



Arduino-Bauteile

Erläuterungen zu den mitgelieferten Komponenten

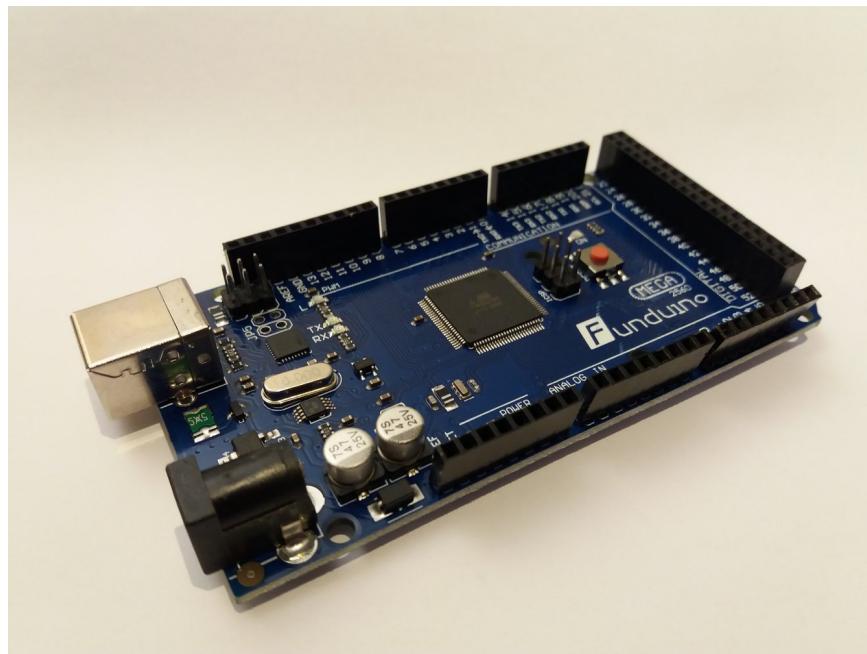
26. Februar 2018

Inhaltsverzeichnis

1 Allgemeine Bauteile	3
1.1 MEGA2560 R3 Mikrocontroller	3
1.2 9V-Batterieklemme	5
1.3 Breadboard	6
1.4 Kabel	6
1.5 Widerstände	7
1.6 Drehpotentiometer	8
1.7 Taster	8
1.8 Diode (1N4001)	9
1.9 LEDs	10
1.9.1 einfarbige LEDs	10
1.9.2 RGB-LED	11
1.9.3 IR-LED (IR-Sender)	12
1.10 Transistor (TIP120)	12
1.11 Relaiskarte	13
1.12 LC-Display (LCD Modul 1602)	14
2 Sensoren	15
2.1 Bewegungsmelder (HC-SR01)	15
2.2 Temperatursensor (TMP36GT9Z)	16
2.3 Tropfensensor	17
2.4 Feuchtigkeitssensor	18
2.5 Ultraschallsensor (HC-SR04)	19
2.6 Fotowiderstand	20
2.7 IR-Empfänger (+ Fernbedienung)	20
2.8 RFID-Kit (RFID-RC522, Chip MFRC522)	22
3 Aktoren	23
3.1 Servo	23
3.2 Schrittmotor (28BYJ-48) mit Treiberplatine (ULN2003)	24
4 Weitere Informationen und Links	25

1 Allgemeine Bauteile

1.1 MEGA2560 R3 Mikrocontroller



Der MEGA2560-Mikrocontroller ist der Kern jeder Schaltung und Programmierung. Über die Steckpins des Boards werden Spannungen ausgegeben, ausgelesen und über das USB-Kabel Code-Sketches hochgeladen. Zu beachten ist, dass nicht an jedem Pin alle Funktionen vorhanden sind. Analoge Eingangssignale können nur an den dafür vorgesehenen Pins (mit A gekennzeichnet) erfasst werden. Digitale Signale (Wert 0 (LOW) oder 1 (HIGH)) können an den übrigen Pins eingelesen und auch ausgegeben werden. Die Spannung beträgt dabei 0 V oder 5 V.

Eine analoge Spannungsversorgung ist an keinem Pin möglich. Über die Pulsweltenmodulation, kurz PWM, kann jedoch ein Ausgangssignal erzeugt werden, das einem analogen Signal entspricht. Dabei wird die Spannungsversorgung in kurzen Zeitabständen an- und abgeschaltet, sodass sich eine gemittelte Spannung zwischen 0 V und 5 V einstellt. Die Zeitabstände werden für eine bestimmte Spannung automatisch reguliert. Die vorhandenen Pins und ihre Funktionen sind in Tabelle 1 aufgelistet. In Tabelle 2 sind die wichtigsten Codes zur Steuerung der Pins gegeben. Weitere Informationen zum Mikrocontroller¹ sind auf der Arduino-Webseite zu finden.

¹<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Tabelle 1: Pins des MEGA2560-Mikrocontrollers

Anzahl	Pin	Verwendung	Spannung
54	Digital	Eingang, Ausgang	0 V, 5 V
15	Digital	Ausgang PWM	0...5 V
16	Analog	Eingang	0...5 V
2	Digital	Ausgang	5 V
1	Digital	Ausgang	3,3 V
4	Ground	Eingang	0 V

Tabelle 2: Codes zur Steuerung der Pins

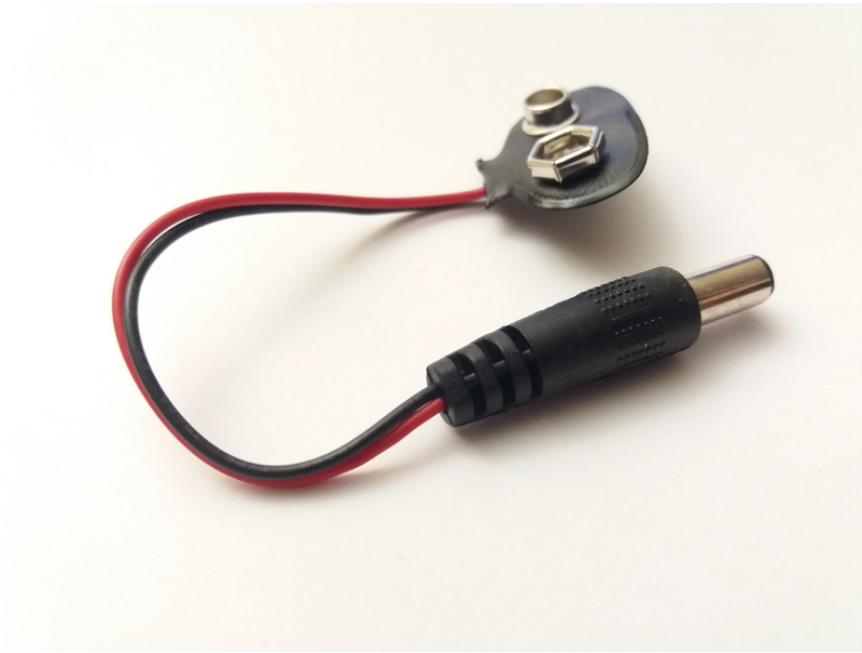
Code	Funktion
<code>void setup() {}</code>	Code-Teil im Sketch zur Festlegung des Pin-Verhaltens und anderer Befehle.
<code>void loop() {}</code>	Code-Teil im Sketch zur Aktivierung der Pins und Ausführung von anderen Befehlen.
<code>pinMode(13, OUTPUT);</code>	Weist Pin 13 an Signale auszugeben.
<code>pinMode(22, INPUT);</code>	Weist Pin 22 an Signale aufzunehmen.
<code>digitalWrite(22, HIGH);</code>	Schaltet Pin 22 an (5 V).
<code>digitalWrite(22, LOW);</code>	Schaltet Pin 22 aus (0 V).
<code>analogWrite(13, 127);</code>	Schaltet Pin 13 mit ca. 2,5 V an, entsprechend dem analogen Wert 127 (zwischen 0 und 255). Es muss ein digitaler Pin mit PWM verwendet werden!
<code>analogWrite(13, 0);</code>	Schaltet Pin 13 aus, der analoge Wert 0 entspricht dem digitalen Wert LOW. Es muss ein digitaler Pin mit PWM verwendet werden!
<code>digitalRead(13);</code>	Liest an Pin 13 den Wert 0 (0 V) oder 1 (5 V) aus.
<code>analogRead(A0);</code>	Liest an Pin A0 Werte zwischen 0 und 1023 (0...5 V) aus. Es muss ein analoger Pin verwendet werden!

Folgender Beispiel-Sketch ist als eine Übung mit Anleitung² auf der Funduino-Webseite zu finden:

```
void setup() {  
pinMode(13, OUTPUT);  
}  
  
void loop() {  
digitalWrite(13, HIGH);  
delay(200);  
digitalWrite(13, LOW);  
delay(200);  
}
```

Die auf dem Mikrocontroller fest integrierte LED an Pin 13 wird in Zeitabständen von 200ms (Befehl `delay(200);`) an- und abgeschaltet.

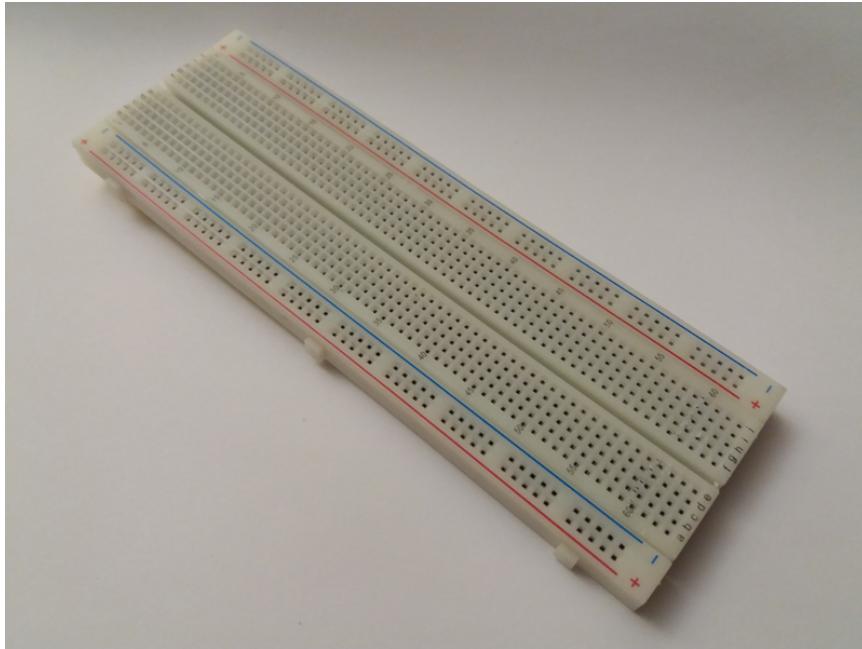
1.2 9V-Batterieklemme



Um den Mikrocontroller mit Strom zu versorgen, kann statt dem USB-Kabel auch eine 9V-Batterie verwendet werden. Diese wird über die Batterieklemme an den Mikrocontroller angeschlossen.

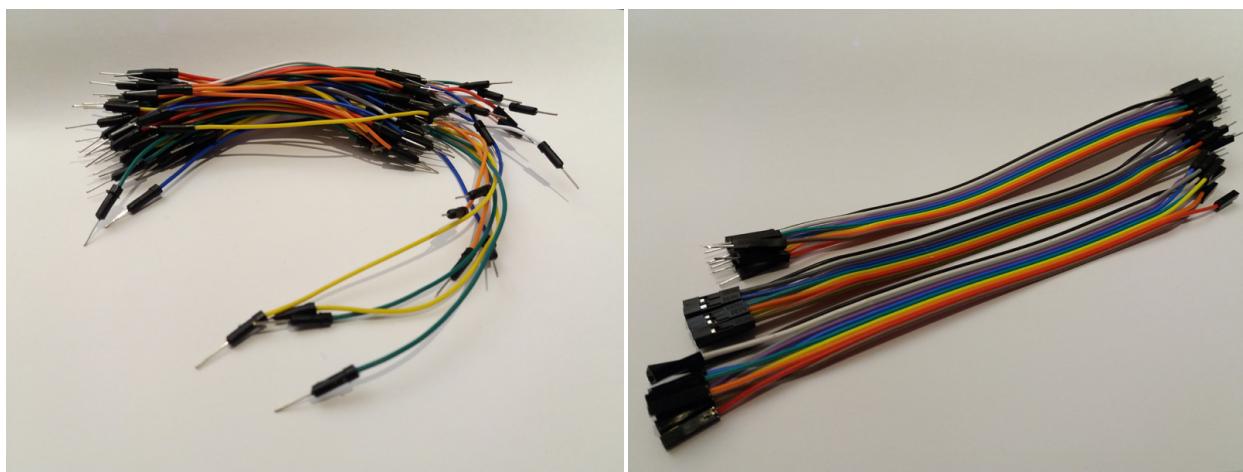
²<http://funduino.de/nr-1-blinkende-led>

1.3 Breadboard



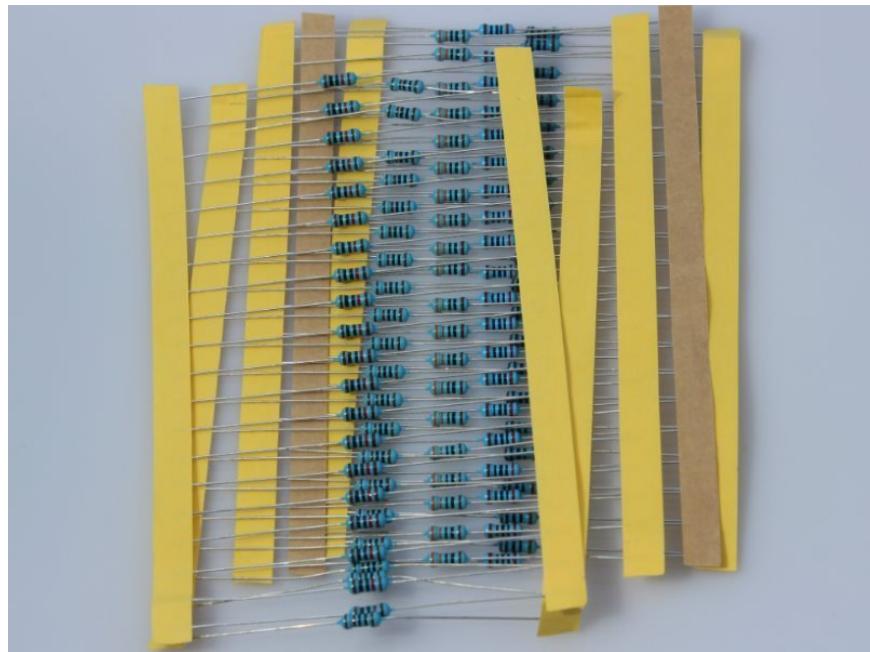
Mit der Steckplatine (*breadboard*) lassen sich verschiedene Bauteile zu Schaltkreisen zusammenstecken, ohne zu löten.

1.4 Kabel



Neben den 65 vereinzelten Kabeln (Bild links) in verschiedenen Farben und Längen (12, 16, 20, 25cm) enthält das Kit weitere Kabel (Bild rechts) mit besonderen Anschlüssen. Mit diesen können beispielsweise Sensoren direkt mit dem Mikrocontroller verbunden werden. Die Kabel haben Anschlüsse der Form weiblich/weiblich, weiblich/männlich oder männlich/männlich und sind jeweils 20 mal vorhanden.

1.5 Widerstände



Dem Bausatz liegen verschiedene Widerstände bei. Diese sind durch Farbkodierung gekennzeichnet. Welche Widerstände vorhanden sind, ist in Tabelle 3 dargestellt.

Der Gesamtwiderstand einer Schaltung aus mehreren Einzelwiderständen lässt sich bei Reihenschaltung nach

$$R_{ges} = \sum_{i=1}^n R_i \quad (1)$$

sowie bei Parallelschaltung nach

$$\frac{1}{R_{ges}} = \sum_{i=1}^n \frac{1}{R_i} \quad (2)$$

berechnen. Damit lassen sich auch aus einer begrenzten Anzahl an Widerständen verschiedene Gesamtwiderstände erreichen.

Tabelle 3: Farbkodierung der beiliegenden Widerstände

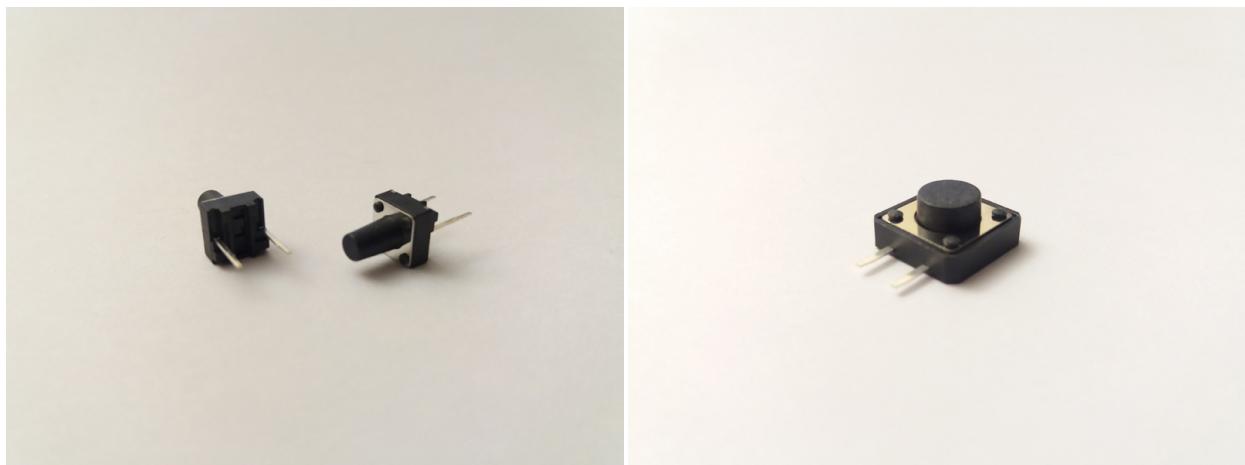
Kennzeichnung	Wert
braun - schwarz - schwarz - schwarz - braun	100 Ω
rot - schwarz - schwarz - schwarz - braun	200 Ω
orange - orange - schwarz - schwarz - braun	330 Ω
braun - schwarz - schwarz - braun - braun	1 k Ω
braun - schwarz - schwarz - rot - braun	10 k Ω

1.6 Drehpotentiometer



Bei dem Drehpotentiometer handelt es sich um einen regelbaren Widerstand, der in einem Spannungsteiler integriert ist. Aus diesem Grund hat das Potentiometer drei Anschlüsse: Im Bild befinden sich die Anode (+) und Kathode (-) an den äußeren Pins und am mittleren Pin liegt die eingestellte Spannung an. Das Spannungssignal lässt sich über einen analogen Pin am Mikrocontroller auslesen und weiterverarbeiten. Der maximale Widerstand des Potentiometers beträgt $10\text{ k}\Omega$. Drehpotentiometer eignen sich z.B. zur Steuerung der Helligkeit von LCDs und LEDs.

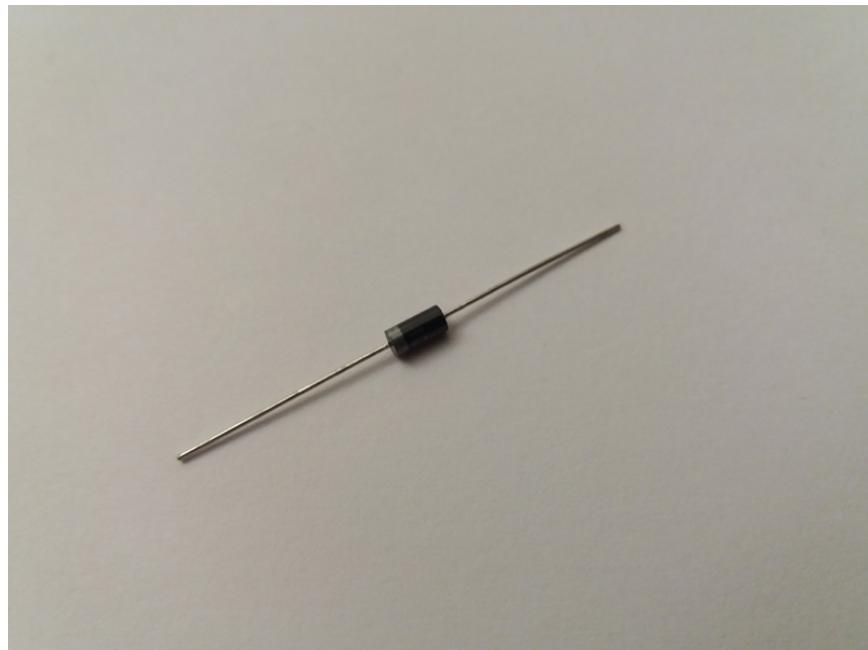
1.7 Taster



Die Taster schließen einen Stromkreis für die Dauer, in der sie gedrückt werden. Dies kann dazu verwendet werden, um Sensoren nach Bedarf zu aktivieren, oder um ein Signal zu senden, damit der Arduino eine Funktion beginnt.

Um eindeutige Werte zu erhalten, sollten Pull-Down- oder Pull-Up-Widerstände verwendet werden (siehe Übung).

1.8 Diode (1N4001)



Dioden lassen elektrische Ströme nur in eine Richtung fließen und sperren in die andere Richtung. Mit einer ausreichend hohen Spannung fließt ein Strom auch in Sperrrichtung, die Diode wird aber zerstört, wenn zulässige Werte überschritten werden. Die maximal zulässigen Spannungen der Diode liegen weit oberhalb der Spannungen, die der Mikrocontroller liefern kann, sodass ein Stromfluss in Sperrrichtung nicht möglich ist.

Im Bild befindet sich links die Kathode (-) und rechts die Anode (+), in Durchgangsrichtung fließt der Strom also von Anode zu Kathode.

1.9 LEDs

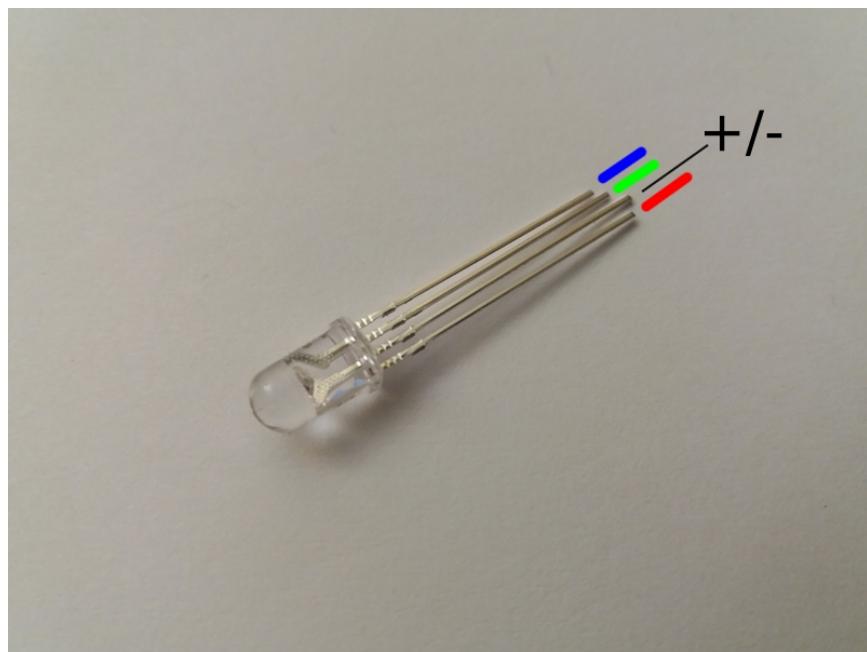
1.9.1 einfarbige LEDs



LEDs müssen mit Widerständen in Reihe geschaltet werden, da sie sonst zerstört werden! LEDs sind Dioden und haben somit eine Sperrrichtung, in der sie keinen geschlossenen Schaltkreis erlauben. Dies ist bei den vorliegenden LEDs durch unterschiedlich lange Beine gekennzeichnet. Das längere Bein sollte mit dem positiven Potential verbunden werden, um Stromfluss zu ermöglichen.

Die LEDs lassen sich über die digitalen Pins ein- und ausschalten. Außerdem können sie über die PWM-Pins (Pins mit Unterstützung für Pulsweitenmodulation) zusätzlich verschieden hell gedimmt werden.

1.9.2 RGB-LED



Zusätzlich zu den einfarbigen LEDs liegt dem Set eine RGB-LED bei. „RGB“ steht für die 3 Grundfarben der additiven Farbmischung: Rot, Grün und Blau. Durch die Mischung dieser Farbkanäle lassen sich viele weitere Farben erzeugen.

Dadurch können beispielsweise Signale, die mehrere Farben darstellen sollen, durch eine einzelne LED umsetzen. Dazu werden die drei Farbkanäle mit einem PWM-Signal angesteuert (Funktion `analogWrite()`).

Dem Set können zwei verschiedene Typen der RGB-LED beiliegen. Sie unterscheiden sich in der Funktion des längsten Beins - dieses kann entweder gemeinsame Anode oder Kathode sein. Die drei kürzeren Beine der LED steuern die einzelnen Farbkanäle an (siehe Abbildung). Wie auch normale einfarbige LEDs **müssen** diese mit Widerständen in Reihe geschaltet werden.

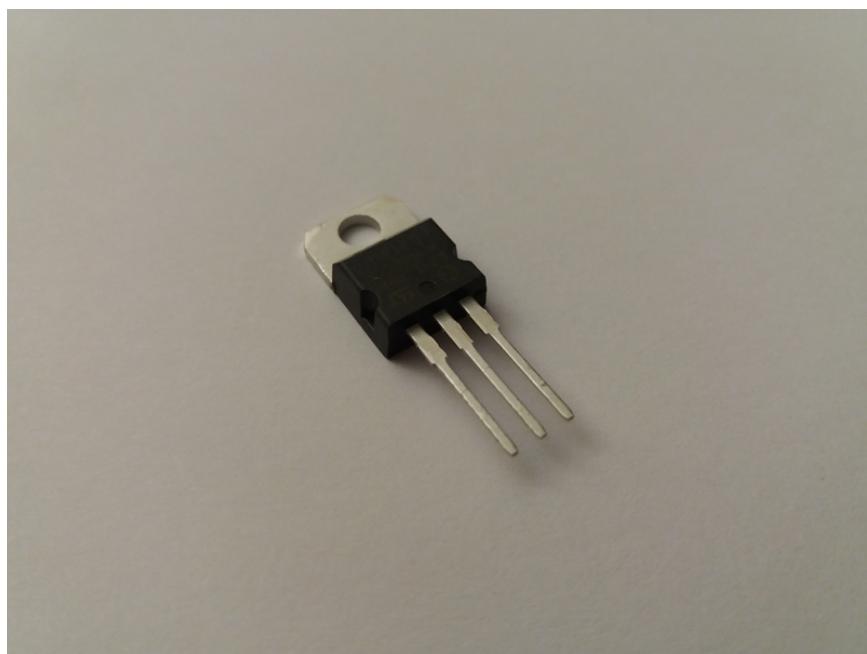
Je nachdem, welche Version der LED dem Set beiliegt, werden die Farben entweder über positive oder negative Spannung angesteuert.

1.9.3 IR-LED (IR-Sender)



Bei der IR-LED handelt es sich um einen Sender, der über Infrarotlicht Signale an den IR-Empfänger senden kann. Wie auch bei anderen LEDs befindet sich die Anode (+) am längeren Anschluss. Durch eine Kamera kann das Infrarotlicht sichtbar gemacht und die LED so getestet werden. Weitere Informationen befinden sich im Kapitel 2.7 IR-Empfänger (+ Fernbedienung).

1.10 Transistor (TIP120)



Ein Transistor ist allgemein ein Bauteil zur Verstärkung von elektrischen Signalen. Ein schwä-

ches Eingangssignal steuert dabei einen Stromkreis höherer Energie, sodass der Stromkreis die Informationen des Signals aufnimmt und diese so verstärkt.

Bei dem vorliegenden Transistor handelt sich um einen Bipolar-Transistor (NPN), welcher drei Anschlüsse hat: Collector, Base und Emitter. Die Bezeichnung NPN bedeutet, dass ein Strom zwischen Collector und Emitter nur dann fließt, wenn auch ein Strom zwischen Base und Emitter fließt. Ohne ein an Base anliegendes Signal fließt also kein Strom durch den Transistor. Die PNP-Bauweise erlaubt einen Stromfluss ohne anliegendes Eingangssignal. Durch diese Transistoren fließt weniger Strom, wenn das Eingangssignal stärker wird.

Im Bild befindet sich links der Base-, mittig der Collector- und rechts der Emitter-Anschluss des NPN-Transistors. Collector und Emitter können auch vertauscht werden, dies hat keine Auswirkungen auf die Funktion. Die Spannung zwischen Emitter und Base sollte 5 V nicht übersteigen.

Eine Alternative zu Transistoren sind Relais, die wesentlich höhere Spannungen (bis zu 220V) erlauben. Wegen des mechanischen Aufbaus ist jedoch keine Regulierung des Stromflusses möglich und nur eine vergleichsweise langsame Reaktion beim Öffnen oder Schließen des Schaltkreises gegeben. Weitere Informationen befinden sich im Kapitel 1.11 Relaiskarte.

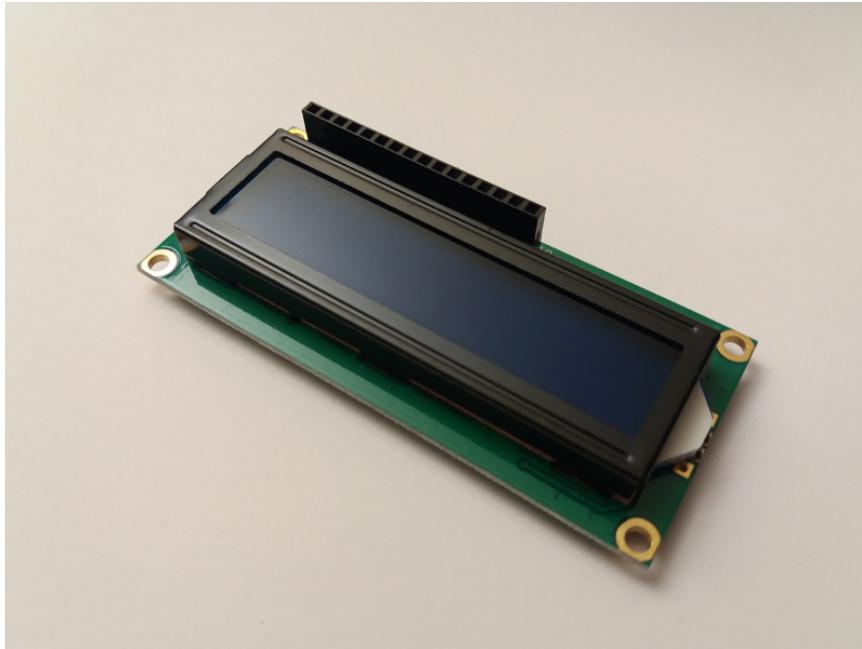
1.11 Relaiskarte



Die Relaiskarte kann genutzt werden, um einen Schaltkreis durch Signale des Arduinos zu öffnen und zu schließen. Dies ist von Vorteil, wenn Bauteile im Schaltkreis größere Ströme benötigen, als der Arduino zur Verfügung stellen kann. Die Funktionsweise der Relaiskarte ist ähnlich zu der eines Transistors, das Relais arbeitet jedoch deutlich langsamer.

Das Relais wird über den mittleren und den linken Pin mit der Spannung von 5 V versorgt. Wird der rechte Pin ebenfalls mit 5 V angesprochen, so schaltet das Relais zwischen den Kontakten A und C.

1.12 LC-Display (LCD Modul 1602)



Im LC-Display können in zwei Zeilen je 16 ASCII-Zeichen angezeigt werden. Dies kann beispielsweise ein statischer Text sein oder die Ausgabe eines Messwertes.

Wenn die entsprechende Bibliothek importiert wurde, lassen sich die Zeilen durch einen einfachen Befehl als string an das Display senden.

Tabelle 4: Pinbelegung des LCDs

Pin am LCD	Pin am Funduino
VSS	GND
VDD	5 V
V0	2
RS	33
RW	GND
E	31
D4	22
D5	24
D6	26
D7	28
A	5 V über 220 Ω
K	GND

2 Sensoren

2.1 Bewegungsmelder (HC-SR501)

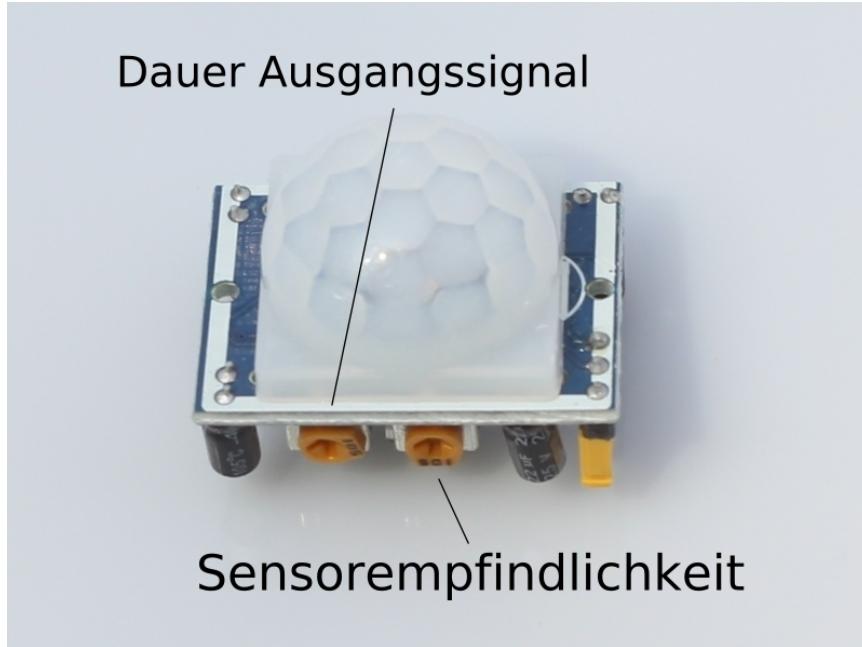


Der Bewegungssensor oder -melder (PIR, *passive infrared*) registriert Änderungen der auf ihn einfallenden Infrarotstrahlung. Diese können beispielsweise durch sich durch das Sichtfeld des Sensors bewegende Menschen ausgelöst werden. Es lassen sich mit dem Bewegungsmelder also Schaltungen realisieren, die ausgelöst werden, sobald sich eine Person in die Nähe des Sensors begibt.

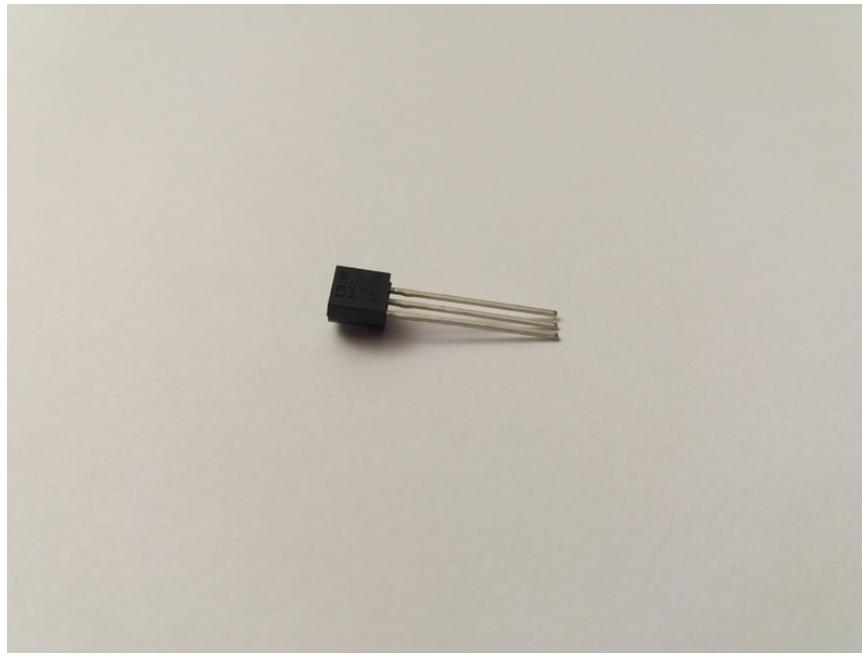
Die Kunststofflinse über dem Sensor lässt sich abnehmen, um die Bezeichnung der Pins lesen zu können.

Der Sensor wird über die äußeren zwei Pins VCC (Eingang) und GND (Ausgang, Masse) mit einer Spannung von 5 V versorgt. Der mittlere Pin OUT dient als Ausgang und wird bei einer detektierten Bewegung ebenfalls mit 5 V belegt. Die Messung dieser Spannung sollte am Arduino also mit einem digitalen Pin erfolgen.

Mit den gelben Stellschrauben an der unteren Seite des Sensors kann eingestellt werden, wie lange das Ausgangssignal bei einer Detektierung einer Bewegung aufrechterhalten wird (linke Schraube) und wie sensibel der Sensor reagieren soll (rechte Schraube).



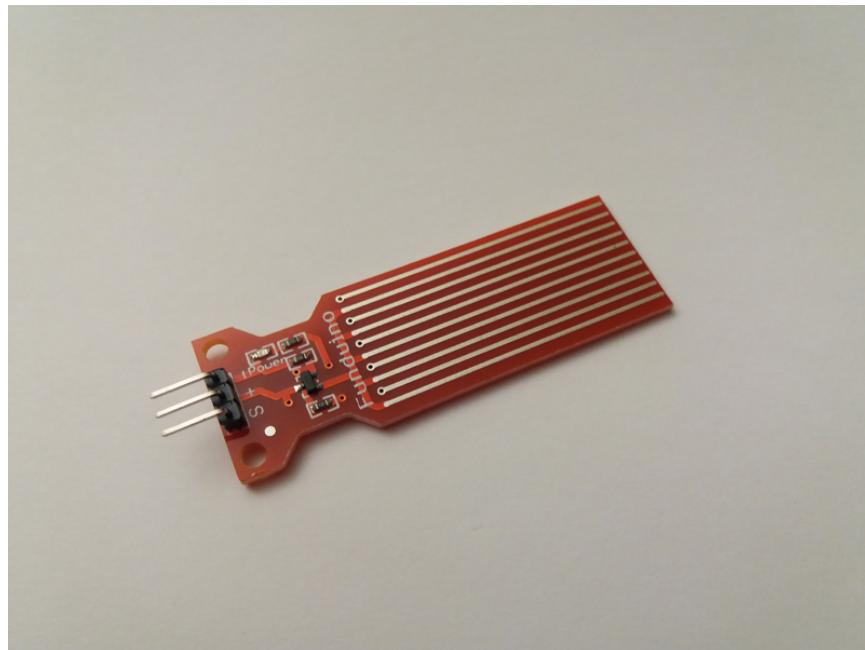
2.2 Temperatursensor (TMP36GT9Z)



Mit dem Temperatursensor lassen sich Temperaturen zwischen -40°C und 125°C messen. Daher kann er beispielsweise zur Bestimmung der Lufttemperatur eingesetzt werden.

Der Temperatursensor TMP36 misst die Temperatur und gibt über den mittleren Pin eine Spannung zwischen 0 V und 5 V zurück, die proportional zur gemessenen Temperatur ist. Daher wird für das Auslesen ein analoger Pin benötigt. Der Funduino erkennt den Spannungsbereich als Wert zwischen 0 und 410, weswegen hier die Verwendung des `map`-Befehls empfohlen wird:

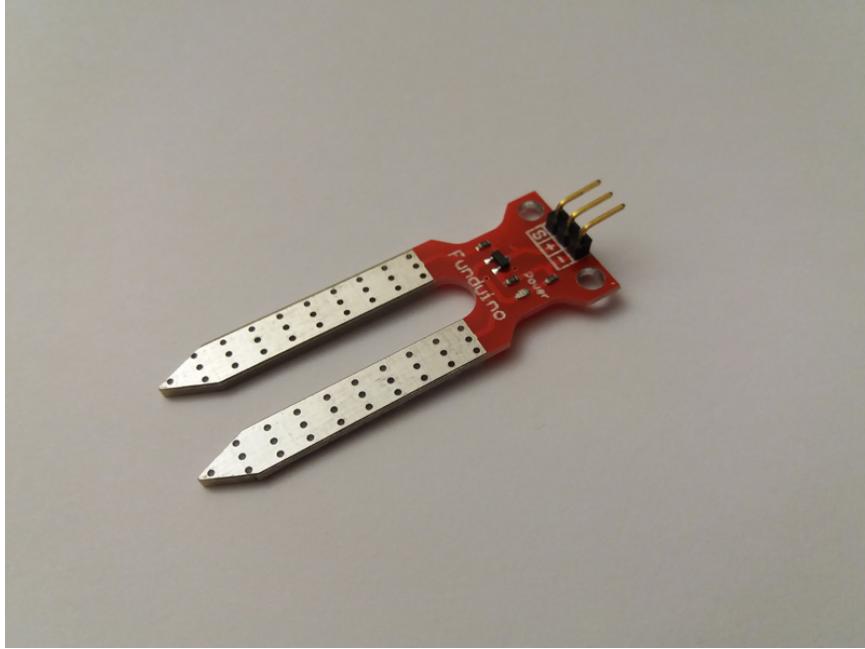
2.3 Tropfensor



Der Tropfensor registriert (elektrisch leitende) Flüssigkeiten, die sich auf ihm befinden. Dies kann beispielsweise genutzt werden, um Regen zu detektieren. Die Signalspannung liegt in analoger Form zwischen den Werten 0 und 1023 vor. Je mehr Tropfen sich auf der Oberfläche des Sensor befinden, die die Leiterbahnen miteinander verbinden, desto größer wird der ausgegebene Wert. Der Höchstwert wird mit Leitungswasser normalerweise nicht erreicht, da die Stärke der Signalspannung neben der Menge der Tropfen auch von der Leitfähigkeit des Mediums abhängig ist.

Weiterhin ist zu beachten, dass es wegen der unterschiedlichen anliegenden Potentiale zur Elektrolyse zwischen den beiden Elektroden des Sensors kommen kann. Daher ist von einer kontinuierlichen Messung abzuraten. Es empfiehlt sich beispielsweise ein Messabstand von 15 min.

2.4 Feuchtigkeitssensor



Der Feuchtigkeitssensor misst die direkt an ihm anliegende Feuchtigkeit, also nicht die Luftfeuchte! Es kann jedoch beispielsweise die Bodenfeuchtigkeit in einem Pflanztopf oder der Füllstand eines Wasserbehälters gemessen werden.

Der Sensor gibt eine Spannung aus, die proportional zur anliegenden Feuchtigkeit ist. Sie sollte daher analog ausgelesen werden. Wie hoch das ausgegebene Signal ist, ist jedoch auch vom einzelnen Sensor sowie der Leitfähigkeit des Mediums abhängig! Daher ist eine Kalibrierung vor dem Einsatz des Sensors empfehlenswert.

Wie auch beim Tropfensor wird wegen der Gefahr einer Elektrolyse empfohlen, nur einzelne Messwertaufnahmen im Abstand von mindestens 15 min durchzuführen.

2.5 Ultraschallsensor (HC-SR04)



Der Ultraschallsensor (Entfernungsmesser) besteht aus einem Lautsprecher und einem Mikrofon. Der Lautsprecher sendet beim Auslösen des Sensors einen Ultraschallimpuls, der an der nächsten Fläche reflektiert und dann vom Mikrofon gemessen wird. Aus der Laufzeit des Schallimpulses lässt sich unter Kenntnis der Schallgeschwindigkeit im vorliegenden Medium die Entfernung des reflektierenden Objektes zum Sensor bestimmen. Somit kann er genutzt werden, um beispielsweise den Zusammenstoß eines Roboters mit einem Hindernis zu vermeiden.

Die beiden äußeren Pins Vcc und Gnd des Sensors versorgen ihn mit der von ihm benötigten Spannung von 5 V, über die inneren Pins Trig und Echo wird der Sensors ausgelöst und sendet ein Signal zurück. Das zurückgesendete Signal besteht aus einem digitalen Signal (HIGH), das in seiner Dauer genau der Laufzeit des Schallimpulses entspricht (vgl. Tabelle 5 für wichtige Codes).

Tabelle 5: Codes zur Steuerung des Ultraschallsensors

Code	Funktion
<code>digitalWrite(Trig, HIGH);</code>	Aussendung einer Schallwelle, aktiviert durch Pin Trig.
<code>digitalWrite(Trig, LOW);</code>	Stoppt das Aussenden von Schallwellen an Pin Trig.
<code>time = pulseIn(Echo, HIGH);</code>	Messung der Zeit bis ein Signal (Impuls durch die Schallwelle) an Pin Echo eingeht.
<code>distance = (time/2)*0.03432;</code>	Umrechnung der gemessenen Zeit in eine Strecke (Schallgeschwindigkeit: 0,03432 $\frac{\text{cm}}{\text{ms}}$).

Hinweis: Der Datentyp der Variablen `time` und `distance` muss vorher festgelegt werden. Hier reicht der Typ `int` (Integer) im Allgemeinen aus.

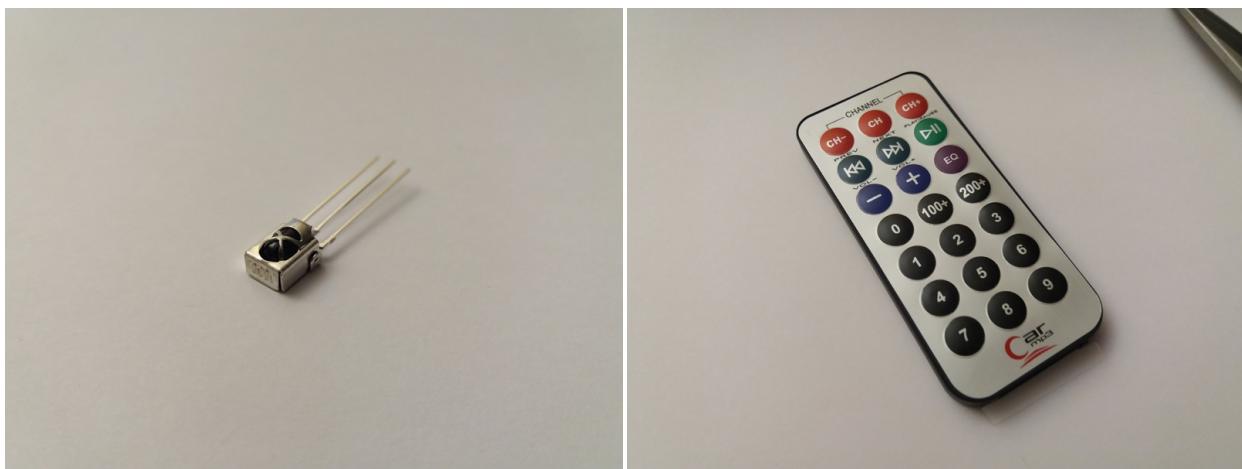
2.6 Fotowiderstand



Der Fotowiderstand (Helligkeitssensor) ändert seinen Widerstand in Abhängigkeit von der Menge des auf ihn einfallenden Lichts. Dieser Widerstand kann vom Arduino gemessen und verarbeitet werden.

Um den vom Helligkeitssensor erzeugten Widerstand zu bestimmen, wird eine Reihenschaltung mit einem weiteren Widerstand benötigt.

2.7 IR-Empfänger (+ Fernbedienung)



Bei dem IR-Empfänger handelt es sich um einen Sensor, der Infrarotlicht detektieren und Spannungssignale ausgeben kann. Der Empfänger hat drei Anschlüsse: Zeigt die dunkle Linse zum Betrachter, befindet sich am linken Pin das Spannungssignal, welches in digitaler Form vorliegt, am mittleren Pin GND und am rechten Pin die Versorgungsspannung von 5 V.

Die Infrarotlicht-Signale können entweder von der Fernbedienung oder von der IR-LED (Kapitel

1.9.3 IR-LED (IR-Sender)) gesendet werden. Da die Signale kodiert übermittelt werden müssen, d.h. der Sender verschickt Informationen über die Blinkgeschwindigkeit und Leuchtdauer, empfiehlt sich für den Einstieg in die Thematik die Nutzung der Fernbedienung, da diese ihre Signale bereits kodiert aussendet. Für die IR-LED muss die Funktion manuell nachprogrammiert werden.

Damit der IR-Empfänger die Signale entschlüsseln kann, wird eine externe Bibliothek (IRremote) benötigt. Diese lässt sich in der Bibliothek-Verwaltung der Arduino IDE über einen entsprechenden Suchbegriff finden und installieren.

Jeder Taste auf der Fernbedienung ist eine Zahlenkombination zugeordnet (z.B. „16724175“ für Taste 1). Über den Serial-Monitor der Arduino IDE können die Zahlenkombinationen aller Tasten ausgelesen werden und durch einen Vergleich der Zahlen in Code-Sketches bestimmte Tasten der Fernbedienung mit einer Funktion belegt werden. In Tabelle 6 sind die wichtigsten Codes zur Steuerung des IR-Empfängers gegeben.

Tabelle 6: Codes zur Steuerung des IR-Empfängers

Code	Funktion
#include <IRremote.h>;	Bindet die Bibliothek „IRremote.h“ für IR-Signale ein. Diese muss zuvor über die Arduino-IDE hinzugefügt werden.
IRrecv irrecv(11); decode_results results;	Legt Pin 11 für die Eingangssignale fest. Speicherung der Ergebnisse in der Variablen „results“.
irrecv.enableIRIn(); if (irrecv.decode(&results)) {}	Startet die Aufnahme möglicher Signale. Wenn Ergebnisse vorliegen, wird der Inhalt der Bedingung ausgeführt.
if(results.value == 16724175) {}	Entspricht das Ergebnis der Zahl „16724175“ (Code-Zahl für Taste 1 auf der Fernbedienung), wird der Inhalt der Bedingung ausgeführt.
irrecv.resume();	Setzt das Programm fort.

Ein kompletter Sketch mit Erklärungen³ ist auf der Funduino-Webseite zu finden.

³<http://funduino.de/nr-11-fernbedienung>

2.8 RFID-Kit (RFID-RC522, Chip MFRC522)



Das RFID-Kit besteht aus einer Karte, einem Schlüsselanhänger und einem Empfänger. In der Karte und dem Schlüsselanhänger befindet sich jeweils ein RFID-Chip, dessen Informationen mit Hilfe des Empfängers ausgelesen werden können.

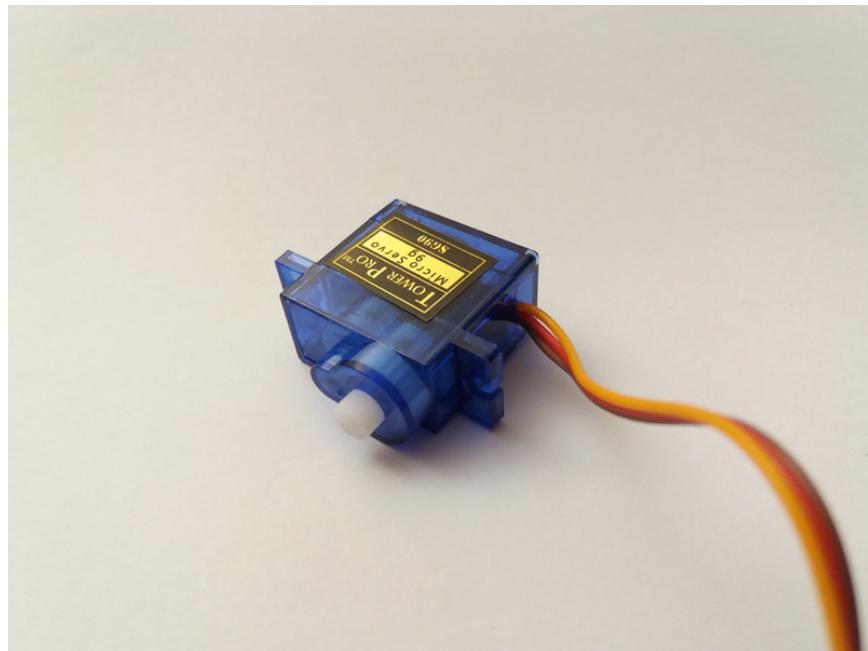
Ein Datenblatt für das RFID-Kit⁴ ist in der Produktbeschreibung im Funduino-Shop⁵ zu finden.

⁴http://www.nxp.com/documents/data_sheet/MFRC522.pdf

⁵http://www.funduinoshop.com/epages/78096195.sf/de_DE/?ObjectID=45510257

3 Aktoren

3.1 Servo



Der Servo kann Positionen zwischen 0 und 180 ° ansteuern. Damit lässt sich beispielsweise die Lenkung eines Autos umsetzen.

Der Servo wird über zwei Pins (schwarzes und rotes Kabel; schwarz: GND) mit seiner Arbeitsspannung von 5 V versorgt. Der dritte Pin liefert dem Servo pulsweitenmodulierte Signale, deren Länge proportional zum gewünschten Winkel ist (siehe Tabelle 7). Zwischen den einzelnen gesendeten Pulsen sollten mindestens 20 ms liegen. Für die einfachere Handhabung existiert weiterhin die Bibliothek Servo.h.

Tabelle 7: PWM-Längen für verschiedene Einstellwinkel des Servomotors

Pulslänge	Winkel
1 ms	0 °
1,5 ms	90 °
2 ms	180 °

3.2 Schrittmotor (28BYJ-48) mit Treiberplatine (ULN2003)



Die Drehposition des Schrittmotors kann sehr genau gesteuert werden. Eine volle Umdrehung entspricht 2048 Schritten.

Der Schrittmotor wird über die Pins GND und VCC der Steuerplatine mit seiner Arbeitsspannung von 5 V versorgt.

4 Weitere Informationen und Links

Arduino Homepage	arduino.cc
Funduino Homepage	funduino.de
Anleitungen zu Funduino	funduino.de/anleitungen
Arduino/Funduino	https://www.arduino.cc/en/Main/ArduinoBoardMega2560
MEGA2560-Mikrocontroller	
Arduino/Funduino Code-Referenz	https://www.arduino.cc/en/Reference/HomePage
Datenblätter zu Bauteilen	http://funduino.de/downloads