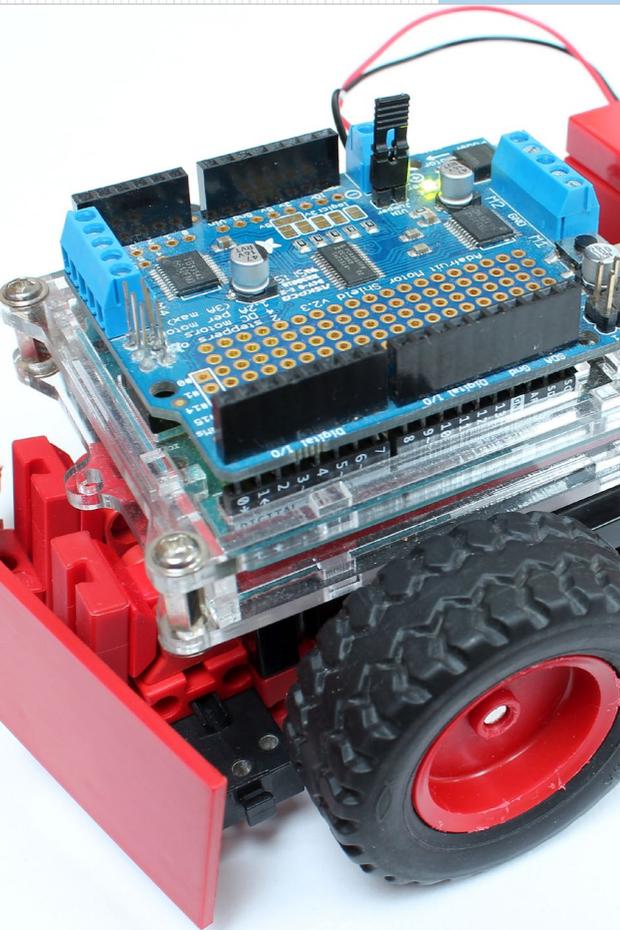


3

Der Buggy

Eine wesentliche Eigenschaft autonomer Roboter ist deren selbstständige Orientierung mithilfe von Sensoren und die daraus abgeleitete autonome Steuerung – ein Thema, bei dem Fischertechnik schon vor fast 40 Jahren die Nase vorn hatte, wie die Geschichte des Buggys beweist. In diesem Kapitel entwickeln wir einen eigenständig fahrenden, Arduino-gesteuerten Fischertechnik-Roboter, den wir – als kleine Reminiszenz an den ersten autonomen Fischertechnik-Roboter aus dem Jahr 1981 – liebevoll »Buggy« taufen.



3.1 Die Geschichte des Buggys

Das Baukastensystem von fischertechnik war schon immer seiner Zeit voraus – aber bei keinem Thema wird das deutlicher als bei der Entwicklung einfacher autonomer Roboter: Der erste fischertechnik-Buggy wurde im Jahr 1981 – dem »Geburtsjahr« des IBM PC und des Commodore VC 20 – von Mike Bostock und Max Townsend für das britische *Microelectronics Education Program* in Zusammenarbeit mit einem Team der BBC entworfen (Abb. 3–1). Die Firma Economatics aus Sheffield vertrieb ihn

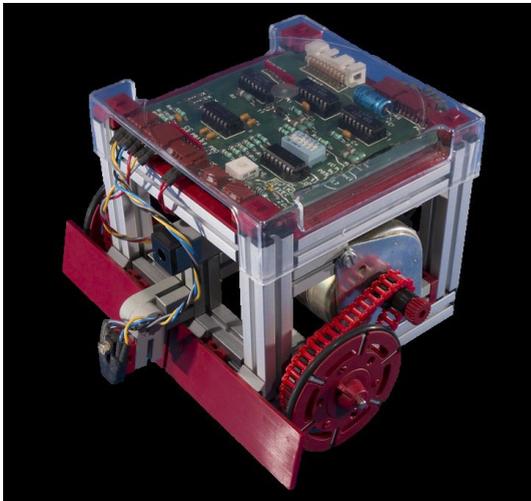


Abb. 3–1 »BBC Buggy« von Economatics (1983)

ab 1983 für den stolzen Listenpreis von 164,35 £. Der Buggy verfügte über eine Steuerungsplatine, zwei Schrittmotoren, eine Fozelle, zwei Mini-Taster und einen Barcode-Reader (IR-Sensor mit IR-Diode). Später wurde das Grundmodell des Buggys von Economatics um ein »Pen Kit« erweitert: einen per Elektromagnet absenkbaaren Stift, der in der Mitte des Buggys montiert wurde und ihn zu einem »mobilen Plotter« oder »Mal-Roboter« machte.

Die Ansteuerung des Buggys erfolgte über den Parallelport mit acht analogen Datenleitungen, über die die Byte-Sensordaten an den Hostcomputer übertragen

wurden. Der Buggy konnte einer Spur folgen und über die beiden Mini-Taster (»Bumper«) Hindernisse oder Wände erkennen; er konnte sich Sackgassen in einem Labyrinth merken und auf dem Rückweg den kürzesten Weg zum Ausgangspunkt wählen, Barcodes mit Kommandos lesen und Linien zeichnen. Im Januar 1984 reichte Max Townsend die Konstruktion des fischertechnik-Buggys in den USA als Patent ein, das im Oktober 1985 veröffentlicht wurde (Abb. 3–2).

Besonders an britischen Schulen war der kleine Roboter beliebt, da er sich mit dem Homecomputer *BBC Micro* (mit einem 2-MHz-6502-Mikroprozessor) in BASIC programmieren und steuern ließ. Dieser Homecomputer wurde von Acorn im Auftrag der BBC für die Serie »*The Computer Programme*« entwickelt und 1981 in den Schulen eingeführt. Am 28. Februar 1983 hatte der »BBC Buggy« einen Fernsehauftritt in der achten Folge der Serie »*Making the most of the Micro*« (*Everything under control*) [2], und die Zeitschrift »Your Computer« widmete dem Buggy in der Aprilausgabe 1983 die Titelseite und einen Leitartikel (Abb. 3–3) [3].

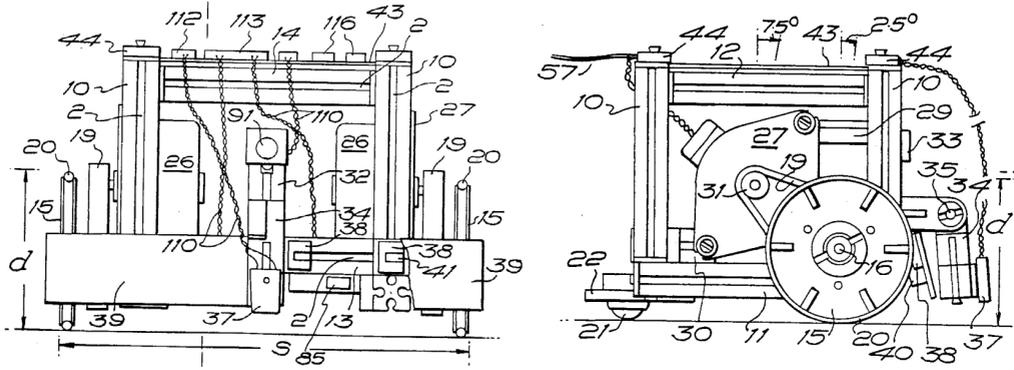
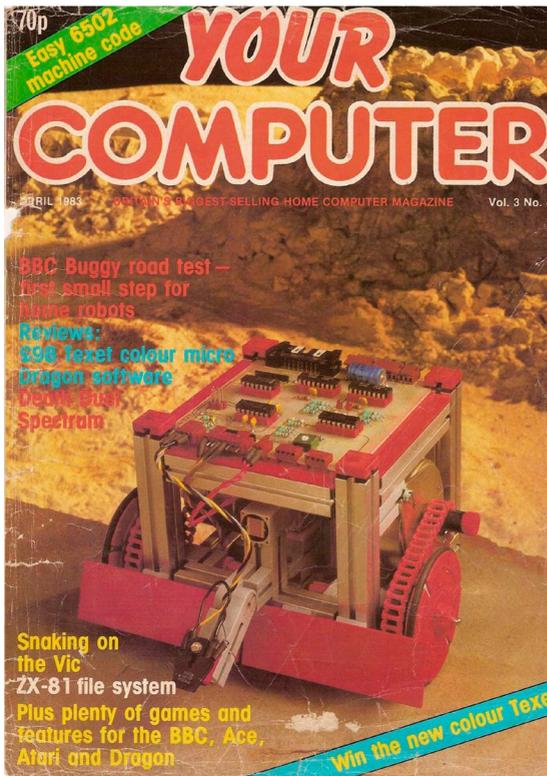


Abb. 3-2 Auszug aus US-Patent No. 4.548.584 (1985) [1]



Designed for the home. Built by screwdriver. Tested by Simon Beesley. Buggy — the world's first affordable robot.

BBC BUGGY

If YOU HAVE grown tired of all those video games and have exhausted your machine's programming potential you can now revive your interest in computing with the BBC Buggy. This is a three-wheeled vehicle which can be controlled by a BBC Micro and programmed to move in any direction, detect collisions, detect light, read a bar-code, and operate a pen-up/down mechanism. In short it is a robot — and at around £120 it is the first to come within the range of the home computer user rather than the electronics hobbyist.

The Buggy is the fruit of a collaboration between the BBC Computer Literacy Project and the Microelectronics Education Programme. After discussing ideas for the BBC's *Making the Most of the Micro* series with producer David Allen, Mike Bostock, Technology Manager for the MEP, built a prototype Buggy using Lego bricks. "Everyone wants to build a robot," he says, "and as the age of 3½ I finally built one!"

When the Buggy goes on sale this month it will come as a construction kit containing a chassis, two stepper motors, three types of sensor, control cables and electronic circuit boards. To go with it there is a tape with 13 programs, documentation, a Buggy handbook and assembly instructions.

Fortunately the review robot arrived ready-built so we did not need to test the claims of Buggy-maker Eiconomatics that the kit can be easily assembled in about two and a half hours using only a screwdriver.

The main body of the vehicle is a five-inch cube driven by two stepper motors which turn the front wheels. At the back there is a ball-bearing which acts as a balance wheel for the vehicle.

Using stepper motors greatly simplifies steering the Buggy since the motor can only be advanced by a fixed step at a time. This allows precise control of the vehicle's movement. Each motor has independent control over its respective wheel and the gearing is such that a single pulse to the motors rotates the Buggy by one degree.

It is comparatively easy to send the Buggy a specified distance forwards or backwards, or rotate it through any given angle. The two motors drive the vehicle at a rather steady pace with sufficient power for it to authoritatively break down obstacles such as books, in bulldozer fashion.

Top speed was measured at one and half miles per hour — hardly enough to trouble the man with the red flag. At the front is a split bumper with left and right microswitch collision detectors and above it a light detector — LDR. There is also a bar-code reader — BCR — mounted on a hinged arm which extends between the bumper and the LDR.

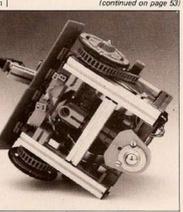
This consists of an infra-red light-emitting diode — LED — and photodiode which respectively send out and receive infra-red light. The BCR detects a black line by measuring the amount of light it reflects. My only criticism of the vehicle's design is that the BCR arm is inconveniently positioned. Although it can fold back it tends to prevent the bumpers below from registering a head-on collision.

Logo-style turtle

In the Buggy's centre of rotation there is a pen-up, pen-down mechanism which is mounted on the centre axle and controlled by an electro-magnet. This will permit the Buggy to be used as a Logo-style turtle. It is not quite as accurate as a dedicated Logo turtle but is £180 cheaper.

On the BBC Micro the Buggy is controlled through the user and analogue-in ports. Both the LDR and BCR return an analogue input proportional to the intensity of light measured. The collision detectors send a digital on/off signal to the user port.

Each of the user port's eight bits provides a control line. Four of the lines from the user



YOUR COMPUTER, APRIL 1983 51

Abb. 3-3 Titelseite und Leitartikel der Zeitschrift »Your Computer«, Ausgabe 4/1983 [3]

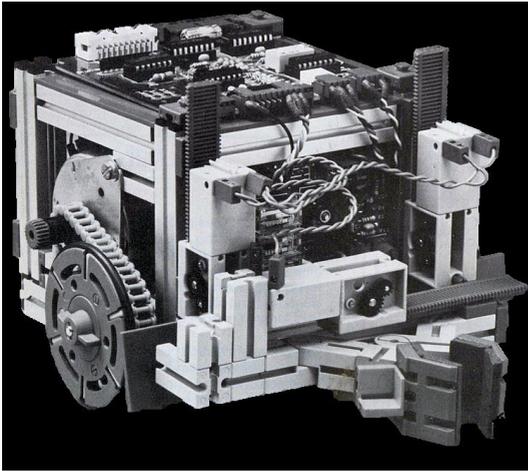


Abb. 3–4 BBC Buggy mit Greifer

Über 5.000 Exemplare des »Ur-Buggys« verkaufte die Firma Economatics allein in Großbritannien. Später gab es auch eine Version, die über das fischertechnik-Interface von anderen Homecomputern wie dem Commodore C64 oder dem Schneider CPC gesteuert werden konnte.

Als Erweiterung bot Economatics für 79 £ ein Ergänzungsset an, das aus einem Greifer bestand, der über drei Mini-Mots mit Hubgetrieben und einer separaten Elektronik steuerbar war. Damit konnte der Buggy gefundene Gegenstände ergreifen und anheben (Abb. 3–4). Ein Strombegrenzer erkannte, wenn der Greifer einen Gegenstand erfasst hatte, und stoppte automatisch den für den Greifer zuständigen Mini-Mot [4].

Heute kann man den BBC Buggy noch im *National Museum of Computing* in Bletchley Park in Funktion bewundern.

Eine weiterentwickelte Version des Buggys, der PIC-Buggy – etwas schlanker, mit S-Motoren und einfachen »Bumpers« –, wurde bis zur Insolvenz der Firma Economatics im Jahr 2008 vertrieben (Abb. 3–5).

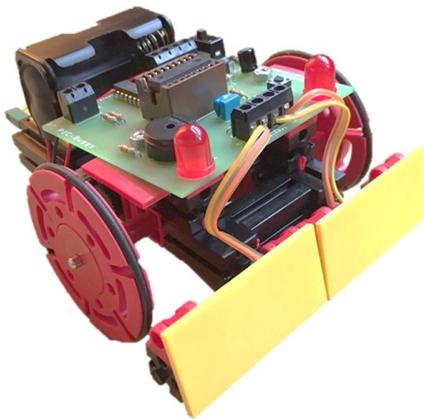


Abb. 3–5 Economatics PIC-Buggy
(ca. 2003, Foto: Dirk Uffmann)

Anders als der Ur-Buggy funktionierte der PIC-Buggy auch »offline«, d.h. ohne Verbindungskabel zum PC: Die Programmierung der Steuerungsplatine erfolgte über einen EPROM-Brenner [5]. Die Programme wurden dafür in einer der fischertechnik-Einsteigersprache ROBO Pro ähnlichen Programmierumgebung, dem PIC-Logicator, als grafisches Flussdiagramm entwickelt.

Ein diesem PIC-Buggy sehr ähnliches Robotermodell findet sich unter der Bezeichnung »Turtle« (Schildkröte) in vielen fischertechnik-Computing-Kästen, vom Baukasten »Computing Experimental« aus dem Jahr 1987 (mit einer 20:1-Untersetzung, Abb. 3–6) über das »Experimentierbuch Profi Computing« von 1991 [6], das Cornelsen-Schulprogramm »Experimenta Computing« (1994, Abb. 3–7), die Turtle-Bauanleitung aus dem Jahr 1996 mit Impulszahnradern (100 Impulse je Radumdrehung, Abb. 3–8) bis zum aktuellen Baukasten »Mini Bots«, der im Jahr 2015 erschien (Abb. 3–9).

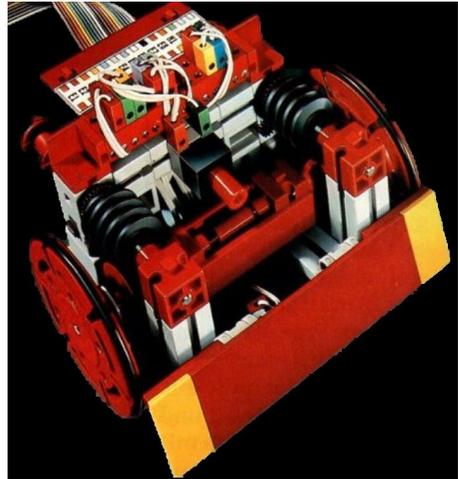
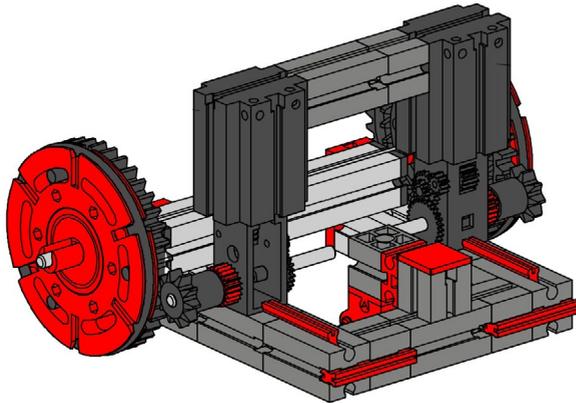


Abb. 3-6, 3-7: Modell »Turtle« aus den Baukästen »Experimenta Computing« (1994), links und »Computing Experimental« (1987), rechts [7]

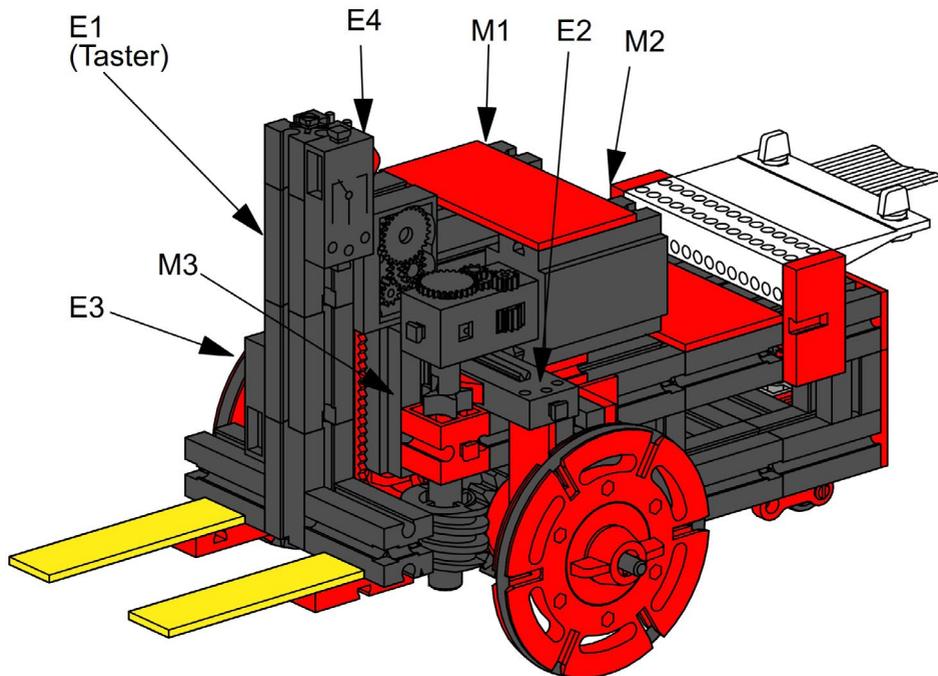


Abb. 3-8 Modell »Turtle« aus einer Bauanleitung des Jahres 1996



Abb. 3–9 Mini Bots (2015, Foto: fischertechnik GmbH)

Das im Folgenden vorgestellte Buggy-Modell, das uns durch dieses Kapitel begleiten wird, greift einige Konstruktionselemente dieser historischen Vorgänger auf. Im Verlauf des Kapitels werden wir unseren Buggy mit verschiedenen Sensoren ausstatten und ihm damit einige autonome Fähigkeiten verleihen, die zum Experimentieren und Weiterentwickeln einladen sollen. Mit dem Arduino wird er so zu einem autonom fahrenden Roboter mit ein wenig »Eigenintelligenz«.

3.2 Das Buggy-Basismodell

Mechanischer Aufbau

Unser kleiner Buggy basiert auf einer ähnlichen Grundkonstruktion wie die zahlreichen historischen Buggys: Zwei Antriebsmotoren und ein Stützrad sorgen für eine hohe Wendigkeit, und zwei »Bumper« an der Frontseite ermöglichen ihm, Hindernisse zu erkennen. Damit er wenig Eigengewicht transportieren muss und leicht nachgebaut werden kann, verwenden wir möglichst wenige Bauteile.

Als Antriebsmotoren nutzen wir – wie beim PIC-Buggy – zwei fischertechnik-S-Motoren (32293), auch als »Minimotor« bezeichnet, mit U-Getriebe (31078) und einer Untersetzung von ca. 56:1. Sie haben mit 0,4 Ncm ein deutlich größeres Drehmoment als die z.B. in den Mini Bots eingesetzten XS-Motoren (0,15 Ncm). Wir verbauen sie so, dass sie zugleich tragende Elemente unserer Fahrzeugkon-



struktions werden: Über die seitlichen Nuten verbinden wir die beiden Motoren mit zwei Federnocken stabil miteinander. In die beiden U-Getriebe setzen wir je eine U-Achse 40 mit Zahnrad Z28 (31064) als Antriebsachse ein (Abb. 3-10).

Die beiden vorderen Stoßdämpfer konstruieren wir aus zwei liegend montierten und über einen BS 7,5 verbundenen Mini-Tastern (37783), auf denen je ein BS 7,5, ein Gelenkwürfel (31426/31436) und darauf ein BS 15×30×5 mit Nut und Zapfen sitzen, die die Taster auslösen. In die Nuten der BS 15×30×5 werden je ein Winkelstein 15° geschoben und daran je eine Bauplatte 30×45 mit dem äußersten Zapfen befestigt (Abb. 3-11).



Abb. 3-10 S-Motoren mit U-Getriebe und U-Achse 40 als Buggy-Chassis

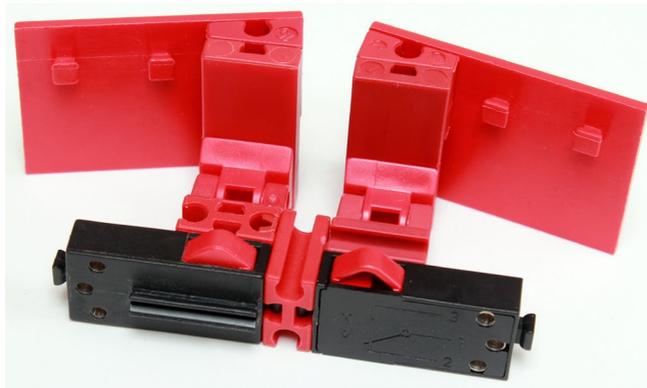


Abb. 3-11 Bumper aus zwei Mini-Tastern mit »hochgeklappten« Bauplatten

Die Bumper-Einheit verbinden wir stabil über das in Abb. 3-12 gezeigte Verbindungselement mit der Antriebseinheit. Zuletzt schieben wir in die oberen Nuten der beiden BS 15 je einen BS 5 und »verriegeln« damit die Verbindung mit den U-Getrieben (siehe Abb. 3-13 und 3-14).



Abb. 3-12 Verbindungselement (zwischen Bumper und Antriebseinheit)

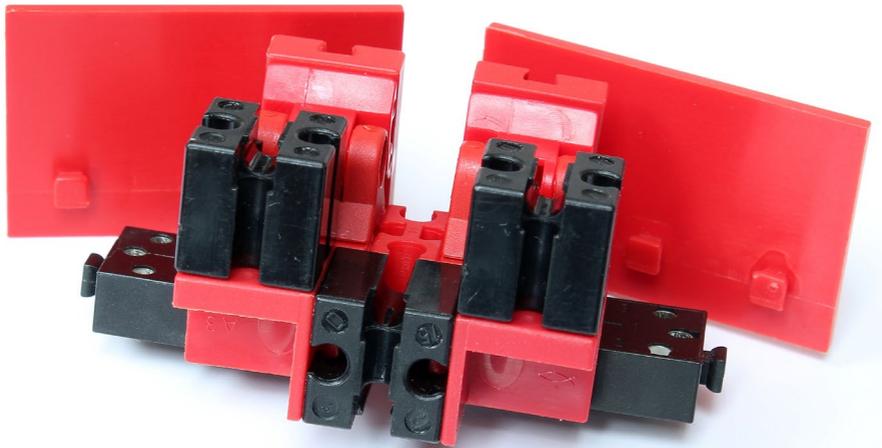
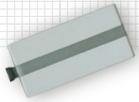


Abb. 3-13 Bumper-Einheit mit Verbindungselement

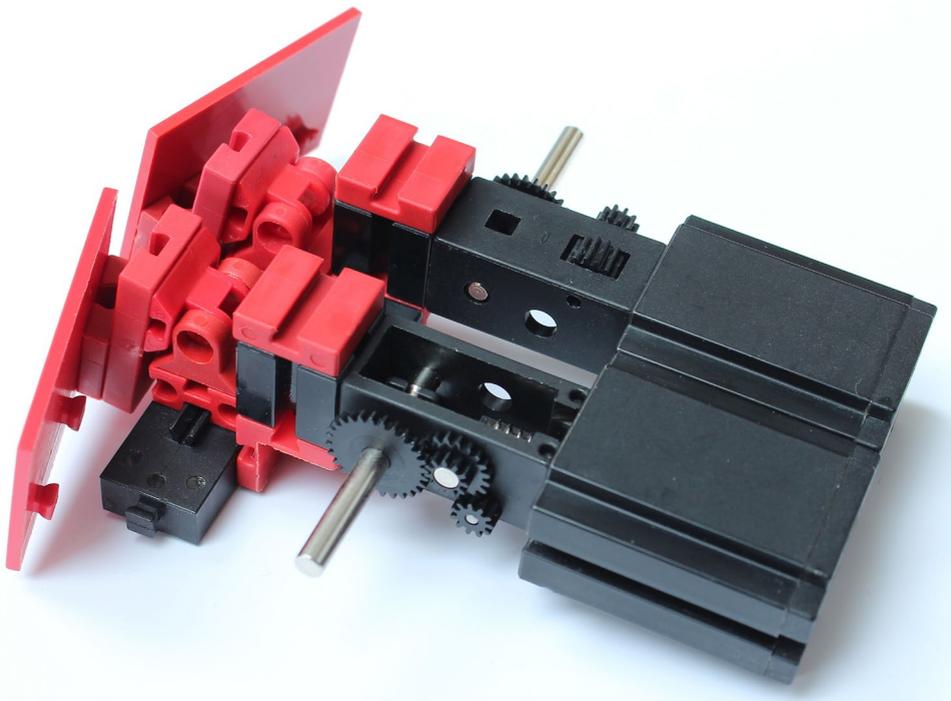


Abb. 3-14 Gesamtansicht der Antriebseinheit mit Bumpen



Abb. 3–15 Bereifung – Traktor- (links) oder einfache Reifen 50 (rechts)

Auf die U-Achsen 40 montieren wir nun die Räder des Buggys. Dafür eignen sich die Felgen 30 (32883), auf die entweder die Reifen 50 (32913) oder die Traktorreifen 50 (106767) bzw. 60 (121661) aufgezogen werden (Abb. 3–15).

Räder mit größerem Durchmesser haben den Charme, dass sie leichter kleinere Hindernisse überfahren können. Wohl auch deshalb wurden in den früheren Buggy- und Turtle-Modellen von fischertechnik Drehscheiben 60 verwendet, bei denen zur Verbesserung der Haftung in die Nuten zwei O-Ringe (Dichtungsringe) 54×3 als Bereifung eingesetzt wurden. O-Ringe 54×3 (35678) sind noch als fischertechnik-Einzelteil erhältlich, finden sich aber nur in wenigen (ausgelaufenen) Baukästen – daher haben wir unseren Buggy mit einer normalen Fahrzeugbereifung ausgestattet.

Hinten, am Ende der Antriebseinheit, fehlt uns noch ein »Stützrad«, das es dem Buggy erlaubt, wie seine Vorfahren auf der Stelle zu drehen. Dafür gibt es zahlreiche Konstruktionsmöglichkeiten: Der »Klassiker« (Abb. 3–16, oben rechts), der sich auch in vielen fischertechnik-Bauanleitungen findet, ist ein Rad 14 (36573) auf einer Rastachse 20 (31690) in einem Rollenbock (32085). Eine hübsche Variante ist es, das Rad 14 zwischen zwei Lenkhebeln (38473) einzuspannen (Abb. 3–16, oben links).

Wer über einen O-Ring 17×4 verfügt, kann damit eine Seilrolle $d=21$ (35797) bereifen und die Rastachse 20 in zwei Kupplungsstücken (38253) lagern (Abb. 3–16, unten) – sicher die optisch ansprechendste dieser drei Konstruktionsvarianten. Allerdings sollte man bei dieser Variante die beiden Vorderräder mit 60er-Reifen ausstatten, sonst steht der Buggy hinten ein wenig hoch.



Abb. 3-16 Varianten für die Konstruktion des Stützrads



Abb. 3-17 Verbindungselement für Stützrad

Das Stützrad wird mit der Aufnahmeachse (31124) in einen Baustein 15 mit Ansenkung (32321) eingesetzt. Den BS 15 mit Ansenkung stecken wir in die seitliche Nut eines BS 30, der mit einem Verbindungsstück 30 (31061) quer hinter den Motoren angebracht wird (Abb. 3-17).

Richtig montiert dreht sich das Stützrad nun frei unter dem Buggy-Chassis. In die obere Nut des BS 30 schieben wir nun noch den Zapfen eines Bausteins 15 × 30 × 5 mit Nut und Zapfen (35049), sodass dieser über dem BS 15 mit Ansenkung zu liegen kommt (Abb. 3-18). Dessen Nut dient uns als Halterung für einen fischertechnik-Akku. Alternativ können wir daneben einen zweiten Baustein 15 × 30 × 5 mit Nut und Zapfen aufschieben und darauf einen Batteriekasten



(Abb. 3–20). Schließlich befestigen wir an der seitlichen Nut des linken Motors unseres Buggys noch einen Mini-Taster, mit dem wir später das Programm des Buggys starten und z.B. Display-Anzeigen umschalten können (Abb. 3–18).

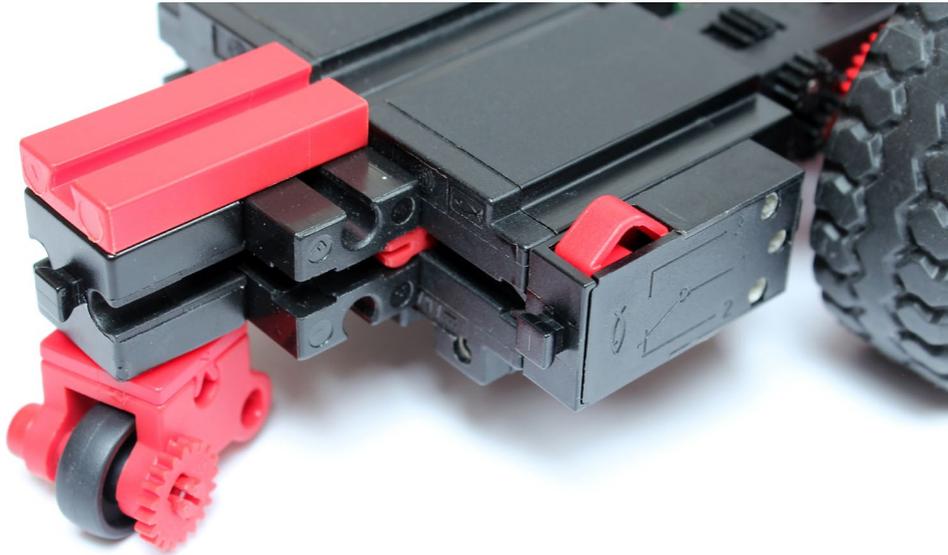


Abb. 3–18 Buggy-Chassis mit Stützrad, Starttaster und Akku-Halterung

Eine fischertechnik-Designer-Datei des Modells findet sich zum Download auf der Website zum Buch. Ausgehend von dieser Grundkonstruktion werden wir im Folgenden für die Ausstattung des Buggys mit dem einen oder anderen Sensor kleine An- oder Umbauten an unserem Buggy vornehmen. Und natürlich sind auch Konstruktionsvarianten zulässig – sie sind sogar ausdrücklich erwünscht.

Akku, Controller und Motorsteuerung

Jetzt müssen wir unseren Buggy noch mit einem Controller und einer Stromversorgung ausstatten. Bevor wir unseren Arduino über den beiden Motoren befestigen, schieben wir in die senkrechte Nut des mittleren BS 15 der Bumper-Halterung eine grüne oder rote em-Verteilerplatte (31327/31328), um darüber später die (knappen) GND-Anschlüsse des Arduino zu verteilen (Abb. 3–20). Alternativ zur em-Verteilerplatte gibt es für alle Pins des Arduino auf dem *Adafruit Motor Shield* je zwei Lötkontakte – wenn wir in die bisher ungenutzten, innen liegenden Kontakte Pinheader löten, können wir weitere GND-Pins dazugewinnen (Abb. 3–19).



Abb. 3–19 Zusätzliche Pinheader für 3,3V, 5V, 2xGND und Vin (9V)

Als Stromversorgung ergänzen wir einen Batteriekasten (32263) oder einen fischertechnik-Akku, den wir in die Halterung über dem Stützrad einsetzen. Der fischertechnik-Akku hält mit 1500 mAh deutlich länger durch als eine 9-V-Blockbatterie oder ein 9-V-Blockakku – aber mit ihm steigt auch das Gewicht unseres Buggys, denn der Akku wiegt knapp 210 g und verschiebt den Schwerpunkt über das Stützrad, das sich dadurch deutlich schwergängiger dreht. Auch hat er einen sichtlich größeren Platzbedarf (Abb. 3–20 und Abb. 3–21).



Abb. 3–20 Buggy mit Batteriekasten



Abb. 3-21 Buggy mit fischertechnik-Akku

Zur Montage des Arduino schieben wir die beiden äußeren Zapfen einer der beiden am Arduino-Gehäuse angebrachten Bauplatten 30×45 (Abb. 1-12) in die Nuten der beiden BS 5, die auf den BS 15 des Verbindungsstücks zwischen Bumpen und Motorblock aufgesteckt sind. Dabei achten wir darauf, dass der Strom- und der USB-Anschluss des Arduino nach hinten zeigen, damit die Anschlussklemmen für die Motoren später ebenfalls hinten liegen. Abb. 3-22 zeigt die Gesamtansicht unseres kleinen Buggys inklusive aufgestecktem *Motor Shield*.

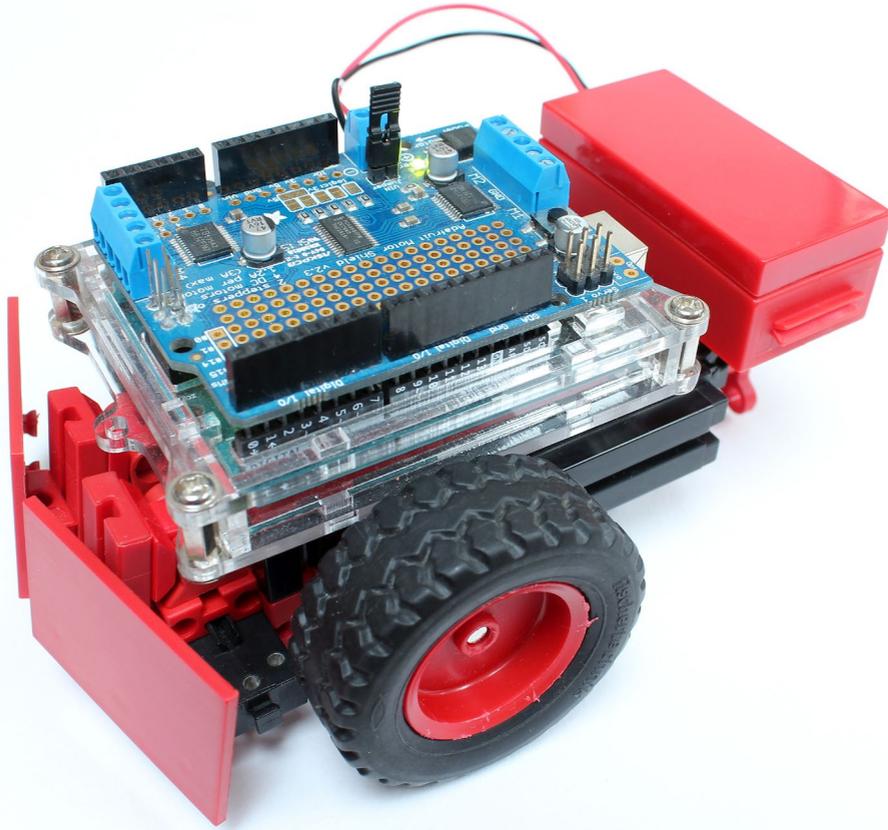


Abb. 3–22 Buggy-Basismodell mit Arduino Uno, Motor Shield und Batteriekasten

Jetzt müssen wir noch die Motoren, den Starttaster und die Stromversorgung an unseren Arduino anschließen. Der Anschluss der beiden S-Motoren ist einfach: Die beiden doppeladrigen Kabel der Motoren werden jeweils mit den mit »M1« (linker Motor) und »M2« (rechter Motor) markierten blauen Schraubklemmen des Motor Shield verbunden. Die korrekte Polung der Anschlüsse überprüfen wir später.

Das Kabel zur Verbindung des Arduino mit der Stromversorgung (fischertechnik-Akku bzw. Batteriekasten) gehört an die mit »(+« (rotes Kabel) und »(-« (grünes Kabel) gekennzeichneten Anschlüsse (gelber Kreis in Abb. 3–23). Damit das Motor Shield die Motoren über den externen 9-V-Anschluss versorgt, muss der »Vin-Jumper« (roter Kreis in Abb. 3–23) aufgesteckt werden.

Zur Nutzung des Mini-Tasters verbinden wir dessen mittleren Anschluss mit Pin A0, den offenen Ausgang mit einem GND-Pin. Die komplette Beschaltung zeigt Abb. 3–23.

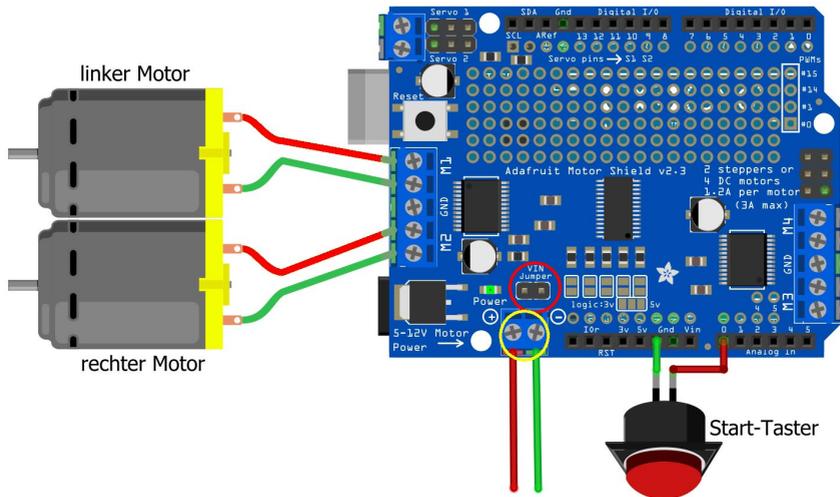


Abb. 3–23 Anschluss von S-Motoren, Starttaster und Stromversorgung (erstellt mit fritzing)

Damit uns der Buggy bei Softwaretests oder der Fehlersuche nicht davonfährt, können wir ihn – wie in einer Werkstatt – mit der in Abb. 3–24 gezeigten Konstruktion »aufbocken«: auf einer roten oder schwarzen V-Grundplatte 45×45 (36593/36596) mit zwei BS 15 (zwei Zapfen). Wer diese kleine Grundplatte nicht in seinem Sortiment hat, kann stattdessen auch eine schwarze Grundplatte 120×60 (35129) verwenden. Dazu schieben wir die Nuten der Motoren auf die beiden freien Zapfen der BS 15 – und schon hängen die Räder des Buggys frei in der Luft.

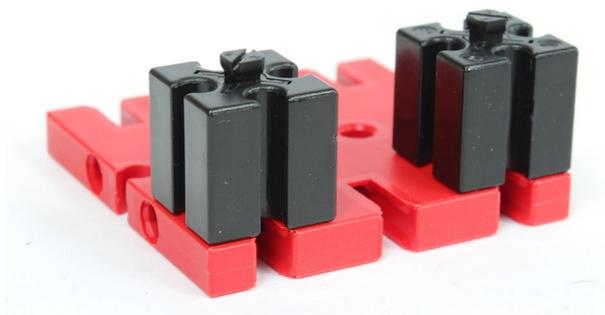


Abb. 3–24 Grundplatte zum »Aufbocken« des Buggys

3.3 Buggy-Steuerung

Die Steuerungsprogramme für unseren Buggy können wir nun auf unserem kleinen »Teststand« mit USB-Verbindung zum PC und einem Netzteil als Stromversorgung testen, bevor wir den Buggy autonom fahren lassen. Für einige der Steuerungen werden wir unseren Buggy ein wenig umkonstruieren, um zusätzliche Sensoren zu befestigen. Soweit verfügbar haben wir dabei fischertechnik-Sensoren verwendet, da sie sich einfacher und eleganter verbauen lassen und sie bei vielen Lesern bereits vorhanden sein dürften. Sie lassen sich aber oft durch ähnliche Sensoren aus dem Modellbau oder der »Maker-Szene« ersetzen. Bezugsquellen für solche alternativen Sensoren haben wir auf der Webseite zum Buch zusammengetragen.

Tanzender Buggy

Zunächst möchten wir unseren Buggy einem ersten Fahrtstest unterziehen. Der folgende kleine Sketch, mit dem wir den Buggy ein wenig »tanzen« lassen, zeigt, wie die Ansteuerung der Motoren funktioniert. Dabei können wir gleich testen, ob beide Motoren mit der richtigen Polarität angeschlossen sind: Wenn alles stimmt, fährt der Buggy nach dem Drücken des Starttasters in ständiger Wiederholung erst kurz vorwärts, dann rückwärts, dreht dann nach rechts, fährt wiederum vorwärts und rückwärts, dreht dann nach links, fährt vorwärts und rückwärts und sollte nach einer erneuten Rechtsdrehung wieder mehr oder weniger in der ursprünglichen Position stehen.

```

/*
 * "Tanzender Buggy"
 */

#include <Adafruit_MotorShield.h>

#define Start A0 // Starttaster
#define Speed 200 // Motorgeschwindigkeit
#define Move 1000 // Dauer der Bewegung in ms
#define Turn 200 // Dauer der Drehbewegung in ms

Adafruit_MotorShield MShield = Adafruit_MotorShield(0x60);
Adafruit_DCMotor *MotorL = MShield.getMotor(1); // linker Antriebsmotor
Adafruit_DCMotor *MotorR = MShield.getMotor(2); // rechter Antriebsmotor

void setup() { // Festlegung der Motorgeschwindigkeit

```