen in der Umwelt.
Systemstopp	ist das Anhalten der Motoren mit der Anzeige einer Warnung.
UMTS	ist ein Mobilfunkstandard der dritten Generation mit dem deutlich höhere Datenübertragungsraten erzielt werden können.
Up-/Downstream	bedeutet die Übertragung von Datenströmen von und zum Gerät.