Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Projekt zur Veranstaltung

Modellierung II

Sommersemester 2013

„Entwicklung eines roboterbasierten Warentransportsystems - Implementierung“

Ausgabe der Aufgabe: 12.06.2013

Abgabefrist (Abgabe per E-Mail an Sebastian.Waetzoldt@hpi.uni-potsdam.de und Übungsgruppenbetreuer): 30.06.2013

Im Rahmen der Implementierungsaufgabe soll das von Ihnen modellierte roboterbasierte Warentransportsystem für einen Simulator vom Robotino-System gebaut werden.

Für die Umsetzung bzw. Simulation des roboterbasierten Warentransportsystems wird eine Rahmenimplementierung mit den im Aufgabenblatt beschriebenen Komponenten bereitgestellt. Diese wird, zusammen mit der Bibliothek zum Ansprechen des Simulators, in einem Eclipse Projekt namens „robotinoSimulation" ausgegeben (Lehrveranstaltungsordner „Projektaufgabe Implementierung"). Dort befinden sich zwei weitere Java Klassen:
Zum einen die „Robot.java", welche die Logik eines einzelnen Roboters implementiert. Jeder Roboter wird als eigenständiger Thread (Runnable) implementiert. Wir haben eine Referenzimplementierung vorgegeben. Im Großen und Ganzen sollte nur die **drive()**-Methode angepasst werden. Eine andere Zusammenstellung der Sensorik und Aktuatoren wird nicht empfohlen.
Zum anderen die „Simulation.java". Diese Klasse beinhaltet die main()-Methode zum Starten der Simulation. Eine Referenzimplementierung zeigt das Erzeugen eines Roboters, das Zuordnen einer Simulationsadresse und das Starten der Simulation.

Im Zuge der Implementierung soll NUR die „Robot.java" angepasst werden. Diese wird als Hauptdatei abgegeben. Während des Simulationswettbewerbs wird eine etwas modifizierte main()-Methode verwendet, die die Anzahl der Roboter verändert, eine veränderte Simulationsumgebung und eventuell andere Werte für Energieverbrauch und Durchsatz des Roboters vorgibt. Werden zusätzliche Bibliotheken, externe Algorithmen verwendet oder weitere (eigene) Klassen benötigt, so müssen diese ebenfalls, in Form einer .jar-Datei (mit Sourcen) angegeben werden. Dabei ist von jeder Gruppe darauf zu achten, dass das Gesamtprojekt weiterhin funktioniert und nicht mit unserer Implementierung in der robotino.jar bwz. Rec_robotino_com_wrap.jar Probleme bereitet.

Die Installation des Simulators wird am Ende dieses Dokuments beschrieben. Eine Szene für den Simulator befindet sich im Projekt im Ordner „szene_simulator". Diese kann mit dem Programm Robotino®SIM Professional eingeladen und verwendet werden.

Wie in dem Simulator und auf Abbildung 1 zu erkennen ist, besitzt der Roboter eine statische Schiebevorrichtung. Mit dieser ist der Roboter in der Lage einzelne Warenkörbe zu transportieren.

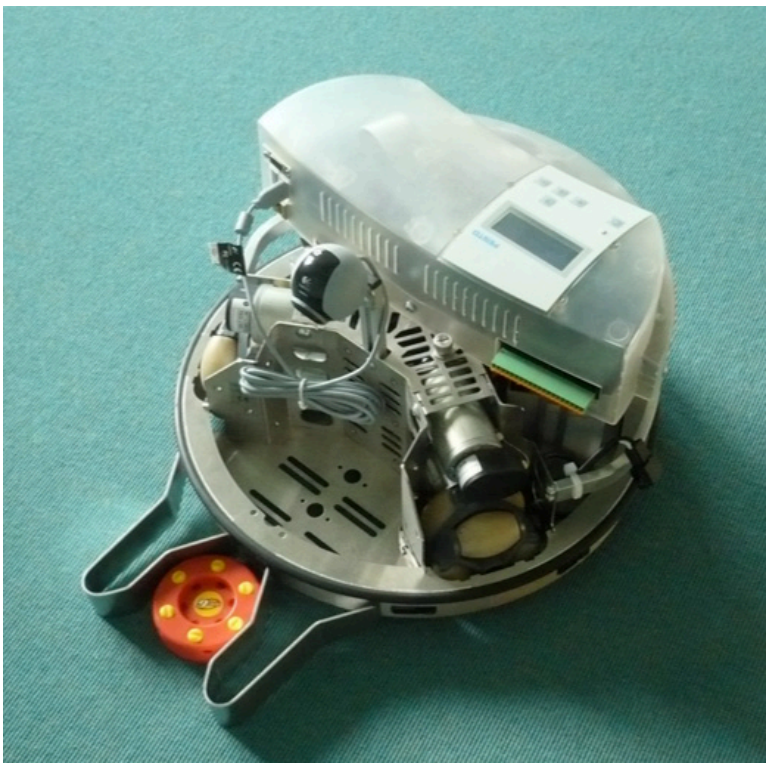


Abbildung 1: Roboter mit Puckschieber und Puck

Im Kontext der Aufgabenstellung kann davon ausgegangen werden, dass einzelne Warenkörbe in Form von Pucks (siehe Abbildung unten) vorliegen.

Diese lassen sich mit Hilfe der statischen Schiebevorrichtung am Roboter transportieren. Die einzelnen Waren im Warenkorb werden im Kontext der Aufgabenstellung nicht explizit dargestellt. Die Warenkörbe haben eine vorgegebene Kapazität, die sie nicht überschreiten sollen. In der Rahmenimplementierung sind entsprechend schon Methoden vorhanden um die Kapazität eines Warenkorbs abzufragen. Eine Bestellung, die die Kapazität eines Warenkorbs überschreitet, soll auf mehrere Warenkörbe verteilt werden.

Der Roboter selbst ist rund, hat aber immer eine Ausrichtung (vorne und hinten sind bekannt).
Als Aktuator kann der Antrieb des Roboters verstanden werden.
(Genauere Informationen bezüglich der Anordnung der Sensoren sowie der zurückgelieferten Winkel können der Abbildung 2 entnommen werden.)

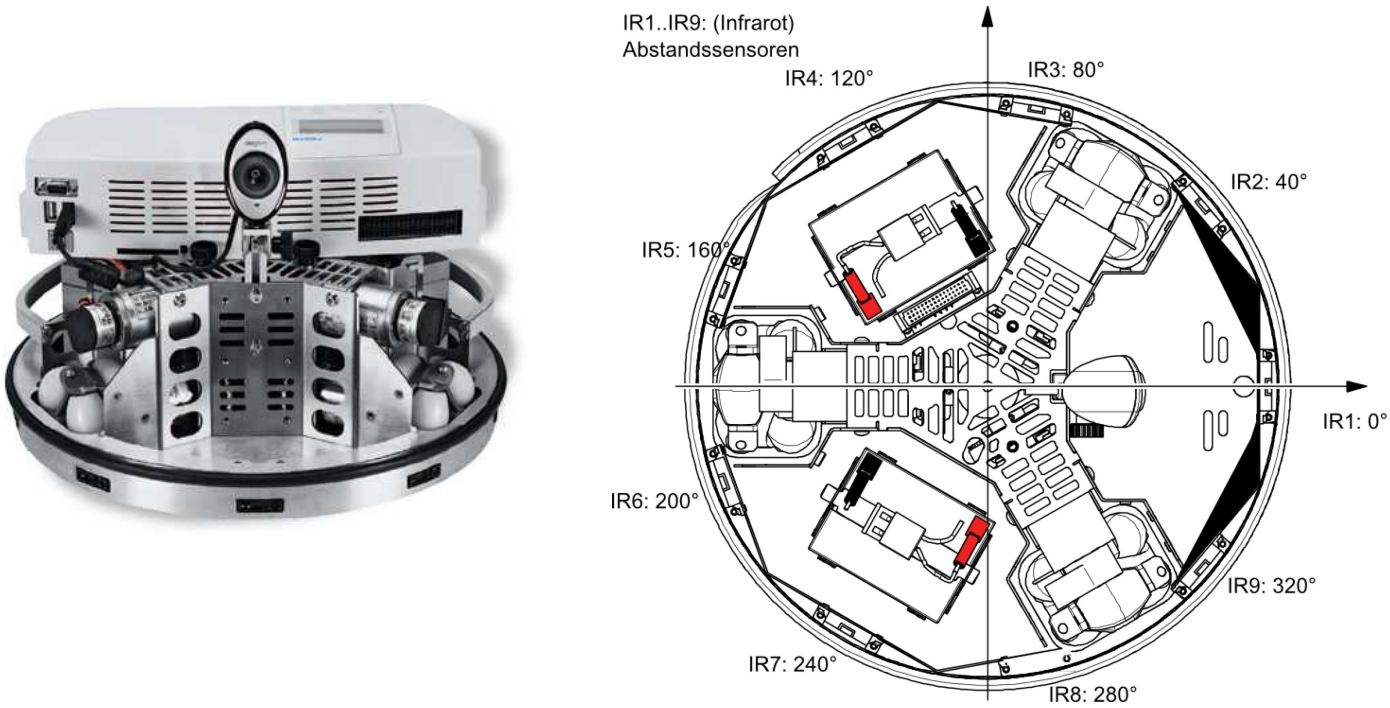
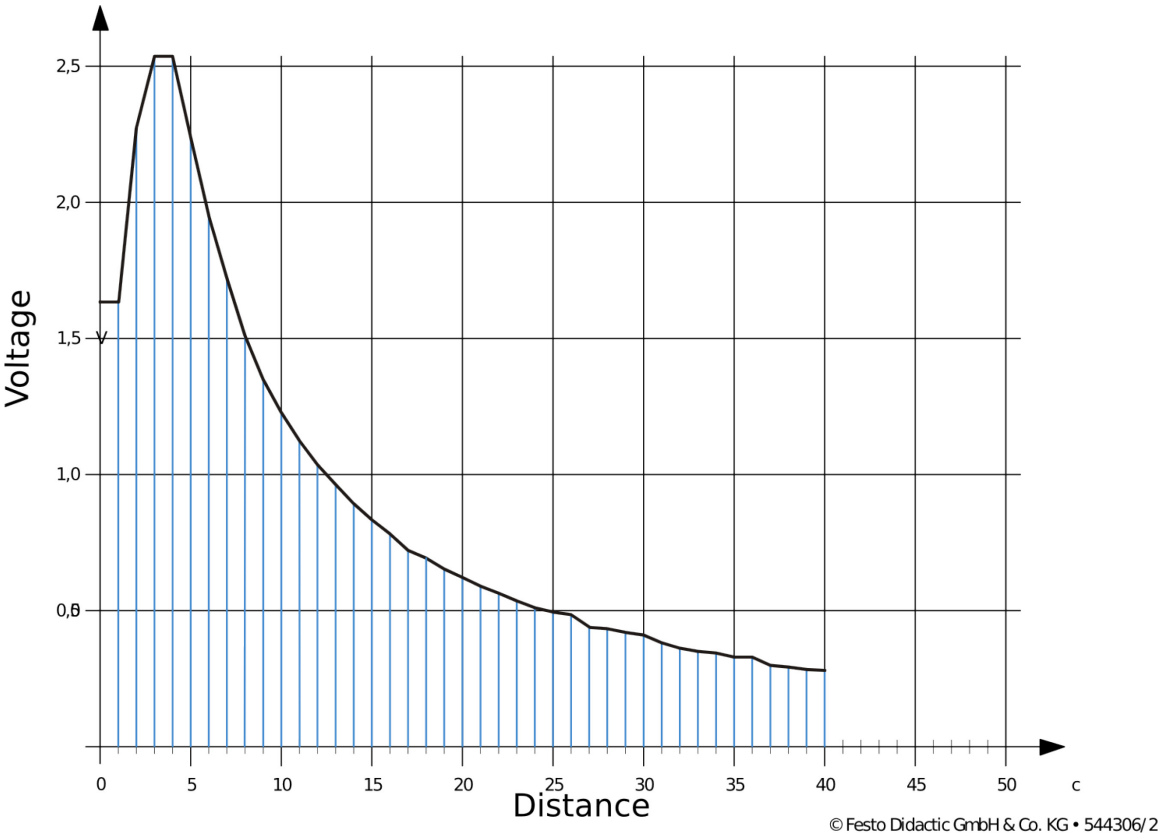


Abbildung 2: Technische Zeichnung eines Roboters

Ein einzelner Abstandssensor liefert eine Spannung, die je nach Entfernung zum Hindernis variiert. Die kleineren Schwankungen, aufgrund der Ungenauigkeiten des Sensors, können dabei vernachlässigt werden. Hierbei ist zu beachten, dass die Spannungswerte bei sehr nahen Hindernissen nicht eindeutig sind (siehe Funktion unten).



Wenn die Funktionsweise einzelner Sensoren oder Aktuatoren unklar ist, so können Details hier (http://doc.openrobotino.org/documentation/OpenRobotinoAPI/1/doc/rec_robotino_com/index.html) nachgelesen werden. In der Referenzimplementierung haben wir allerdings die Hauptbestandteile eines Robotino Roboters noch einmal gekapselt.

Die bereits existierende Bewegungslogik (MoveLogic) implementiert ein grundlegendes Bewegungsverhalten des Roboters und kann bei Bedarf durch eine eigene Implementierung ersetzt werden. Ebenso muss das Ausweichverhalten des Roboters bei Hindernissen umgesetzt werden. Achtung: die zweistelligen Koordinaten im Simulator sind in Meter angegeben. Die MoveLogic rechnet allerdings in Millimeter (Umrechungsfaktor 1000 siehe Referenzimplementierung).

In Abbildung 3 wird zusammenfassend illustriert wie ein Transportsystem zur Laufzeit im Simulator beispielhaft aussieht.

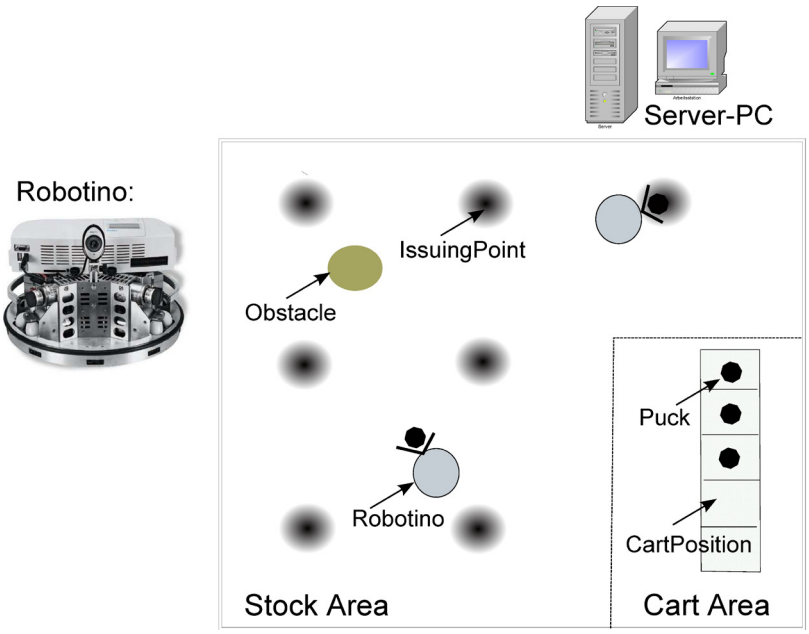


Abbildung 3: Beispiel eines Transportsystems zur Laufzeit

Installation des Simulators

ACHTUNG: Der Simulator läuft nicht auf einem Windows 8 Betriebssystem.

1. robotino.zip enthält ein Java Projekt, welches im Eclipse Workspace importiert warden kann.

2. RobotinoSimPro-1.2.3.exe installieren

   - Programm starten und bei fehlender Lizenz folgende Schritte durchführen

   - Öffne CodeMeter Control Center aus Symbolleiste

   - "WebAdmin" im "License"-Tab auswählen

   - Auf der Webseite "configuration" und neuen Eintrag in "Server search list" hinzufügen - Server Name: fb14srv2, Dann OK, dann Apply - Innerhalb des HPI Netztes kann das Programm nun benutzt werden - Nach dem Starten des Programms eine Landschaft auswählen und den Simulationsmodus starten (zweiter Button von rechts in der Aktionsleiste)

3. OpenRobotinoAPI-vc100-1.8.31.exe installieren

   - Die *.dll Dateien in den folgenden Installationspfad der API schreiben .../1/bin/win32 (Alternativ die entsprechende Path Variable anpassen)

4. Eclipse installieren (Voraussetzung ist ein System mit 32bit JVM)

   - import Java projekt und die robotino.zip auswählen

   - Die Klassen Simulation.java und Robot.java geben einen Eindruck über die Verwendung der RoboterAPI

   - Zum Testen der Konfiguration Simulation.java als Java Application ausführen (rechte Maustaste RunAs)

   - Sollten die *.dll Dateien der API von Java nicht gefunden werden ist ein Neustart notwendig

   - bitte darauf achten, dass sowohl die robotino.jar als auch die robotino_api*.jar im build Path im Eclipse Projekt hinzugefügt wurden.