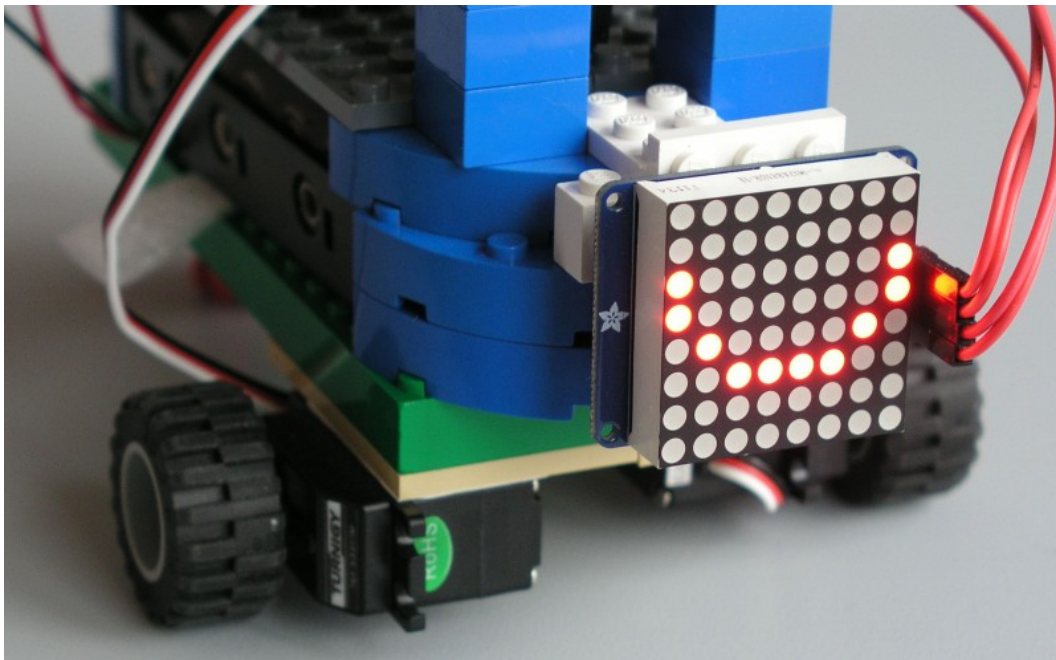
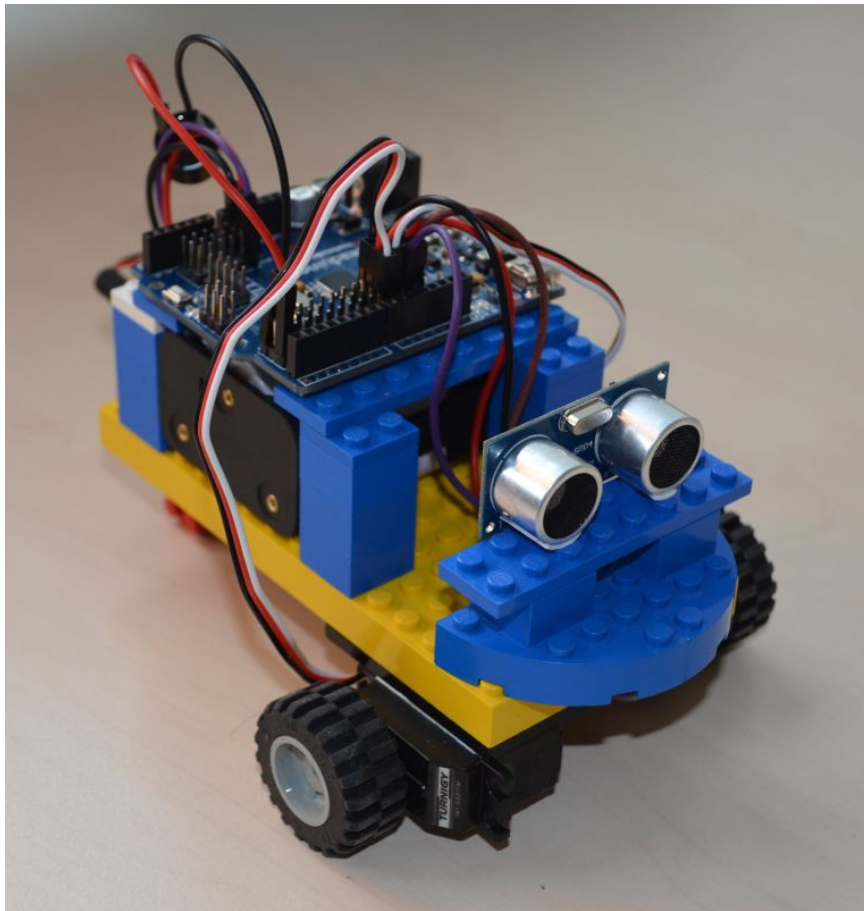


Legoino: Low Cost Lego-Roboter auf Arduinobasis



Inhaltsverzeichnis

Legoino: Low Cost Lego-Roboter auf Arduinobasis.....	1
1 Zielgruppe.....	3
2 Motivation.....	3
3 Konzept.....	4
3.1 Leitlinien des Legoino-Konzepts:.....	4
3.2 Systemdesign des Legoinos.....	4
3.2.1 Beispiel: Legoino als autonomer Roboter mit Ultraschallsensorik.....	6
3.2.2 Beispiel: Legoino als Smartphone-gesteuerter Roboter.....	6
4 Realisierung.....	7
4.1 Auswahl der 180°-Servos.....	7
4.2 Umbau des S3317M Servos auf volldrehende Funktion.....	7
4.3 Adaption von Lego-Rädern an die Servos.....	10
4.4 Ansteuerung der Motoren mit direkter Stromversorgung über den Arduino.....	10
4.5 ArduSmartPilot App zur Steuerung des Legoino.....	11
5 Gesamtkosten des Legoino.....	11
5.1 Kosten Beispiel Legoino ohne Smartphonesteuerung:.....	12
5.2 Kosten Beispiel Legoino mit Smartphonesteuerung:.....	12
6 Mögliche Probleme und deren Lösungsmöglichkeiten.....	12
6.1 Volldrehende Servos bewegen sich bei 90° Signal.....	12
6.2 Sketch kann nicht auf den Arduino übertragen werden.....	13
6.3 Arduino geht beim Betrieb des Roboters unkontrolliert in den Reset-Zustand.....	13
6.4 Android-App funktioniert nicht.....	13
6.5 Bluetoothmodul kann nicht parametrisiert werden.....	13
6.6 Ich möchte meinen vorhandenen Arduino UNO statt des Itearduino verwenden.....	14
6.6.1 Spannungsversorgung der Servos.....	14
6.6.2 Anschließen der Sensoren und Servos.....	14
7 Fazit.....	14
8 Anhang.....	16
8.1 Hardwarekomponenten.....	16
8.2 Pinbelegung Beispiel Legoino ohne Smartphonesteuerung.....	17
8.3 Pinbelegung Beispiel Legoino mit Smartphonesteuerung.....	17

1 Zielgruppe

Die nachfolgende Präsentation und Bauanleitung richtet sich in erster Linie an Erwachsene mit grundlegenden Elektronik- und Programmierkenntnissen, die mit Kindern oder Jugendlichen ohne großes Finanzbudget einen kreativen und individuellen Roboter bauen möchten.

Die nötigen Grundkenntnisse können über das „Basteln“ mit einem Arduino gewonnen werden.

Hierzu gibt es sowohl eine große Anzahl von Publikationen im Internet als auch Fachbücher [1], [2], [3] Bevor Sie mit einem Legoino beginnen, sollten Sie folgende Kenntnisse besitzen:

1) Anschließen von Sensoren und Modellbauservos an einen Arduino.

2) Programmierung eines einfachen Arduino-Sketches unter Verwendung von Bibliotheken wie z.B. `Servo` oder `NewPing`.

Eine besondere Zielgruppe sind Lehrer und Tutoren, die im Bildungsbereich Roboterprojekte mit Schülern durchführen.

2 Motivation

In Spielwarengeschäften und Elektronikmärkten sind eine Vielzahl von Robotern erhältlich, die mit Sensoren ausgestattet sind und über Fernbedienungen oder Smartphones gesteuert werden. Diese Spielzeuge sind entweder teuer und/oder wenig robust und verlieren schnell an Reiz, da weder deren Hardware noch Software erweiterbar ist.

Abgesehen davon ist der Lerneffekt beim Umgang mit diesen Systemen äußerst gering.

Das ist anders bei elektronischen Robotersystemen mit frei programmierbarem Mikrocontroller (μC), die für den Hobbybereich angeboten werden:

Der preiswerteste davon ist der **Asuro des Herstellers Arexx Engineering**¹. Er kostet als Bausatz ca. 50 €.

Hierbei handelt es sich um eine ungeschützte Elektronikleiterkarte als Basis, die mit zwei über Gleichstrommotoren angetriebenen Rädern bewegt wird. Vom mechanischen Aufbau her ist der Asuro-Roboter recht fragil und absolut unflexibel. Er kann aber sehr flexibel elektronisch z.B. durch Sensoren erweitert werden. Allerdings sind hierfür gewisse Elektronik-Grundkenntnisse und Lötarbeit Voraussetzung. Einen recht ähnlichen Roboter gibt es vom selben Hersteller auch auf Arduino-basis zum gleichen Preis.

Im Schul- und Privatbereich wird oft das **Lego Mindstorms System**² eingesetzt.

Dieser Roboterbausatz befindet sich mit ca. 350 € am oberen Ende der Preisskala. Er basiert auf Lego Technic Bausteinen und ist dadurch von der Mechanik her absolut flexibel und robust.

Im Gegensatz zum Asuro-Roboter kann er nicht mit beliebiger Hardware wie Sensoren oder Funktionsschnittstellen erweitert werden. Hier muss man auf Zubehörteile von Lego oder von Drittanbietern³ zurückgreifen, die allesamt recht hochpreisig sind.

Die aktuelle Version EV3 von Lego Mindstorms basiert auf einer Steuerung mit Linux-Betriebssystem, welches zwar sehr leistungsfähig aber gleichzeitig auch sehr komplex ist.

Lego Mindstorms ist somit eine fast ideale Plattform zum Bau von Robotern im Hobby- und Schulbereich.

Dessen breiter Einsatz scheitert meist aber an den hohen Systemkosten, insbesondere wenn für Kurse oder Schulklassen gleich 20 oder mehr Systeme angeschafft werden müssen.

In den vergangenen Jahren hat die Open-Source-Software- und -Hardware-**Plattform Arduino**⁴ sich zu einem Standard von μC -Projekten im Hobby- und Ausbildungsbereich entwickelt.

Entscheidend dafür sind die niedrigen Hardwarekosten, die kostenlosen Programmierumgebungen (IDE) sowie eine Vielzahl preiswerter Peripheriehardware wie z.B. Sensoren, Displays oder Kom-

1 Siehe www.arexx.com.

2 Siehe www.mindstorms.lego.com.

3 Siehe z.B. www.mindsensors.com.

4 Siehe www.arduino.cc.

munikationsmodule⁵.

Insofern ist es naheliegend, einen Low Cost Roboter auf Basis des Arduino zu entwickeln.

Was das „Arduino-Ökosystem“ nicht liefern kann, sind die Antriebe und einen flexiblen mechanischen Aufbau des Roboters.

Als robuste und preiswerte Antriebe können jedoch Modellbauservos verwendet werden. Und für den mechanischen Aufbau des Roboters bieten sich normale Legosteine (nicht zwingend Lego Technic) an, da diese bei vielen Erwachsenen und Jugendlichen ohnehin aus Kindertagen noch verfügbar sind.

Rechnet man die schon vorhandenen Legosteine nicht mit ein, so kann mit einer Arduino-µC-Platine, mit zwei modifizierten Servos und mit einigen Sensoren **für ca. 50 € ein flexibler „Legoino-Roboter“** aufgebaut werden. Dieser Roboter ist für den Hobby- und Schulbereich vergleichbar mit einem sechs mal so teuren Lego Mindstorms System.

Der Legoino besitzt sogar gegenüber Lego Mindstorms den Vorteil einer **Open Source IDE**, einer ungleich höheren Anzahl möglicher Hardwareerweiterungen und einer breiten Internetgemeinschaft, die bei der Softwareerstellung und dem Anschluss der Hardware Hilfe leistet.

3 Konzept

3.1 Leitlinien des Legoino-Konzepts:

1. Open Source Software für die Roboterprogrammierung.
2. Open Source Hardware für die Robotersteuerung und Elektronikperipherie.
3. Verwendung standardisierter Low Cost Bauteile mit Zielpreis Roboter <50 €.
4. Vermeidung von Lötarbeit durch vorkonfektionierte Steckverbinder.
5. Verwendung vorhandener einfacher Legosteine für die Robotermechanik.
6. Verbindung der Elektronikkomponenten mit Legosteinen für einen flexiblen Roboteraufbau.
7. Nur lösbare Verbindungen damit Roboterkomponenten uneingeschränkt wiederverwendet werden können.
8. Integration von Android-Endgeräten für die Kommunikation mit dem Legoino.

3.2 Systemdesign des Legoinos

Die mechanische Basis bilden normale Legosteine mit Noppen, also Legosteine die in fast jedem Haushalt mit (früheren) Kindern verfügbar sind. Falls nicht, dann können solche Legos auch sehr preiswert auf Flohmärkten oder im Internet im gebrauchten Zustand erworben werden.

Als Antriebe werden Modellbauservos verwendet. Normalerweise dreht sich deren Achse nur um einen Winkelbereich von $\pm 90^\circ$, weshalb diese auch als „180°-Servos“ bezeichnet werden. Im Modellbauhandel sind aber auch sogenannte volldrehende Servos verfügbar: Bei dieser Spezialversion bewirkt ein Steuersignal nicht eine definierte Winkelstellung wie bei normalen 180°-Servos, sondern eine definierte kontinuierliche Drehzahl.

Alle analogen 180°-Servos können grundsätzlich in volldrehende Servos umgebaut werden. Für den Legoino wird dieser Weg gewählt, da hier eine weitaus größere Auswahl an Servomodellen zur Auswahl steht, die gleichzeitig wesentlich kostengünstiger sind.

Die Energieversorgung wird über handelsübliche und ungefährliche AA-Akkus gewährleistet, die in externen Akkuladegeräten aufgeladen werden. Anders als bei LiPo-Modellbauakkus ist also kein thermisches Durchgehen der Akkus zu befürchten und leere Akkus können „im Notfall“ durch normale Batterien ersetzt werden.

⁵ Siehe z.B. sparkfun.com, exp-tech.de oder watterott.com.

Streng genommen sollte der Legoino zwei getrennte Stromversorgungen haben:

Eine für den μC und eine für die Antriebe. Der μC hat eine Betriebsspannung von 5 V. Er besitzt aber eingangsseitig einen DC-DC-Wandler der eine Gleichspannung zwischen 7 und 12 V benötigt.

Bei Verwendung von AA-Akkus sind also mindestens 6 Akkus nötig, die dann eine Spannung von knapp 7 V bereitstellen.

Servos sind für eine Versorgungsspannung von ca. 4,8 bis 6 V ausgelegt. Daher können sie nicht direkt über den Akkupack versorgt werden. Den DC-DC Wandler des Iteaduinios für die Servos mit zu benutzen, ist kritisch, da dieser nicht auf hohe Servoströme ausgelegt ist. Trotzdem wird beim Legoino diese Lösung gewählt, um die Kosten und Komplexität des Systems gering zu halten.

Die Steuerung des Legoino übernimmt eine Arduino- μC -Platine. Aus Kostengründen wurde dafür das Hardwaredesign „Arduino UNO / Duemilleanova“ ausgewählt, das auf einem 8-Bit Atmel μC basiert.

Aus der Vielzahl der auf diesem Design basierenden Arduino-Klone wurde das Modell „Iteaduno“ ausgewählt⁶: Als Besonderheit besitzt dieser Arduino einen DC-DC-Wandler, der die für einen Batteriebetrieb nötige Energieeffizienz gewährleistet und zusätzlich die Antriebe versorgt. Weiter besitzt dieser Arduino vorkonfektionierte Steckplätze für Servos und andere Hardwareerweiterungen, die ein Löten von Anschlussleitungen weitgehend überflüssig macht.

Es ist aber auch möglich, einen schon vorhandenen Arduino Uno (egal ob im Original oder ein sogenannter „Klon“) zu verwenden. Siehe dazu die Anmerkungen in Abschnitt 6.6.

Der Arduino selbst wie auch sämtliche Hardwareerweiterungen werden mit Legosteinen verbunden. Dadurch sind diese flexibel am Legoino platzierbar. Die Legosteine sollen dabei nicht irreversibel verändert werden, weshalb die Verbindung mittels doppelseitiger lösbarer Klebebänder realisiert wird.

Die elektrischen Verbindungen sollen ausschließlich über Steckverbinder erfolgen, die wenn möglich schon fertig konfektioniert an den Hardwareerweiterungen vorliegen.

Als Hardwareerweiterungen kommen Sensoren, Aktuatoren und Kommunikationsbausteine in Frage. Diese Erweiterungen gibt es teils fertig konfektioniert für die am Iteaduno vorhandenen Steckplätze. Neben Servos kommen z.B. Leuchten (LEDs, Laser), Summer, Lautsprecher oder Gleichstrommotoren in Frage. Eine Kommunikation kann z.B. über Bluetooth (BT), WLAN oder Infrarot erfolgen. Diese Elektronikbauteile werden jeweils auf Legosteine geklebt.

Als Benutzerinterface (HMI) werden neben Grafikdisplays auch Android-Endgeräte verwendet, die über BT mit dem Legoino kommunizieren.

Das Programmieren des μC geschieht über die Open Source Arduino IDE, welche kostenlos für verschiedene Plattformen verfügbar ist.

Wird eine Android-Endgerät als HMI verwendet, dann wird dessen App über die Open Source IDE Processing⁷ (im Android-Modus) erstellt, die ebenfalls kostenlos für verschiedene Plattformen erhältlich ist.

Beide IDEs können auf einem preiswerten PC (z.B. Windows 7 mit ca. 2 GHz Taktrate und 2 GB Arbeitsspeicher) ohne Einschränkungen verwendet werden.

Nachfolgend sind von den schier unendlichen Möglichkeiten zwei konkrete Legoinomodelle näher beschrieben:

Das erste Modell nutzt einen Ultraschallsensor zur Kollisionsvermeidung und kann damit autonom umherfahren.

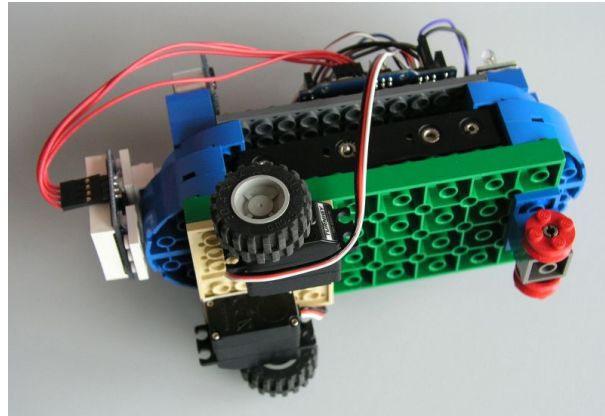
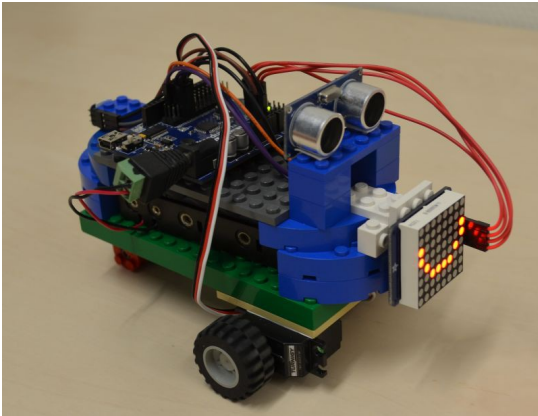
Das zweite Modell wird über ein Android-Endgerät via BT ferngesteuert.

Eine Kalkulation der Hardwarekosten für beide Modelle finden Sie in Abschnitt 5, eine exemplarische Stückliste in Abschnitt 8.1.

6 Siehe www.imall.iteadstudio.com/im130312001.html.

7 Siehe www.processing.org.

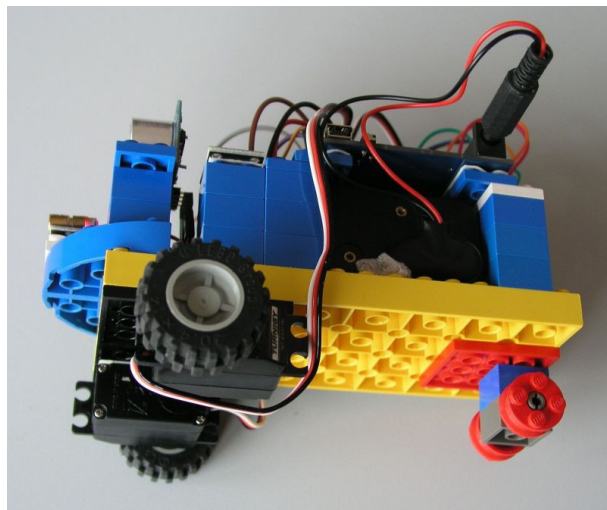
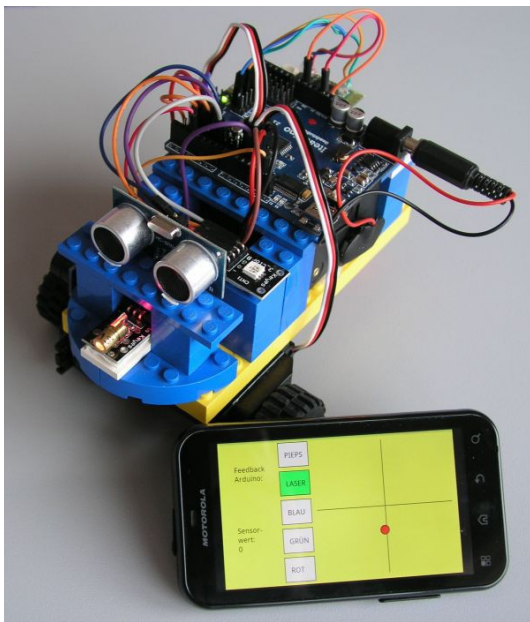
3.2.1 Beispiel: Legoino als autonomer Roboter mit Ultraschallsensorik



Nach Einschalten der Spannungsversorgung wird dieser Roboter aktiviert, indem man seinen Lichtsensor abdunkelt. Daraufhin setzt sich der Legoino in Bewegung. Erfasst sein Ultraschallsensor ein Objekt vor ihm, dann ändert er seine Mundwinkel im Display auf „traurig“ und weicht diesem Hindernis aus.

Die Arduino-Software hierzu heißt: `LegoinoFinalOhneSmartphone`.

3.2.2 Beispiel: Legoino als Smartphone-gesteuerter Roboter



Dieser Arduino baut über sein BT-Modul eine Verbindung zu einem Android-Endgerät auf, wenn auf diesem eine entsprechende (vorhandene aber selbst modifizierbare) App gestartet wird.

Dann kann der Roboter durch Kippen des Android-Endgeräts gesteuert werden.

Zusätzlich können die LED, der Laserpointer und der Piepser des Legoino über die Android-App ein- und ausgeschaltet werden.

Der Ultraschallsensor des Legoino überwacht den Abstand zu einem möglichen Hindernis vor dem Roboter und gibt dessen gemessene Entfernung an die Android-App zurück.

Die Arduino-Software hierzu heißt: `LegoinoFinalMitSmartphone`, die Android-App `LegoinoBT`.

4 Realisierung

4.1 Auswahl der 180°-Servos

Aus Kostengründen und aufgrund der weitaus höheren Auswahl an 180°-Servos ist ein selbst durchgeführter Umbau dem Kauf volldrehender Servos zu bevorzugen.

180°-Modellbauservos (im Folgenden als „Servos“ bezeichnet) gibt es in einer großen Variantenvielfalt.

Grundsätzlich unterscheidet man zwischen analogen und digitalen Servos. Bei analogen Servos erfolgt die Regelung auf die durch ein PWM-Signal vorgegebene Sollposition durch einen analogen Schaltkreis dessen Bestandteil ein Drehpotentiometer ist, das mechanisch mit der Servoachse verbunden ist⁸. Für den Umbau in volldrehende Servos wurde nur diese Servovariante in Betracht gezogen, da hier - wie nachfolgend beschrieben - nur die mechanische Verbindung zwischen Servoachse und Potentiometer gekappt werden muss.

Servos werden spezifiziert nach Drehmoment und Stellgeschwindigkeit, wobei letztere Eigenschaft für einen Legoino praktisch nicht relevant ist. Je höher das Drehmoment ist, desto „stärker“ werden die daraus gebauten Antriebe. Andererseits steigt die Stromaufnahme eines Servos mit dessen Drehmoment, was wie in Abschnitt 4.4 dargestellt Probleme mit der elektrischen Ansteuerung nach sich zieht.

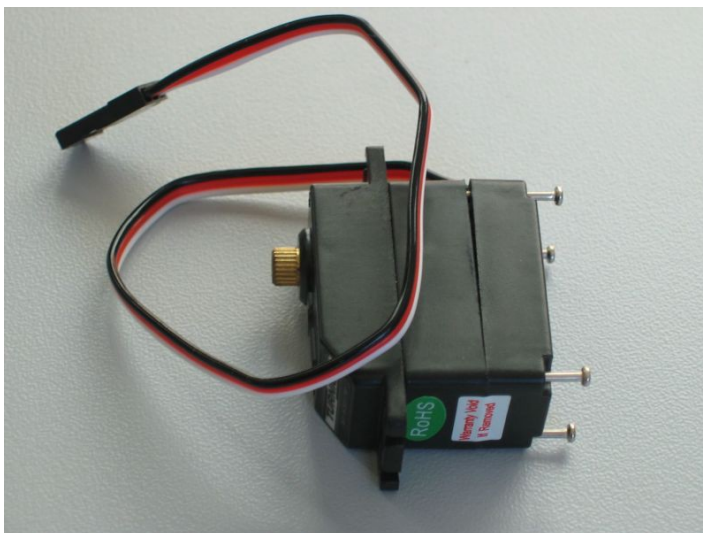
Ein weiteres Qualitätskriterium von Servos ist deren Getriebebauart: Hier unterscheidet man zwischen Kunststoff- und Metallgetrieben. Servos mit Kunststoffgetriebe sind leichter und preiswerter, Servos mit Metallgetriebe sind robuster. Daher wurde für den Legoino ein Metallgetriebeservo ausgewählt:

Es handelt sich hierbei um den Servo S3317M des Lieferanten hobbyking.com.

4.2 Umbau des S3317M Servos auf volldrehende Funktion

Nachfolgend ist an einem speziellen Servo dessen Umbau auf volldrehende Funktion dargestellt. Bei anderen Servos kann der Umbau evtl. ganz anders funktionieren. Sie finden im Internet oder in Fachbüchern [4] verschiedene weitere Umbauanleitungen zu jeweils anderen Servotypen.

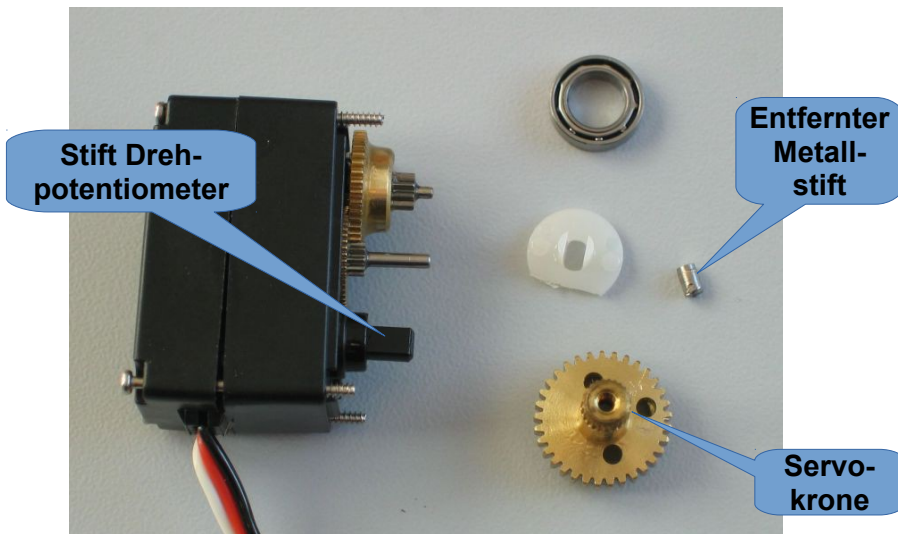
Öffnen Sie das Servogehäuse durch Lösen der vier Schrauben auf der Unterseite des Servos.



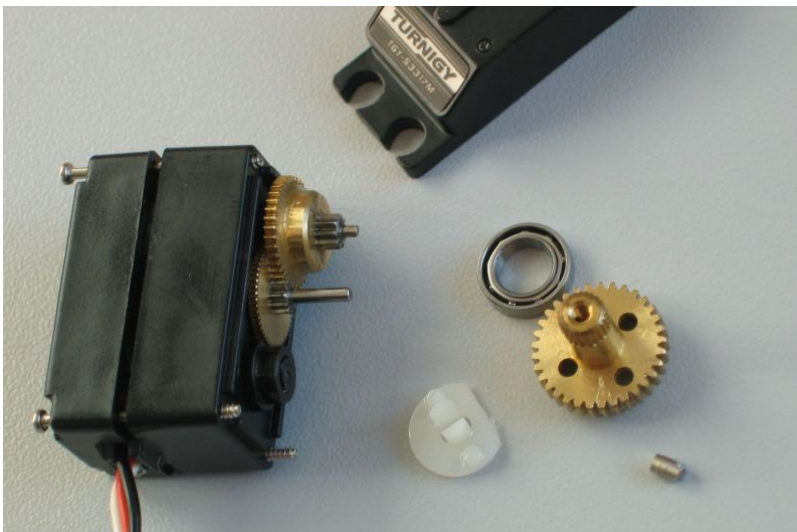
Entfernen Sie das Zahnrad (auf der Anschlussseite des Servos) mit der Servokrone inkl. der weißen Adapterscheibe und dem Lager.

⁸ Siehe z.B. www.de.wikipedia.org/wiki/Servo.

In dem entfernten Zahnrad befindet sich ein kleiner Metallstift, der den Winkel der Servoachse auf 180° begrenzt. Entfernen Sie diesen Stift (z.B. mit einer Zange hinaus ziehen oder mit einem Dorn heraus schlagen).



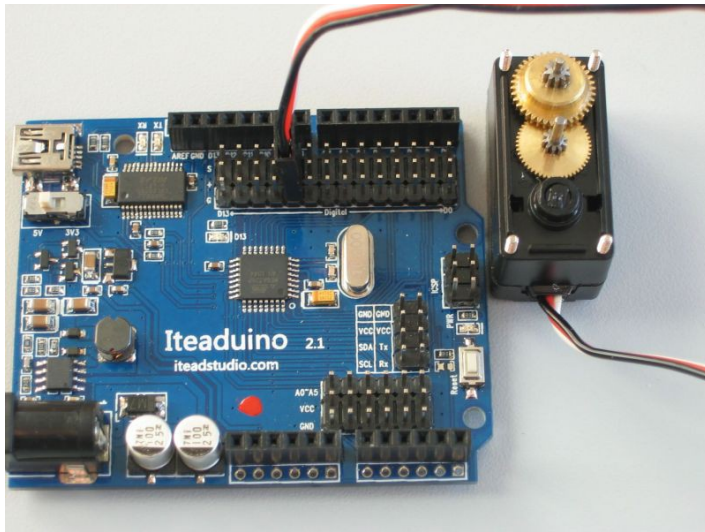
Kürzen Sie den Kunststoffstift des Drehpotentiometers (z.B. mit einem Seitenschneider abschneiden und den restlichen Überstand mit einem Cuttermesser kürzen), so weit, dass dieser vom wieder aufgesetzten Zahnrad nicht mehr mitgenommen wird.



Bevor Sie das Zahnrad mit der Servokrone wieder aufsetzen, müssen Sie den Servo über das Potentiometer abgleichen. Damit ist Folgendes gemeint:

Wenn der Servo das PWM-Signal für die Mittelstellung (90°) erhält, dann soll der (voll-drehend modifizierte Servo) stillstehen.

Diesen Abgleich erreichen Sie über folgende Schritte:



Laden Sie das Programm `ServoEinstellen` auf den Iteduino. Auf dem Digitalpin 4 des Iteduino wird nun ein 90° Stellsignal erzeugt.

Schließen Sie den Servo an Digitalpin 4 des Iteduino an.

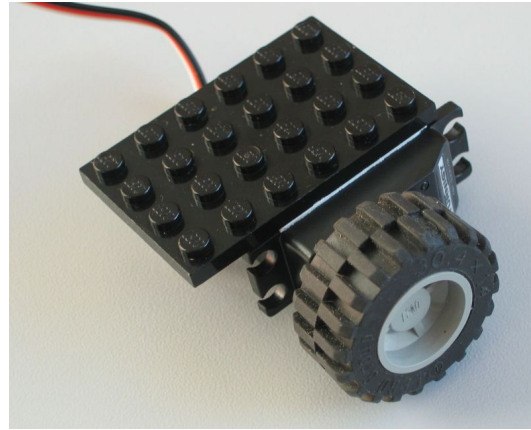
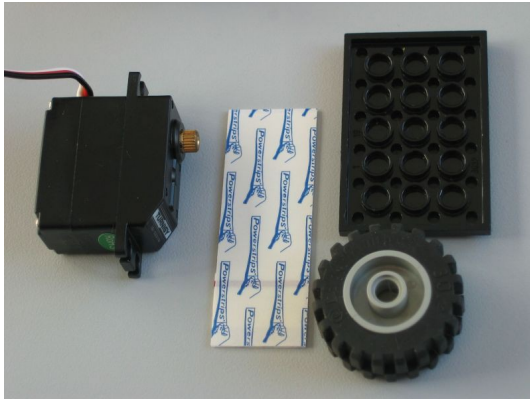
Der Antrieb unter dem oberen Zahnrad (siehe Bild oben) gegenüber dem Zahnrad mit der Servokrone wird sich nun mehr oder weniger schnell drehen. Stecken Sie einen Elektronschraubendreher in den zuvor gekürzten Stumpf des Potentiometerstifts und drehen Sie diesen so weit, dass die Bewegung zum Stillstand kommt. Haben Sie dies erreicht, dann ist der Servo abgeglichen.

Nun müssen Sie diesen Abgleich mechanisch fixieren. Dafür tropfen Sie etwas Sekundenkleber in den Spalt zwischen den Potentiometerstift und dessen Lager, damit sich der Stift nicht mehr drehen kann.

Später durch den Zusammenbau des Servos und bei anderer Temperatur wird sich dieser Abgleich jedoch noch leicht ändern, so dass der Servo trotz Nullsignal sich noch ganz langsam dreht.

Dieser Effekt ist prinzipiell nicht zu verhindern. Er lässt sich wie in Abschnitt 6.1 aufgeführt über Software beheben falls gewünscht.

4.3 Adaption von Lego-Rädern an die Servos



Der Servo S3317M besitzt zufällig eine Servokrone, die optimal in Bohrung eines einfachen Lego Technik Rad eingepresst werden kann. Falls die Pressung nicht ausreichend fest ist, und die Servokrone im Rad durchdreht, kann dies über das Einlegen eines dünnen Papiers oder einer dünnen Folie behoben werden. Das Einkleben oder Verschrauben des Legorads ist nicht nötig.

Generell besitzen die Servokronen immer ein Innengewinde, auf dem sogenannte „Servohörner“ aufgeschraubt werden (Schrauben und Hörner liegen den Servos meistens bei). Wenn aufgrund eines zu kleinen Servokronendurchmessers die oben beschriebene Passung nicht funktioniert, dann kann das Legorad auch wie ein Servohorn mit einer Schraube befestigt werden. Dafür muss aber das Legorad evtl. in seiner Achse durchbohrt werden, d.h. die Adaption ist nicht mehr zerstörungsfrei.

Größere Lego-Technik-Räder besitzen eine Achse zum Einstecken der bekannten Stifte mit kreuzförmigen Querschnitt. Hier ist nur eine (nicht zerstörungsfreie) Montage auf die Servokrone über eine Schraube möglich.

Anschließend wird der Servo mit doppelseitigem Klebeband (z.B. Tesa Powerstripes, oder 3M VHB Klebeband) auf eine Legoplatte fixiert.

4.4 Ansteuerung der Motoren mit direkter Stromversorgung über den Arduino

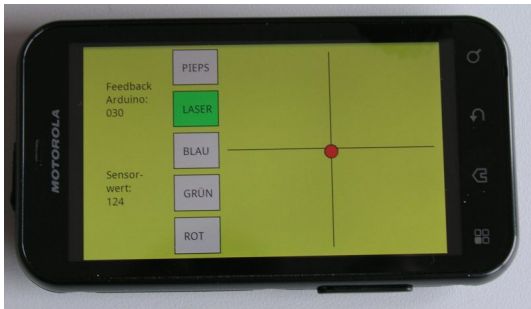
Generell ist es keine gute Idee, einen Antrieb direkt aus einem μC mit Strom zu versorgen. Denn die Ausgänge des μC können nur kleine Ströme liefern wohingegen selbst kleine Antriebe wie Servos teils hohe Spitzenströme zum Anlauf oder Richtungswechsel benötigen.

Dadurch kann die Spannung am μC so stark einbrechen, dass dieser einen Reset ausführt.

Normalerweise steuert man daher die Antriebe über eine separate Leistungselektronik an oder versorgt die Servos getrennt zum μC mit Spannung. Die erste Möglichkeit scheidet aus Kostengründen aus. Die separate Stromversorgung benötigt wie in Abschnitt 3.2 erklärt einen DC-DC-Wandler.

Beim Legoino ist eine rein softwarebasierte Lösung realisiert worden: Das Auftreten von Spitzenströmen an den Servos wird durch ein sanftes Anfahren und Bremsen der Servos verhindert. Wenn der Legoino den Befehl erhält loszufahren, dann werden die Servos nicht ruckartig auf die Solldrehzahl angesteuert. Die Software fährt in diesem Fall automatisch die Drehzahl stufenweise von Null auf den gewünschten Endwert hoch.

4.5 ArduSmartPilot App zur Steuerung des Legoino



Wird der Legoino mit einem Bluetoothmodul ausgestattet, so kann er über ein Android-Endgerät (z.B. Smartphone) gesteuert werden. Zusätzlich können vom Legoino aufgenommene Sensorwerte auf das Android-Endgerät übertragen und dort dargestellt werden.

Diese Funktionalität ist in der Arduinosoftware `LegoinoMitSmartphone` berücksichtigt. Dabei funktioniert die Kommunikation zwischen dem Iteaduino und dem Android-Endgerät identisch zum Projekt Ar-

duSmartPilot (siehe www.ardusmartpilot.de). Der einzige Unterschied ist, dass die Steuerbefehle des Android-Endgeräts nicht so oft in Servosignale umgesetzt werden, da die Antriebe des Legoino durch das sanfte Anfahren und Bremsen, siehe Abschnitt 4.4, wesentlich träger sind als beim ArduSmartPilot.

Wie beim ArduSmartPilot wird die Android-App über die Entwicklungsumgebung (IDE) Processing erstellt und unterscheidet sich lediglich durch ein anderes Icon und eine andere Beschriftung der Tasten für die Motorleistung. Über diese Tasten wird beim Legoino nicht die Motorleistung, sondern eine RGB-LED, ein Laserpointer und ein Piepser angesteuert.

Die Processing-Software hierzu heißt: `LegoinoBT`.

Alternativ kann die Android-App auch über eine andere IDE erstellt werden. Speziell für Projekte mit Kindern und Jugendlichen bietet sich dafür die grafische IDE „App Inventor“⁹ an. Eine wenn auch nur kurze Anleitung, wie man dabei die BT-Kommunikation einbindet findet sich in dem Arduinobuch (nur in der Auflage 2014!) von E. Bartmann .

5 Gesamtkosten des Legoino

Die Basis des Legoino bilden herkömmliche Legosteine, die nicht in die Kostenberechnung mit eingehen. Ebenfalls werden die Kosten für die Akkus nicht mit einbezogen. Kleinteile wie Jumperkabel sind pauschal mit 3 € einberechnet.

Die Preise für die meisten Elektronikkomponenten sind bewusst in US-Dollar angegeben, da sie sehr günstig direkt von den Herstellern in dieser Währung bezogen werden können.

Den Gesamtkosten liegt der Einfachheit halber ein Eurokurs zu US-Dollar von 1:1 zugrunde.

Exemplarisch sind hier die Kosten für die beide in Abschnitt 3.2 beschriebenen Legoinos aufgeführt.

9 Siehe www.appinventor.mit.edu/explore/.

5.1 Kosten Beispiel Legoïno ohne Smartphonesteuerung:

Bauteil:	Nötige Anzahl:	Gesamtkosten (USD = €):
Iteaduino UNO	1	17
Turnigy S3317M Servo	2	16
Ultraschallsensor HC-SR04	1	2
8x8 LED Matrix	1	10
RGB-LED	1	1
Lichtsensord (Photodiode)	1	1
Batteriehalter mit Steckerkabel	1	3
Kleinteile	1	3
Gesamtkosten Legoïno		53

5.2 Kosten Beispiel Legoïno mit Smartphonesteuerung:

Bauteil:	Nötige Anzahl:	Gesamtkosten (USD = €):
Iteaduino UNO	1	17
Turnigy S3317M Servo	2	16
Ultraschallsensor HC-SR04	1	2
Piepser	1	1
RGB-LED	1	1
Laserdiode	1	2
Batteriehalter mit Steckerkabel	1	3
BT-Modul HC-05	1	6
Kleinteile	1	3
Gesamtkosten Legoïno		51

6 Mögliche Probleme und deren Lösungsmöglichkeiten

6.1 Volldrehende Servos bewegen sich bei 90° Signal

Die käuflich erhältlichen volldrehenden Servos haben meistens eine kleine Stellschraube, mit der der Servostillstand abgeglichen werden kann. Damit wird gewährleistet, dass bei einem 90° Stellsignal der Servo nicht rotiert.

Bei selbst modifizierten Servos ändert sich die einmal eingestellte und mit Klebstoff fixierte Potentiometerposition mit der Zeit oder bei wechselnder Temperatur. Deshalb rotieren diese Servos manchmal auch bei einem 90° Signal noch ein wenig. Für die Funktion der Roboter ist dieser Effekt nicht wirklich problematisch.

Man kann dieses Problem beheben, wenn man in der Software einen Korrekturoffset für die Servoansteuerung einfügt: Zuerst probiert man mit einem gesonderten Arduino-Sketch aus, bei welchem Winkel der Servo tatsächlich still steht. Die Differenz dieses Winkels zu den idealen 90° wird dann als Korrekturwert verwendet.

6.2 Sketch kann nicht auf den Arduino übertragen werden

Dieses Problem kann verschiedene Ursachen haben. Allgemein kann dies durch die falsche Auswahl der COM-Schnittstelle oder des Arduino-Typs verursacht sein (beim Iteaduino muss unter „Board“ die Option „Arduino Duemillennove or Diecimilia“ gewählt werden). Speziell beim Legoino gibt es aber weitere spezifische Ursachen hierfür:

1. Die COM-Schnittstelle ist vom BT-Modul belegt. Lösung: Über Ausstecken des GND-Anschlusses das BT-Modul ausschalten.
2. Die Servos, helle LEDs oder eine andere Hardwareerweiterung hat einen zu großen Stromverbrauch, so dass die USB-Stromversorgung zusammenbricht. Lösung: Stromversorger temporär ausstecken bzw. den Legoino parallel über die Batterien versorgen.

6.3 Arduino geht beim Betrieb des Roboters unkontrolliert in den Reset-Zustand

Dies passiert z.B. bei ruckartigem Anfahren der Servos, da diese in diesem Moment einen so großen Stromverbrauch haben, dass die Batteriespannung zu sehr einbricht. In den Legoino-Programmen `LegoinoFinalMitSmartphone` und `LegoinoFinalOhneSmartphone` werden daher die Drehzahl der Motoren immer nur stufenweise erhöht.

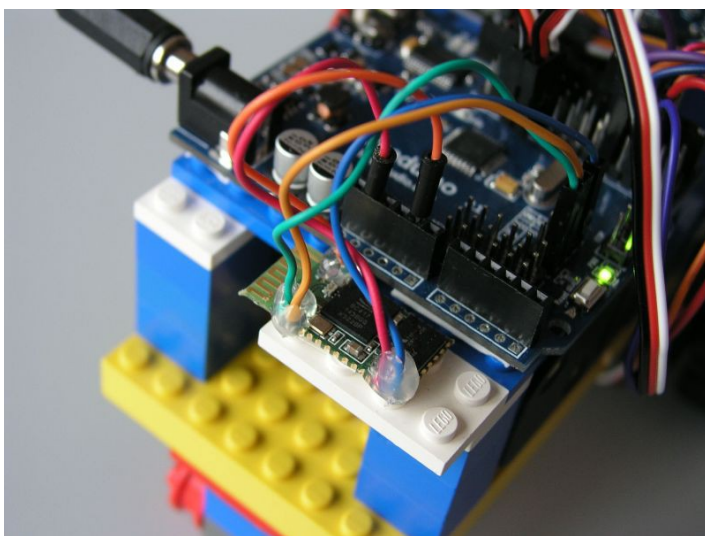
Hilft dies nicht, dann sollten kleiner Servos verwendet werden, oder die Servos sollten nicht über den Arduino sondern direkt von der Batterie ihre Versorgungsspannung erhalten. Damit die Servos durch eine zu hohe Spannung (>6 V) nicht zerstört werden, wird ein DC-DC-Wandler (zusätzlich zu dem auf dem Iteaduino vorhandenen) benötigt, der die Batteriespannung auf 5 V reduziert.

6.4 Android-App funktioniert nicht

Schauen Sie am besten in der Dokumentation des ArduSmartPilot-Projekts¹⁰ nach. Die Kommunikation zwischen dem Android-Endgerät und dem Arduino ist dort absolut identisch.

Die häufigsten Fehlerursachen sind: RX und TX beim Anschluss des BT-Moduls vertauscht, BT ist beim Android-Endgerät nicht eingeschaltet, in der Android-App ist nicht die richtige MAC-Adresse des BT-Moduls eingetragen.

6.5 Bluetoothmodul kann nicht parametrieren werden



Die Inbetriebnahme des BT-Moduls HC-05 ist manchmal recht hakelig. Zum Glück gibt es hierfür aber einen großen Fundus hilfreicher Informationen im Internet sowie ein ausführliches Datenblatt¹¹.

Über das Parametrieren des Moduls kann man dessen Namen und dessen Baudrate einstellen. Dafür muss TX des Moduls mit TX des μ C verbunden werden. Das Gleiche gilt für die RX Pins. Zuerst muss das BT-Modul mit Spannung versorgt werden (Achtung! Unbedingt 3,3 V und nicht 5 V verwenden!). Dann muss der KEY Pin des Moduls auf 3,3 V gelegt werden (normalerweise ist er mit GND verbunden).

¹⁰ Siehe www.ardusmartpilot.de.

¹¹ Siehe „HC Serial Bluetooth Products: User Instruction Manual“.

Jetzt sollte man in der Lage sein, mit einem Terminalprogramm mit dem Modul zu kommunizieren. Hierfür beim Terminalprogramm die zuletzt beim BT-Modul eingestellte Baudrate und CR und NL als Abschlusszeichen verwenden.

Zeigen sich im Terminalfenster seltsame Zeichen, dann stimmt die Baudrate nicht. Kommt gar keine Kommunikation zustande, dann liegt dies entweder an den Abschlusszeichen oder an der Verkabelung bzw. an der Reihenfolge der Verkabelung (= Reihenfolge Spannungsversorgung).

Wenn Sie sich nicht zutrauen wie im Bild oben, ein „nacktes“ HC-05 Modul mit abgeschnittenen Jumperkabeln zu konfektionieren, dann verwenden Sie besser eine fertig konfektionierte Version dieses Moduls (siehe Abschnitt 8.1).

6.6 Ich möchte meinen vorhandenen Arduino UNO statt des Iteduino verwenden

Ein Arduino Uno kann problemlos anstatt des Iteduino verwendet werden. Die Software ändert sich dadurch nicht. Es sind lediglich zwei Änderungen bei den elektrischen Anschlüssen nötig:

6.6.1 Spannungsversorgung der Servos

Da der Arduino UNO und die meistens Klons dazu keinen DC-DC-Wandler besitzt, kann die Servos nicht mehr über vom Arduino erzeugte 5 V Betriebsspannung versorgen. Entweder man versorgt die Servos direkt über den Akkupack, was aber die Gefahr birgt, dass sie durch eine zu hohe Spannung bei frisch aufgeladenen Akkus beschädigt werden. Um dem vorzubeugen bietet sich ein externer DC-DC Wandler¹² an, der zwischen dem Akkupack und der Versorgungsspannung der Servos geschaltet wird.

Diese Konfiguration hat dann den Vorteil, dass man die Servos nicht mehr langsam anfahren oder bremsen muss, um zu hohe Spitzenströme zu vermeiden.

6.6.2 Anschließen der Sensoren und Servos

Der Arduino UNO stellt pro Aus- bzw. Eingang immer nur ein Pin zur Verfügung. Beim Ansteuern der Servos ist dies z.B. der Pin, an dem das PWM-Signal anliegt. Bei Sensoren ist dies z.B. der Pin, der das Spannungssignal des Sensors aufnimmt.

Für die Spannungsversorgung und GND existieren beim Arduino UNO jedoch nur 1 bzw. 2 Pins. Diese müssen sich dann alle angeschlossenen Sensoren und Servos teilen. Mit ein wenig Lötarbeit, bei der mehrere Kabel auf einen Stecker geführt werden ist dies aber problemlos zu bewerkstelligen.

7 Fazit

Bei einem Lego Mindstorms Roboter kann man sich ganz auf den mechanischen Aufbau und die Robotersoftware konzentrieren. Dies wird allgemein als Segen empfunden, da man sich nicht um die Probleme auf der Ebene der Elektronikkomponenten kümmern muss: Wackelkontakte, falsch angeschlossene Sensoren oder ein Spannungseinbruch durch zu hohen Stromverbrauch sind normalerweise kein Thema bei Lego Mindstorms.

Solche „Herausforderungen“ treten jedoch beim Legoino auf, was eine etwas höhere Frustrationstoleranz vom Roboterbastler fordert.

Wer nur wenig Vorkenntnisse auf dem Gebiet der Elektronik hat, sollte daher einen kompetenten Ansprechpartner zu Seite haben, der seinen elektronischen Aufbau auf Fehler prüfen kann.

Generell eignet man sich beim Bau eines Legoinos sehr viel mehr Systemwissen und Detailwissen an:

- Welcher Mindstorms-Bastler hat je schon das Getriebe seiner Motoren zu Gesicht bekommen?

¹² Siehe z.B. Pololu 5V 600mA Step-Down Spannungsregler D24V6F5 in der Tabelle Abschnitt 8.1.

men?

- Welcher Mindstorms-Bastler kann nachvollziehen, wie eine Bluetooth-Kommunikation funktioniert?
- Welcher Mindstorms-Bastler weiß wie man einen I²C-Sensor anschließt?
- Welcher Mindstorms-Bastler schreibt seine eigene Android-App?

Der Legoino bietet Ihnen also neben dem geringeren Kosten und dem weitaus größerem Kreativitätspotential auch noch viel mehr Möglichkeiten, ein fundiertes Wissen auf dem Gebiet der Mechatronik zu erlernen!

8 Anhang

8.1 Hardwarekomponenten



In dieser Tabelle sind die zwingend erforderlichen Komponenten sowie optionale Erweiterungen wie Sensoren aufgeführt. Neben dem aktuellen Kaufpreis sind auch mögliche Lieferanten angegeben.

Beachten Sie bitte, dass es funktional gleiche Bauteile oft fertig konfektioniert mit steckbaren Kabeln oder als „Break Out Board“ gibt. Ein Break Out Board muss durch einfache Lötarbeit noch mit Anschlusskabeln versehen werden, ist dafür aber preiswerter.

Komponentenname	Bestellnummer	Hersteller (Internethop)	Vertrieb in Deutschland	Preis	Bemerkung
Arduino-Platine und Verbindungskabel:					
Iteduino UNO	IM130312001	imall.iteadstudio.com	www.exp-tech.de	\$17,00	Optimal geeignet, da anders als bei den übrigen Arduinos schon fertige Anschlüsse für Sensoren, Kommunikation und Motoren vorhanden.
3 Pin Dual-female Jumper Wire - 20cm	IM120530011	imall.iteadstudio.com	www.exp-tech.de	\$0,90	Für Anschließen Servos und einfacher analoger Sensoren.
3 Pin Dual-female to Grove 4 pin Converter Cable - 20cm	IM121025001	imall.iteadstudio.com	www.exp-tech.de	\$0,80	Für Anschließen einfacher analoger Sensoren der Grove-Serie.
4 Pin Dual-female to Grove 4 pin Converter Cable - 20cm	IM121025002	imall.iteadstudio.com	www.exp-tech.de	\$0,80	Für Anschließen digitaler Sensoren der Grove-Serie.
4 Pin Dual-female Jumper Wire - 20cm(Parallel)	IM120530012	imall.iteadstudio.com	www.exp-tech.de	\$1,20	Für Anschluss von BT-Modul oder digitale Sensoren. Je nach Anordnung der Pins muss die „Cross“-Variante verwendet werden.
4 Pin Dual-female Jumper Wire - 20cm(Cross)	IM120530013	imall.iteadstudio.com	www.exp-tech.de	\$1,20	Für Anschluss von BT-Modul oder digitale Sensoren. Je nach Anordnung der Pins muss die „Parallel“-Variante verwendet werden.
1 Pin Dual-Female Splittable Jumper Wire - 200mm (40Pcs Pack)	IM130725003	imall.iteadstudio.com	www.exp-tech.de	\$1,53	Daraus einzelne Jumperkabel und „3 Pin Dual-female Jumper Wire“ herstellbar.
Motor:					
Turnigy S3317M Servo	TGY-S3317M	www.hobbyking.com		\$8,00	Normaler hochwertiger und drehmomentstarker Analogservo. Einfach zum vordrehenden Servo modifizierbar. Servokrone passt optimal auf Legrad. Erhältlich auch in diversen deutschen Modellbauhops (z.B. Such nach „S3317M“ bei eBay).
Servo - Large Full Rotation ROB-09347	EXP-R05-124		www.exp-tech.de	\$13,28	Vollstehender Servo, der nicht modifiziert werden muss. Ruhepunkt kann extern eingestellt werden. Passt jedoch nicht optimal auf Legrad.
Sensoren:					
Electronic Brick - PIR Sensor Brick	IM120710015	imall.iteadstudio.com	www.exp-tech.de	\$7,00	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Sound Sensor/Microphone Brick	IM120710013	imall.iteadstudio.com	www.exp-tech.de	\$1,00	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Light Sensor Brick	IM120710017	imall.iteadstudio.com	www.exp-tech.de	\$1,50	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Touch Sensor/Button Brick	IM120710023	imall.iteadstudio.com	www.exp-tech.de	\$2,00	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Track Sensor	IM121012001	imall.iteadstudio.com	www.exp-tech.de	\$1,80	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Ultrasonic Ranging Module HC-SR04	IM120628012	imall.iteadstudio.com	www.exp-tech.de	\$2,00	Ein Pin muss separat eingeschlossen werden, die restlichen mit dem „3 Pin Dual-female Jumper Wire“
Grove - Single Axis Analog Gyro	SEN05091P	www.seeedstudio.com	www.exp-tech.de	\$9,00	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Grove - Infrared Reflective Sensor	WLS07061P	www.seeedstudio.com	www.exp-tech.de	\$4,90	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Grove - 80cm Infrared Proximity Sensor	SEN39046P	www.seeedstudio.com	www.exp-tech.de	\$13,90	Für Anschluss an Iteduino muss das mitgelieferte Anschlusskabel modifiziert werden.
Grove - 6-Axis Accelerometer&Compass	SEN05072P	www.seeedstudio.com	www.exp-tech.de	\$19,90	Digitaler I2C Sensor, Anschluss über „4 Pin Dual-female to Grove 4 pin Converter Cable“
Grove - 3-Axis Digital Accelerometer(±16g)	SEN04051P	www.seeedstudio.com	www.exp-tech.de	\$9,00	Digitaler I2C Sensor, Anschluss über „4 Pin Dual-female to Grove 4 pin Converter Cable“
Bluetooth-Steuerung:					
Electronic Brick - HC06 Serial Bluetooth Brick	IM120710006	imall.iteadstudio.com	www.exp-tech.de	\$10,00	Anschluss über „4 Pin Dual-female Jumper Wire“
Serial-Port Bluetooth Modul (Master/Slave) HC-05	IM120723009	imall.iteadstudio.com		\$6,00	„Nacktes“ Modul: Anschlüsse müssen selbst gelötet werden.
Sonstiges:					
3M Hochleistungs-Verbindungssystem VHB™ 4941P	C78412		www.voelkner.de	\$8,53	Doppelseitiges Klebeband alternativ zu Tesa Powerstrips.
Pololu 5V 600mA Step-Down Spannungsregler D24V6F5	EXP-R25-031		www.exp-tech.de	\$5,33	Wird benötigt, wenn Servos direkt über Akkupack versorgt werden.
Electronic Brick - Buzzer	IM120710008	imall.iteadstudio.com	www.exp-tech.de	\$0,90	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Big Button	IM120710022	imall.iteadstudio.com	www.exp-tech.de	\$0,90	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
Electronic Brick - Lighting Emitting Diode	IM130418002	imall.iteadstudio.com	www.exp-tech.de	\$0,80	Anschluss über „3 Pin Dual-female to Grove 4 pin Converter Cable“
128 x 64 OLED Module	IM130625003	imall.iteadstudio.com	www.exp-tech.de	\$5,00	Anschluss über Anlöten Stiftleiste und Jumperkabel.
Adafruit Mini 8x8 LED Matrix w/2C Backpack - Rot	EXP-R15-263		www.exp-tech.de	\$9,13	Anschluss über Anlöten Stiftleiste und Jumperkabel.
6xAA Battery Holder with DC 2.1 Power Jack	EXP-R07-034		www.exp-tech.de	\$2,80	
Laser Pointer Modul	438000060	www.hobbyking.com		\$2,00	!! Unbedingt Laserschutz beachten !!

8.2 Pinbelegung Beispiel Legoino ohne Smartphonesteuerung

Peripheriehardware	Anschlusspins
Servo Antrieb rechts	Digitalpins D9: S,+ und G (Servostecker).
Servo Antrieb links	Digitalpins D10: S,+ und G (Servostecker).
Ultraschallsensor	Digitalpins D11: Echo auf S, VCC auf +, GND auf G. Digitalpins D12: TRIG auf S.
RGB-LED	Digitalpins D0: Blau auf S. Digitalpin D1: R auf S, GND auf G.
Lichtsensord	Analogpins A0: VCC auf VCC, OUT auf A0, GND auf GND.
8x8 LED-Matrix	I ² C Pins: SCL auf SCL, SDA auf SDA, GND auf GND, VCC auf VCC.

8.3 Pinbelegung Beispiel Legoino mit Smartphonesteuerung

Peripheriehardware	Anschlusspins
Servo Antrieb rechts	Digitalpins D9: S,+ und G (Servostecker).
Servo Antrieb links	Digitalpins D10: S,+ und G (Servostecker).
Ultraschallsensor	Digitalpins D11: Echo auf S, VCC auf +, GND auf G. Digitalpins D12: TRIG auf S.
RGB-LED	Digitalpins D6: Blau auf S. Digitalpin D5: R auf S, Digitalpin D4: G auf S, - auf G.
Laser	Digitalpins D12: S auf S, - auf GND.
BT-Modul	Serial Pins: RX auf TX, TX auf RX, GND auf GND, VCC auf 3V3, KEY auf GND. Zur Parametrierung KEY auf 3V3 und RX/TX tauschen!
Piepser	Digitalpins D2: I/O auf S, VCC auf +, GND auf G.

Literaturverzeichnis

- 1: Igoe, Tom: Making Things Talk. O'Reilly, Sebastopol, 2011.
- 2: Margolis, Michael: Arduino Kochbuch. O'Reilly, Sebastopol, 2012.
- 3: Bartmann, Erik: Die elektronische Welt mit Arduino entdecken. O'Reilly, Sebastopol, 2014.
- 4: Karvinen, K. und T.: Make: Arduino Bots and Gadgets. O'Reilly, Sebastopol, 2011.