

HAUPTSEMINAR AUTOMATISIERUNGS-, MESS- UND REGELUNGSTECHNIK: ANLEITUNG UND AUFGABENSTELLUNG MODUL HUMAN-MACHINE-INTERFACE (HMI)

VERSION 4.00

1 INHALT

2	Aufgabenstellung	1
2.1	Vorgaben	2
3	Beispielprogramm	2
3.1	Installation der nötigen Software	2
3.2	Öffnen des Beispielprojektes	2
3.3	Testen der App auf dem Tablet	2
3.4	Programstruktur	3
3.5	Modul kopieren	3
4	Wichtige Hinweise zur Bearbeitung der Aufgabe	4
5	Dokumentation	6

2 AUFGABENSTELLUNG

Im Rahmen des HMI-Moduls ist eine Mensch-Maschine-Schnittstelle zur Beobachtung und Bedienung des NXT-Roboters beim Fahr- und Einparkvorgang zu implementieren. Dazu wird ein 7 Zoll Android Tablet als mobiles Endgerät verwendet. Das bereitgestellte Android-Projekt soll unter Berücksichtigung der im folgenden Abschnitt vereinbarten Ziele und Hinweise erweitert werden. Der Schwerpunkt liegt auf dem Ausbau, der Verbesserung und Modifizierung des vorhandenen Android User-Interface.

Die fertige Nutzerschnittstelle muss folgende Funktionen implementieren:

1. Beobachtung: Anzeige des Parcours als statische Karte mit dem darauf befindlichen beweglichen Fahrzeug, dem gefahrenen Pfad und den entdeckten Parklücken. Die wichtigsten Sensorsignale sollen in aufbereiteter Form (z.B. Abstandsvisualisierung) dargestellt werden und der aktuelle Fahrzeugstatus (DRIVING, INACTIVE, EXIT) muss zu jeder Zeit erkenntlich sein.

2. Bedienung: Zuweisung von Stellflächen für vollautomatische Einparkmanöver und Einstellung des Betriebsmodus. Die Zuweisung erfolgt durch Auswahl der Parklücke auf der Karte per Touch-Geste.

Der Roboter kann in vier verschiedene Modi versetzt werden¹:

- PAUSE: Roboter ruht
- SCOUT: Roboter erkundet seine Umgebung nach Parklücken
- PARK_THIS: Roboter parkt schnellstmöglich in der gewählten Parklücke ein
- DISCONNECT: trennt Bluetooth-Verbindung zwischen Roboter und Tablet

¹ Achtung es besteht ein Unterschied zwischen *Status* und *Modus*.

Weiterhin sollte beachtet werden, dass Ausparken nicht als eigenständiger Modus realisiert wird sondern über Scout aktiviert wird. Das Signal zum Ausparken wird dabei implizit über die Aktivierung von SCOUT gegeben.

2.1 VORGABEN

- 7 Zoll Tablet PC mit Android Betriebssystem (API 15-19)
- Vorbereitetes Beispielprojekt
- Java Interface auf Android Seite

3 BEISPIELPROGRAMM

3.1 INSTALLATION DER NÖTIGEN SOFTWARE

Zur Entwicklung der Mensch-Maschine-Schnittstelle für den autonomen Roboter muss folgende Software installiert werden:

- Android Studio
- Eclipse (für Änderungen an der Roboter Software)
- NXT-Bibliotheken (mit der LeJOS NXJ Software) und Eclipse Plugin

Eine ausführliche Anleitung dafür ist in der allgemeinen Anleitung zu finden (Abschnitt 7).

3.2 ÖFFNEN DES BEISPIELPROJEKTES

Laden Sie das Beispielprojekt aus dem OPAL Downloadbereich herunter (NXT_Fernsteuerung_v*.zip) und entpacken Sie es an beliebiger Stelle. Starten Sie Android Studio und nutzen Sie auf dem Welcome Screen die Auswahl *Open an existing Android Studio project*, um das entpackte Beispielprojekt importieren zu können. Wählen Sie wie gefordert den von Ihnen entpackten Ordner aus und bestätigen Sie Ihre Auswahl. Daraufhin kann es vorkommen, dass eine Meldung erscheint. Diese sagt aus, dass der Pfad zum SDK nicht zu einem Ordner gehört. Android Studio passt diesen Pfad automatisch auf den Pfad des bei Ihnen installierten SDKs an. Klicken Sie dementsprechend OK, um die Änderung zu bestätigen. Nachfolgend wird die IDE mit dem Beispielprojekt geöffnet.

Im Projektexplorer werden jetzt die nachfolgenden zwei Module angezeigt:

- myapplication Android App
- nxt Javacode des Roboters

Hinweis: Alle Abhängigkeiten zwischen den Modulen und Bibliotheken sind bereits voreingestellt. Sollten Sie, statt das Beispiel zu erweitern, eine eigene App als neues Modul anlegen wollen, so müssen Sie die Abhängigkeiten in den Modul-Eigenschaften (Rechtsklick -> *Open Module Settings*) einstellen.

Die Pakete *parkingRobot* in Android Studio und Eclipse müssen stets synchronisiert werden, da hier die Kommunikationsschnittstelle beschrieben ist. Das bedeutet, wenn Änderungen an diesen Dateien vorgenommen werden, so müssen diese zwischen allen Gruppenmitgliedern synchronisiert werden. Hierfür empfehlen wir die Verwendung einer Versionsverwaltung, wie Git oder SVN. Informationen zur Verwendung von Versionsverwaltung finden Sie auch in der allgemeinen Anleitung in Abschnitt 7.3.

3.3 TESTEN DER APP AUF DEM TABLET

Voraussetzung für diesen Schritt ist ein fehlerfreies Android-Projekt sowie ein programmierter NXT-Roboter (Informationen hierzu finden Sie in der allgemeinen Anleitung).

Zum Testen der App auf dem Tablet müssen Sie dieses zuerst in den Debugging-Modus versetzen. Wählen Sie auf dem Tablet in den Android-Einstellungen die *Entwickler-Optionen* aus. Aktivieren Sie hier die Option *USB-*

Debugging. Nun können Sie das Gerät per USB mit ihrem Entwicklungs-PC verbinden. Hinweis: die OEM USB-Treiber für das Android Tablet müssen gemäß der Anleitung² installiert worden sein. Außerdem sollten Sie Bluetooth einschalten um den Roboter fernsteuern zu können.

Sie müssen nun noch die ADB Verbindung erlauben indem sie den Geräte-Fingerprint zulassen. Das dazugehörige Popup erscheint bei der ersten Verbindung zwischen PC und Tablet.

Vor der eigentlichen Ausführung der Android-App sollten Sie das NXT-Projekt einmal kompilieren und auf den NXT uploaden (siehe Allgemeine Anleitung Abschnitt 7.2).

Stellen Sie nun sicher, dass in der Toolbar die *Debug-Konfiguration* für *myapplication* ausgewählt wurde und klicken Sie dann rechts davon auf *Run*. Wählen Sie im folgenden Dialog Ihr Gerät und bestätigen mit *OK*. Die App wird nun auf Ihrem Gerät ausgeführt.

Es ist nicht sinnvoll, mittels des Android-Emulators zu testen. Dort steht kein Bluetooth zur Verfügung und die Ausführungsgeschwindigkeit ist meist um ein Vielfaches langsamer.

3.4 PROGRAMMSTRUKTUR

Die Beispiel-App ist in drei Pakete unterteilt.

com.example.myapplication

Dieses Paket enthält die Beispiel-App und besteht lediglich aus zwei Activities. MainActivity.java stellt den Hauptbildschirm bereit und BluetoothActivity.java ermöglicht die Auswahl eines NXT-Roboters zum Verbinden.

Sie sollten dieses Paket kopieren und erweitern um die Lösung der Aufgabe zu implementieren. Dazu wird es nötig sein neue Activities zu erstellen, das Android-Manifest abzuändern, etc. Wie Sie ein Modul kopieren lesen Sie in Abschnitt 3.5.

de.amr.plt.rcParkingRobot

Dieses Paket enthält Klassen zur Bluetooth-Kommunikation und das Java-Interface zum NXT-Roboter. **Dieses Paket sollten Sie nur wohl überlegt verändern**, da die Verbindungsstabilität und die Schnittstellen zum Roboter betroffen sein können.

lejos.pc.comm

Paket mit benötigten LeJOS Bibliotheken. **Bitte nicht verändern**.

3.5 MODUL KOPIEREN

Um ein Modul zu kopieren müssen Sie die Ansicht des Projektexplorers von *Android* auf *Projekt* umstellen. Die auszuwählende Schaltfläche ist in Abbildung 1 dargestellt. Abbildung 2 zeigt das resultierende Auswahlmenü.

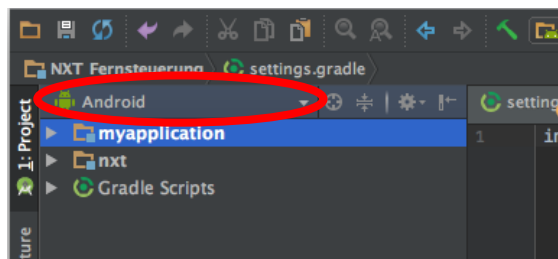


ABBILDUNG 1 PROJEKTEXPLORER (KLICKEN SIE AUF DAS MARKIERTE DROPMENÜ, UM VON ANDROID AUF PROJEKT UMZUSTELLEN)

² OEM USB-Treiber: <http://developer.android.com/tools/extras/oem-usb.html>

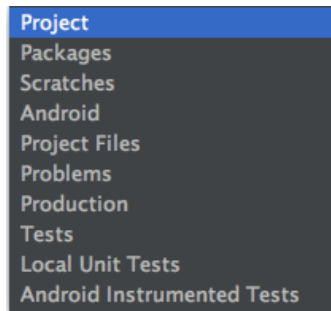


ABBILDUNG 2 PROJEKTEXPLORER (AUSWAHLMÖGLICHKEITEN)

Nachdem Sie die Ansicht erfolgreich umgestellt haben kopieren Sie anschließend den Modulordner *myapplikation* und fügen diesen parallel zu dem bestehenden alten Ordner wieder ein. Daraufhin öffnet sich ein Dialog in dem Sie dem Modul einen neuen Namen vergeben. Nachdem Sie diesen Schritt abgeschlossen haben ist es wichtig, dass Sie in der Datei *settings.gradle* Ihr neues Modul hinzufügen, damit dieses auch kompiliert wird. Ein Beispiel hierfür zeigt Abbildung 3. Die weiteren Abhängigkeiten in den *ModuleSettings* wurden automatisch mitkopiert und müssen nicht mehr angepasst werden. Wechseln Sie jetzt wieder in die Ansicht *Android* und testen Sie Ihre Kopie.

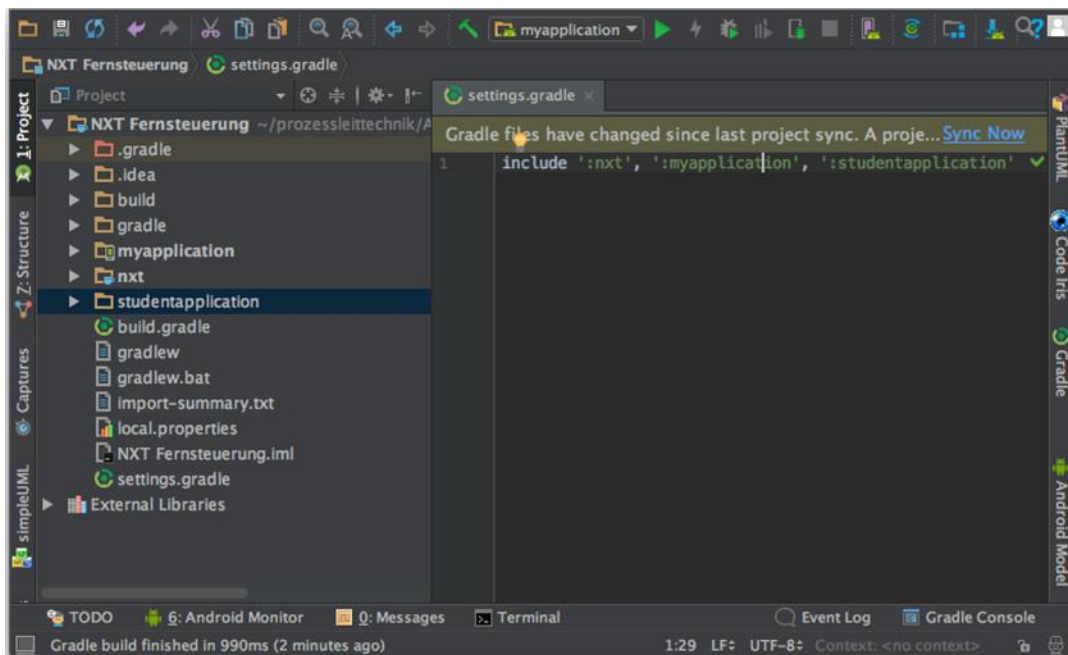


ABBILDUNG 3 NOTWENDIGE VERÄNDERUNGEN IN DER SETTINGS.GRADLE FÜR DAS KOPIERTE MODUL MIT DEM NAMEN STUDENTENAPPLIKATION

4 WICHTIGE HINWEISE ZUR BEARBEITUNG DER AUFGABE

Bitte beachten Sie, dass Ihre Teammitglieder von Ihnen abhängig sind. Ohne Ihre Arbeit lassen sich viele Funktionalitäten des Gesamtsystems nicht testen.

Bis zur ersten Verteidigung sind deshalb folgende Funktionen fertigzustellen und zu präsentieren:

- Anzeige des Parcours
- Rudimentäres Zeichnen des gefahrenen Pfades
- Rudimentäre Anzeige der gefundenen Parklücken
- Vollständiges Designkonzept als MockUp

Danach haben Sie noch bis zur zweiten Verteidigung Zeit für Optimierungen und weitere Funktionen.

Die Modalitäten der einzelnen Präsentationen werden in der Allgemeinen Anleitung im Abschnitt 6 vorgestellt. Ergänzend erfolgt an dieser Stelle eine kurze Zusammenfassung der wesentlichen Inhalte, die in den jeweiligen Präsentationen durch den HMI-Verantwortlichen vorgestellt werden sollen. Aufgrund der beschränkten Zeit, die für die jeweiligen Präsentationen zur Verfügung steht muss jeweils selbstständig entschieden werden, welche der Inhalte in der Power-Point Präsentation und welche während der Funktionsvorführung gezeigt werden.

- Zwischenpräsentation
 - o Zusammenfassung des aktuellen Stands
 - o Vorstellung eines rudimentären Designs der zu entwickelnden App als MockUp
 - o Aktuelle Probleme
 - o Nächste Ziele
- 1. Verteidigung
 - o Entwurf
 - Softwareentwurf
 - Designkonzept
 - o Umsetzung der Anforderungen
 - Anzeige des Parcours
 - Rudimentäres Zeichnen des gefahrenen Pfades
 - Rudimentäre Anzeige der gefundenen Parklücken
- 2. Verteidigung
 - o Umsetzung der Anforderungen
 - o Konzept inklusive Vorstellung des finalen HMI-Designs
 - o Verbesserungsmöglichkeiten

Damit die Android App Anforderungen wie Funktionalität, Stabilität und Nutzerfreundlichkeit erfüllen kann, sind verschiedene Aspekte während der Implementierung zu berücksichtigen. Im folgenden Abschnitt werden kurz einige relevante Aspekte aufgeführt. Die Einhaltung dieser Hinweise geht in die Notenbildung ein.

- Wenn Sie noch keine Erfahrungen mit der Android-Programmierung in Java haben, ist es äußerst empfehlenswert eines der vielen Tutorials im Internet durcharbeiten oder sich ein entsprechendes Buch aus der Bibliothek auszuleihen. In dieser Anleitung kann nicht auf die Einzelheiten der Android-Entwicklung eingegangen werden.
- Die Gebrauchstauglichkeit des User-Interface ist ein wichtiges Ziel. Hinweise dazu finden sich z.B. in der EN ISO 9241-110 (Grundsätze der Dialoggestaltung) und der EN ISO 9241-11 (Anforderungen an die Gebrauchstauglichkeit). Praktische Hinweise für Android finden sich in den Android Guidelines³ und Design Principles⁴
- Die ausführliche Kommentierung des Quelltexts ist unerlässlich. Es fördert die Klarheit und Verständlichkeit des Programms. Die Kommentierung soll nach Javadoc⁵ erfolgen, welches gegenüber gewöhnlichen Kommentaren automatisch eine HTML-Dokumentation anlegt.
- Der Entwickler sollte mit dem Activity-Lifecycle vertraut sein und wissen, wie und wann dessen Methoden benutzt werden sollten. Dazu muss man sich verdeutlichen, wie das System auf Ereignisse und Unterbrechungen reagiert und welche Effekte das auf den Activity-Lifecycle hat. Beispiele dafür

³ Android Guidelines: http://developer.android.com/guide/practices/ui_guidelines/index.html

⁴ Android Design Principles: <http://developer.android.com/design/get-started/principles.html>

⁵ Javadoc: <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

sind das Starten einer anderen Anwendung oder das Drücken der Zurück-Taste. Der Umgang mit solchen Ereignissen ist in der Implementierung zu berücksichtigen.

- Zu einer vollständigen Android-App gehört auch ein Icon. Das Icon sollte individuell gestaltet sein und sich an die Android Design Guidelines⁶ halten.
- Alle verwendeten Grafiken sind in den passenden Auflösungen für mdpi, hdpi und xhdpi in den jeweiligen Ordnern des Android Projektes vorzuhalten⁷.

5 DOKUMENTATION

In der Dokumentation sollen die Lösungen und Lösungswege zu den gestellten Aufgaben nachvollziehbar und ansprechend dargestellt werden. Der Umfang der Dokumentation soll **15 echte Seiten pro Student (ohne evtl. Anhang, ohne Titel und Inhaltsverzeichnis)** nicht überschreiten. Hochwertige kürzere Dokumentationen sind willkommen. Weitere Hinweise zu Bewertung und Abgabe der Dokumentation sind in der allgemeinen Anleitung zu finden.

Es ist empfehlenswert während der Durchführung des Hauptseminars bereits begleitend an der Dokumentation zu schreiben.

Der Inhalt muss sich an nachfolgend gegebenem Inhaltsverzeichnis orientieren. Erläuterungen stehen in eckigen Klammern.

1. Analyse der Aufgabenstellung
 - 1.1. Detaillierte Anforderungsbeschreibung
 - 1.2. Geplantes Vorgehen
 - 1.3. Schnittstellen und Zusammenarbeit zu/mit anderen Modulen/Modulverantwortlichen
2. Entwurf der Mensch-Maschine-Schnittstelle
 - 2.1. Entwurfsprinzipien [Berücksichtigte Guidelines, Gestaltgesetzte, Designprinzipien]
 - 2.2. Funktionen des HMI
 - 2.2.1. [Funktion, Zuordnung zu Anforderungen, Entwurf (Zeichnungen oder Screenshots), Erläuterung]
 - 2.2.2. ...
3. Implementierung der Mensch-Maschine-Schnittstelle
 - 3.1. Implementierung des visuellen Designs
 - 3.1.1. Vorgehen [Tatsächlicher Ablauf]
 - 3.1.2. Ergebnis [Erläuterung, Screenshots]
 - 3.1.3. Besondere HMI-Merkmale [Worauf sind sie besonders stolz?]
 - 3.2. Implementierung der Funktionalität
 - 3.2.1. Vorgehen [Tatsächlicher Ablauf]
 - 3.2.2. Ergebnis [Ablaufplan und/oder Klassendiagramm + Erklärung]
 - 3.2.3. Besonderheiten der Umsetzung [Worauf sind sie besonders stolz?]
4. ... (selbst hinzugefügte Abschnitte)
5. Anmerkungen und Verbesserungsmöglichkeiten

⁶ Android Guidelines for Launcher Icons

http://developer.android.com/guide/practices/ui_guidelines/icon_design_launcher.html

⁷ Android API-Guidelines: Supporting Multiple Screens

http://developer.android.com/guide/practices/screens_support.html#DesigningResources