Roboterfahrzeugsteuerung mittels Tinkerforge und Raspberry

HAUSARBEIT

des Studienganges Informationstechnik

an der Dualen Hochschule Baden-Württemberg Mannheim

von

Sebastian Barz

Simon Buttke

Ferdinand Frank

Sebastian Hantelmann

Yusuf Peker

10.12.2013

Bearbeitungszeitraum	12 Wochen
Kurs	TINF11ITNS
Vorlesung	Robotik
Dozent der Dualen Hochschule	Prof. Jochem Poller

Inhaltsverzeichnis

1 Einleitung

asdf

2 Tinkerforge

2.1 Erste Schritte

Zur Steuerung des Trucks wird ein Tinkerforge Stapel, bestehend aus einem Master-Brick, zur Kommunikation innerhalb des Stapels, einer WLAN-Masterextension, als Verbindungsmodul zwischen Tinkerforge und gewünschten Steuergerät, sowie einem Servo-Brick, zur Ansteuerung der Servo-Motoren, verwendet.

Master-Brick

Servo-Brick

WLAN-Masterextension

Die adäquate Steuerung und Benutzung des Trucks wird grundsätzlich über das Wifi-Bricklet des Tinkerforge ermöglicht.
Initial muss hierfür aber eine USB-Verbindung hergestellt werden, worüber Einstellungen zur Einrichtung, aber auch erste Tests durchgeführt werden können. Zur Benutzung des Tinkerforge über einen PC muss zuvor ein BrickDeamon (Brickd) und ein Brickviewer (BrickV) installiert und gestartet werden. Diese können auf der Internetseite der Firma Tinkerforge unter http://www.tinkerforge.com/de/doc/Downloads.html heruntergeladen werden.
Bevor jedoch der Zugriff über den Brickviewer möglich ist, muss ein Update der Firmware der Bricklets durchgeführt werden. Dies ist durch eine Schaltfläche des Brickviewers möglich.
Über den Brickviewer wird die Einrichtung des WLAN-Bricklets durchgeführt. Das Team entschied sich für die Einrichtung eines Access Points mit einer statischen IP.

Die Wahl fiel auf diese WLAN-Einstellung da sie die konsistenteste und stabilste bei einigen Versuchen zu sein scheint.
Durch den Brickviewer können somit Servo-Motoren Tests durchgeführt und ein Verständnis für die einzelnen Parameter und Einstellmöglichkeiten erlangt.
Das Befestigen des Tinkerforge Stapels mithilfe einer Platte und einigen Schrauben an der Querstrebe des Trucks sorgt für Stabilität und Sicherheit der Hardware, welche zuvor nur durch die verbunden Kabel auf dem Truck gehalten wurden.

Ggf.
Um nun eine Schnurlose Verbindung zum Tinkerforge WLAN-Briklet herzustellen sind die Internetoptionen anzupassen und eine zur statischen IP des Access Points des Trucks passende IP zu konfigurieren. Dann sollte eine Verbindung zum Bricklet vorhanden sein.

2.2 Ansteuerung über die Java API

Nachdem der Tinkerforge mit den Servos und dem Computer verbunden worden war, sowie einige Tests mit dem Diagnose-Tool durchgeführt worden waren, soll der Mikrocontroller nun über die Java API angesteuert werden. Der Truck soll mit den Pfeiltasten der Tastatur gefahren und gelenkt werden.
Zuerst muss analysiert werden, welche Signale an den Servo für die Lenkung und an das Steuerungsmodul für den Antrieb gesendet werden müssen. Wichtigstes Attribut für das Servo-Objekt in der Java API ist dabei die „Position“. Dieses Attribut hat einen Minimal- und einen Maximalwert. Bezogen auf unseren Truck bedeuten diese Werte beim Servo „links“ und „rechts“ und beim Steuerungsmodul „vorwärts“ und „rückwärts“. Da die Servo-Schnittstellen des Tinkerforge impulsgesteuert sind, muss beim Antrieb eine geeignete Pulsweite eingestellt werden.
Ein Verbindungsaufbau sieht wie folgt aus:

```
IPConnection ipcon = new IPConnection();
BrickServo servo = new BrickServo(UserID, ipcon);
ipcon.connect(host, port);
```

In einem ersten, einfachen Java-Programm werden Event-Listener für die vier Pfeiltasten programmiert. Das Drücken der Tasten setzt die Positions-Werte der

entsprechenden Objekte auf den Minimal- oder Maximalwert, das Loslassen wieder zurück auf Null (Ruheposition, Lenkungmittig, kein Antrieb).
servo.setPosition(servoNumber, position);
Möchte man zum Beispiel nach links lenken, gibt man für „servoNumber“ den Anschluss des Lenkungs-Servo am Tinkerforge an und für „position“ den frei definierbaren Maximalwert (linke Stellung).
Zu diesem Zeitpunkt kann der Truck bereits ferngesteuert werden.

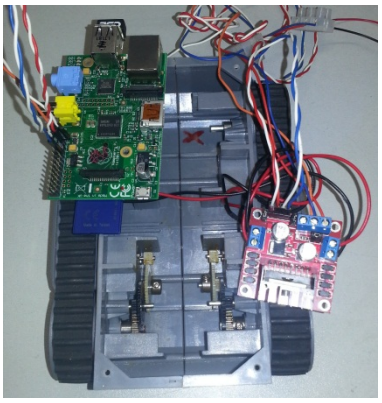
2.3 Ausblick

Der nächsten Schritte sind, den Tinkerforge mit Infrarot-Abstandsmessern auszurüsten, herauszufinden, wo sie am besten am angebracht werden, die Messwerte über die API abzufragen und dann zu implementieren, wie der Truck seine Position bestimmen und Hindernissen ausweichen kann.

3 Raspberry

3.1 Hardware

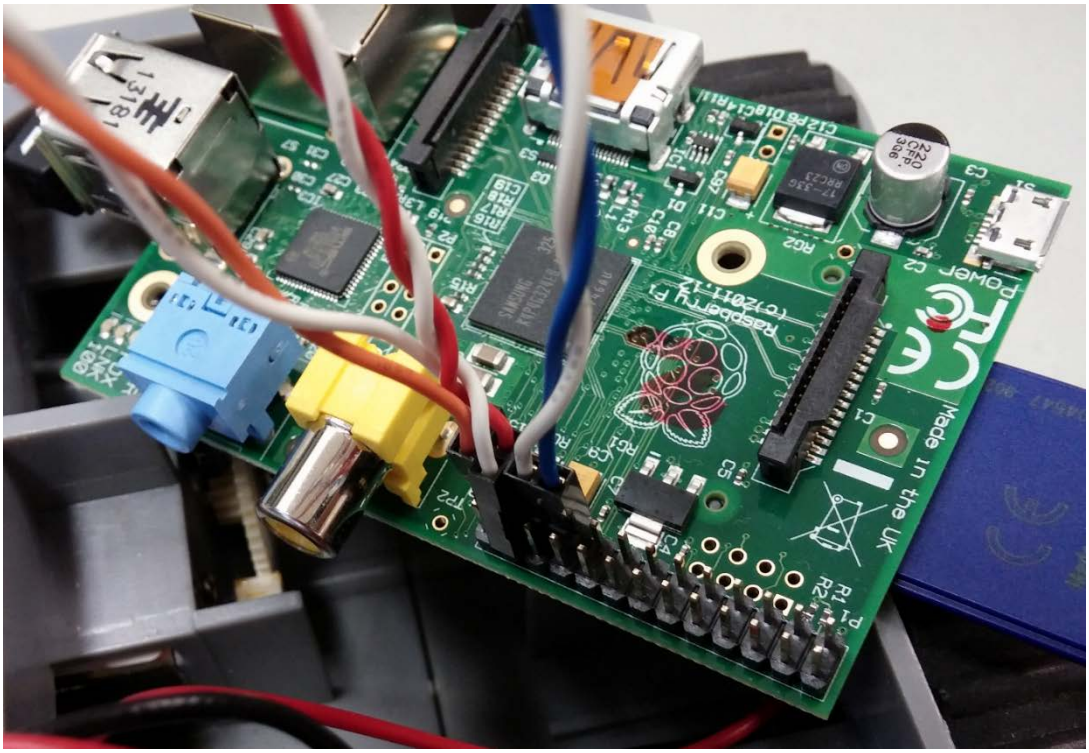
Um einen Roboter wie diesen zu bauen, werden zahlreiche Einzelkomponenten benötigt. Dazu hier zunächst ein Gesamtüberblick, wie der Roboter am Ende aussehen soll. Als Grundgestell dient der Roboter RP5, und darauf als Steuereinheiten sowohl einen Raspberry Pi als auch einen Schrittmotor.



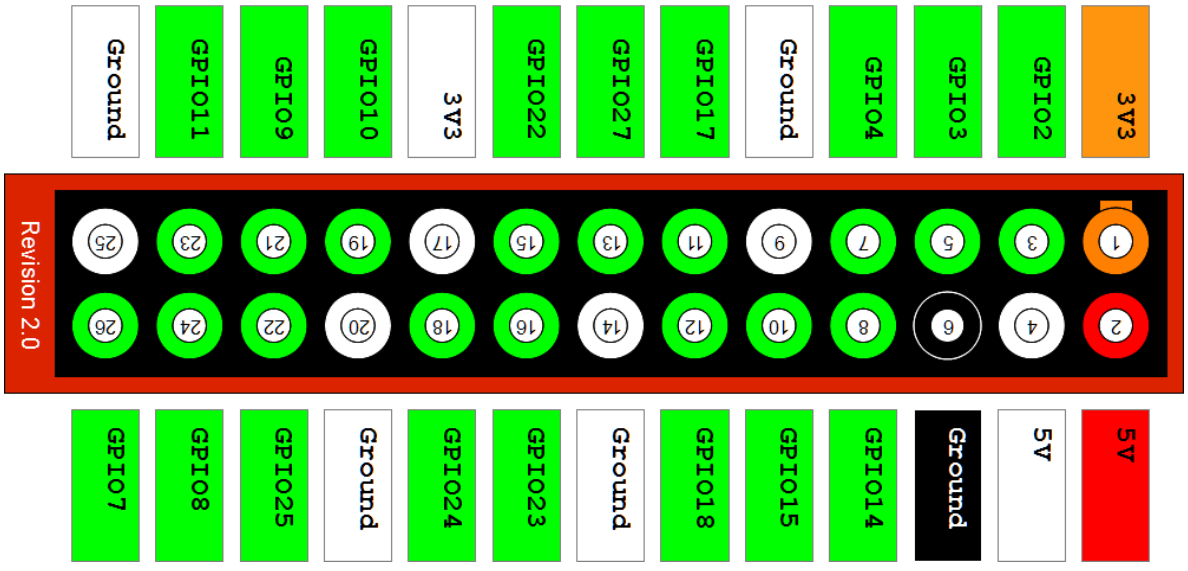
Im Folgenden wird dieser Gesamtüberblick aufgeteilt auf die einzelnen Bauteile.
Der Roboter RP5 ist ein einfaches Fahrgestell mit Kettenantrieb. Dadurch besitzt es auf jeder Seite jeweils einen Motor, welcher vorwärts oder rückwärts betrieben werden kann. Die Steuerung ist dadurch simpel, da für Vorwärts- / Rückwärtsbewegungen beide Seiten nur in die gleiche, für Drehungen in unterschiedliche Richtung steuern brauchen.
Wie jedes elektronische Bauteil brauchen auch die hier verwendeten eine Energieversorgung. Der Raspberry Pi lässt größenbedingt nur einen Micro-USB-Anschluss zu, daher wird hierfür eine USB-Powerbank verwendet. Zur Versorgung des Schrittmotors werden 7.2V benötigt, welche sich mittels normaler AA-Akkupacks und einem Batteriehalter realisieren lassen.

3.2 Zusammenbau

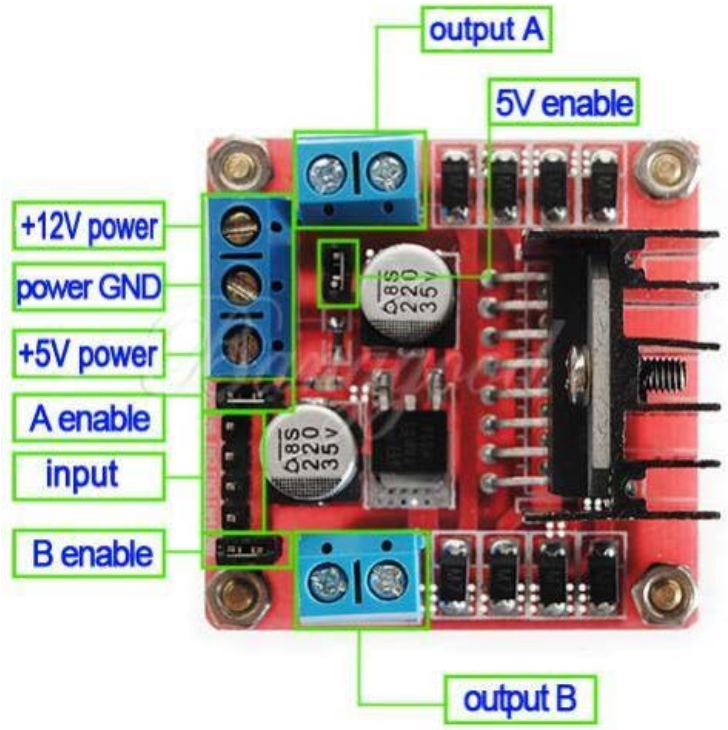
Die Montage des Roboters bestand im westentlichen aus 2 Schritten. Der erste ist es, wie in Bild 1 zu sehen, den Raspberry Pi mit dem Stepper zu verbinden um ihn anzusteuern und ihn mit Strom zu versorgen und der letzte die Outputs vom Stepper mit den Motoren des Fahrzeugs zu verbinden, wie im folgenden Bild zu sehen.



Im folgenden Bild ist die Belegung der GPIO Pin zu finden. Für den Stepper werden die Pins 25 als Ground, Pin 19, 21, 23, 24 als Output an den Stepper.



Die Outputs kommen an +5 power, Input, A und B Enable, die Ground Verbindung werden miteinander verbunden. Output A und B werden mit den Motoren des Fahrzeugs verbunden.



Nach Abschluss dieser Tätigkeiten kann man zum Programmieren übergehen.

3.3 Software

3.3.1 Raspbian

Raspbian ist eines der vielen verfügbaren Betriebssystemen für den RPi und liegt in der neuesten Version „wheezy“ vom September 2013 vor.
Raspbian basiert auf der bekannten Linux Distribution Debian und bietet neben der Anpassung an die ARM-Struktur einige Optimierung speziell für den RPi. Zur diesen Optimierungen gehört auch die Integration der GPIO Python Library, mit der es möglich ist die RPi GPIO direkt mittels Python anzusprechen.

3.3.2 Python

Python ist eine universelle, üblicherweise interpretierte höhere Programmiersprache, die sich besonders durch ihre einfache Programmstruktur und damit hohe Programmlesbarkeit auszeichnet. Nicht wie bei anderen Sprachen mit

Klammern, wird in Python nur mit Einrückungen gearbeitet. Genauso kann auf ein Abschlusszeichen am Zeilenende verzichtet werden.
Auf dem RPi verwendeten Betriebssystem „Raspbian“ ist Python in Version 2.5.4 bereits integriert und lässt sich ohne weiteren Interpreter direkt mit vi oder nano schreiben und mit der Kommandozeile ausführen.

3.3.3 Programmcode

Dank der Kettenfahrzeugsteuerung und der Möglichkeit die GPIO direkt anzusprechen, lässt sich das Fahrzeug mit ein paar Grundbefehlen simpel steuern. Hier als Beispiel ein Auszug aus dem Programmcode um das Fahrzeug geradeaus fahren zu lassen:

```
#Get the GPIO module
import RPi.GPIO as GPIO
#Set the GPIO mode
GPIO.setmode(GPIO.BOARD)
#Set the pins to be outputs
GPIO.setup(19,GPIO.OUT)
GPIO.setup(21,GPIO.OUT)
GPIO.setup(23,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)

def ControlAPairOfPins(FirstPin,FirstState,SecondPin,SecondState):
    if FirstState == "1":
        GPIO.output(int(FirstPin),True)
    else:
        GPIO.output(int(FirstPin),False)
    if SecondState == "1":
        GPIO.output(int(SecondPin),True)
    else:
        GPIO.output(int(SecondPin),False)
    return

while True:
```

```
MyChar = raw_input("Press a character:")
print "You pressed: " + MyChar
if MyChar == "w":
    ControlAPairOfPins("19","1","21","0")
    ControlAPairOfPins("23","0","24","1")
    print "Forward"
```