Fachschule für Mechatroniktechnik Herzogenaurach

R2D2

Nachbau des Originals

Bei meinen MCT-Projekt „R2D2“ habe ich mich optisch, wie auch aus technischer Sicht, an den originalen Roboter aus den „Star Wars“ Filmen orientiert. Bei der Konstruktion handelt es sich, um lackierten 3D-Druck. Die Elektronik, die den Roboter Leben ein verhaucht, wird zentral über einen programmierten Mikrocontroller gesteuert. Dieser lässt sich via Bluetooth mit einen Smartphone verbinden und durch einer kompatiblen App ist es möglich den „R2D2“ steuern.

Knauer Johannes

24.4.2018

Inhaltsverzeichnis

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

1. Beschreibung der Idee

Bei der Ideenfindung war mir direkt klar, dass mein Projekt etwas mit Star Wars zu tun haben muss, da ich ein Fan dieses „Universums“ bin und mit Hilfe eines Mikrocontrollers nachbauten von Raumschiffen oder Robotern möglich sind. Zuerst war ein Landgleiter der Räder und Motoren besitzt angedacht. Unteranderem sollte eine Objektverfolgung mit Hilfe eines speziellen Sensors und einiger Programmierlogik realisiert werden. Des Weiteren steckte ich mir das Ziel, den Gleiter mit ein Smartphone zu steuern.

Abbildung 1: „Luke´s Landgleiter“

Es hat sich aber herausgestellt, dass der Landgleiter zu wenig Platz für alle Bauteile hatte, deshalb musste eine neue Idee her. Auf der Internetseite www.thingiverse.com bin ich auf einen rein mechanisch gesteuerten „R2D2“ gestoßen. (ChaosCorTech, 2018)

Dieser „R2D2“ wurde mit Hilfe eines 3D-Druckers hergestellt, die nötigen Informationen bzw. Dateien stehen für jedermann zum Download bereit. Da mein Schulkollege, Blaufelder Daniel, über so eine Vorrichtung verfügt, half er mir bei Umsetzung der Konstruktion und druckte mir alle benötigten Bauteile.

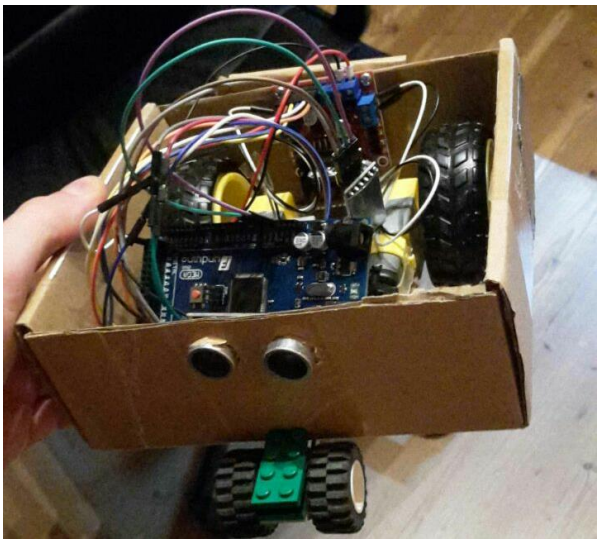


Abbildung 2: “Prototyp 1 von 2“

Bis zur Fertigstellung dieser, beschäftigte ich mich mit den elektrischen Bauteilen und ihrer Programmierung. Durch den Bau zweier Prototypen testete ich alle Bauteile, dadurch fielen mir weitere Funktionen ein, die ich umsetzten könnte. Nach der Fertigstellung aller zu druckenden Teilen, der Lackierung, den Zusammenbau, der Verkabelung und Änderung vieler Kleinigkeiten verging eine Menge Zeit.

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

Doch schlussendlich erreichte ich alle Ziele/Funktionen die ich mir gesteckt habe.

Die Ziele waren:

- Via Bluetooth und Smartphone den „R2D2“ steuern
- Antrieb mit zwei DC-Motoren über die „H-Bridge/L298N“ regeln
- Objektverfolgung durch den „IR-Compound-Sensor“
- RGB-Led integrieren
- Kopfbewegungen mit den Servomotor „Tower Pro/MG995“ realisieren
- Die typischen Geräusche des „R2D2s“ über einen Lautsprecher ausgeben
- Mit den SD-Card-Reader die Geräusch-Dateien zur Verfügung stellen
- Optisch den Original so nahe wie möglich kommen

Als besonders „Zeitfressend“ stellte sich das lackieren, das Zusammenbringen aller elektrischen Bauteile und ausbessern von Kleinigkeiten heraus.

2. Funktionsbeschreibung

Der Bediener sendet Befehle über sein Smartphone an den Bluetooth-Empfänger bzw. den Arduino-Board. Dort werden diese Signale verarbeitet und die zugeordneten Aktionen ausgeführt.

Um die Funktionsweise bzw. die Logik hinter den komplizierteren Aktionen leichter zu verstehen, werde einige Bilder folgen. Aber zuerst ein paar Worte zu den „einfacheren“ Funktionen.

Die Funktion „Geräuschausgabe“ wird aus der Kombination des SD-Karten-Lesers und den Lautsprecher realisiert.

Der Mikrocontroller bekommt die nötigen Informationen zur Geräuscherzeugung von der SD-Karte und sendet die dementsprechenden Signale Richtung Lautsprecher.

Die „leuchtenden RGB-Led“ ist sehr simple, denn je nachdem welcher Pin „High“ ist, ändert die Led ihre Farbe (Quellcode Zeile: 287-318).

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

Das Antriebssystem des „R2D2“ wird durch zwei DC-Motoren und einen Servomotor realisiert.

Antriebssystem:

→ siehe (QZ = 1-7)

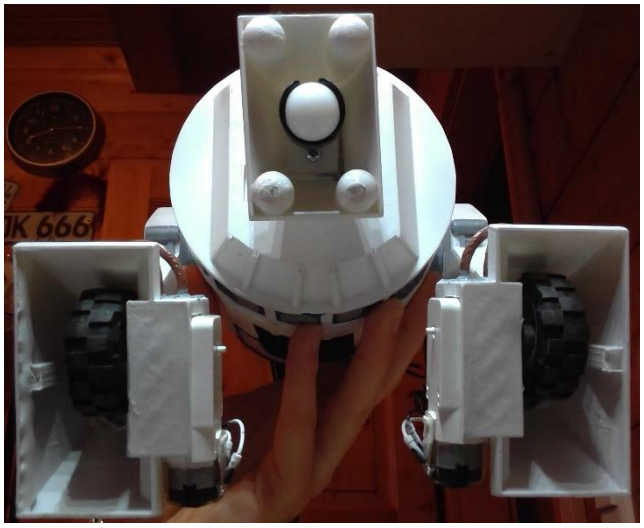


Abbildung 3: „DC-Motoren mit Räder und kugelförmiger Fuß“

→ siehe (QZ = 38-43)

+ (QZ = 493-511)

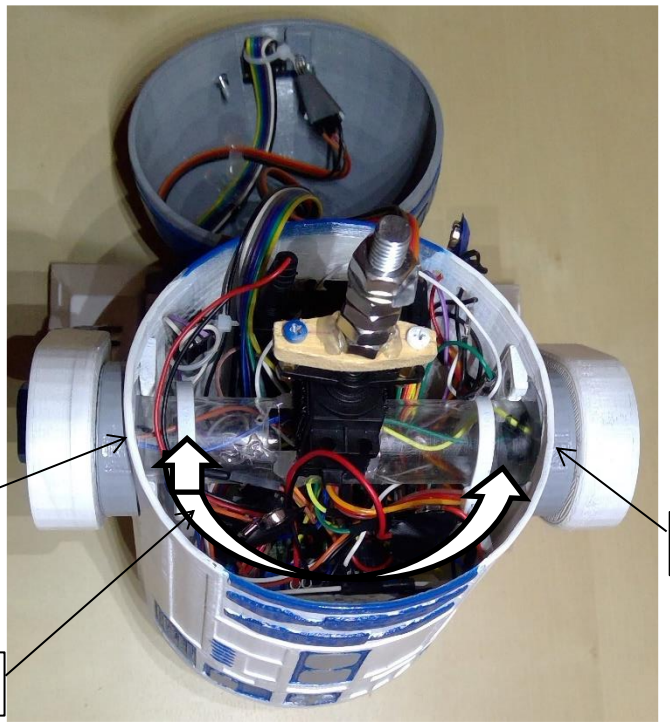


Abbildung 4: „Servo in Grundstellung = 90°“

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

Durch die Kombination des „Antriebssystems“ und des „IR-Compound-Eye“ kann eine Objektverfolgung realisiert werden. Das „IR-Compound-Eye“ sendet mit seinen Infrarot-Leds ein kurzes Signal. Danach misst es die Konzentration der Strahlen, die vom Objekt zurück reflektiert werden. Die unterschiedlichen positionierten Infrarotsensoren detektieren diese. Bei keinem Infrarotanteil bekommt der Mikrocontroller die Signalstärke „0“ geliefert, bei maximaler „1023“. (Analoger Eingang)

→ 1023 Objekt direkt vor den Sensor

→ 500 Objekt ca. 8-10cm entfernt

→ siehe (QZ = 23-31, 374-415)

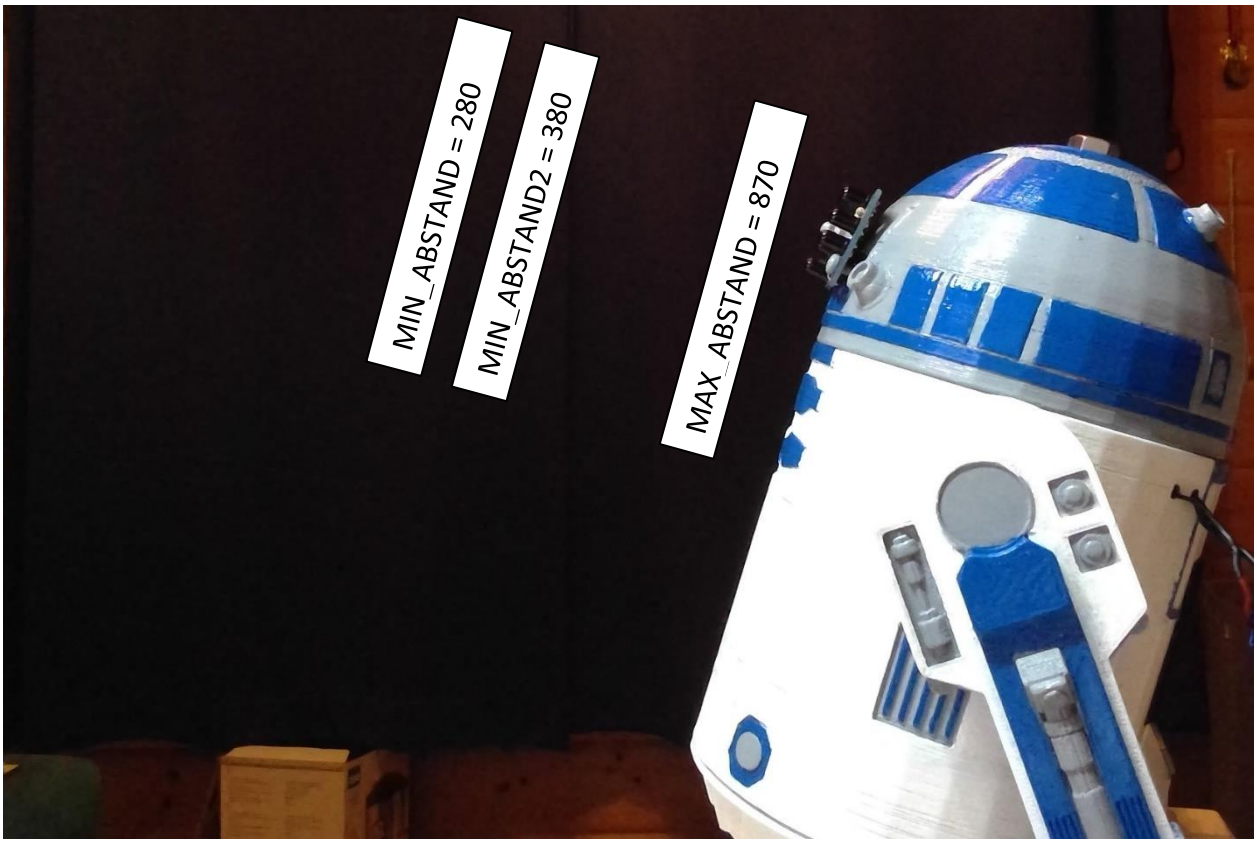


Abbildung 5: „Abstände Objektverfolgung“

Um die Methode „followObject()“ (QZ = 374-415) leichter zu verstehen ist es hilfreich, sich die Bilder und Variablen anzuschauen und mit den if-Abfragen bzw. den Bedingungen aus den Quellcode bildlich zu verknüpfen.

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

3. Beschreibung der Schaltung

Die Schaltung wurde im Programm „Fritzing“ angefertigt. Ich entschied mich dafür mehrere Fritzing-Dateien anzufertigen. Zum einen, weil es bei so vielen Bauteilen/Leitungen sehr unübersichtlich wird. Zum anderen, weil das Programm immer wieder abstürzte, wenn ich alle Bauteile in eine Schaltung zusammenbringen wollte. Deshalb fertigte ich vier Fritzing-Schaltungen an:

→ SD-Card-Reader

→ HC-05 + RGB LED

→ IR-Compound-Eye + H-Bridge

→ Lautsprecher + Servo

Der Verwendete „Arduion Mega 2560“, wie auch die „H-Bridge“ werden von einer 9V Blockbatterie mit Energie versorgt. Einige „GND“ und „5V“ Anschlüsse sind in der Realität verlötet, das hat aber keinen Einfluss auf irgendwelche Funktion. Die vier kleinen Fritzing-Schaltungen ergeben zusammen eine große. Der Quellcode ist sehr gut beschrieben, deshalb werde ich des Öfteren darauf verweisen. Im Anschluss werde ich einige Besonderheiten ansprechen:

→ SD-Card-Reader:

Wenn man einen SD-Karten-Lesegerät verwendet, muss man je nach Mikrocontroller-Model darauf achten, welche Geräteanschlüsse man auf welchen Mikrocontroller Pin setzt.

Arduino Mega 2560: (admin, 2018) (Quellcode Zeile: 49, 56-59)

→ Pin 53 = CS → Pin 52 = SCK → Pin 51 = MOSI → Pin 50 = MISO

→ HC-05 + RGB LED:

Es gibt zwei verschiedene Ausführungen der RGB-Led, die eine hat eine Kathode, die Andere eine Anode. Bei diesem Projekt handelt es sich, um die erst genannte. (Funduino, 2018) (QZ: 61-66) Der Bluetooths Sender und Empfänger „HC-05“ hat zwei besondere Kommunikationsanschlüsse. Den „TX“ und den „RX“ Anschluss. (Funduino, 2017) (QZ: 11-12)

→ Pin RX0 = TX → Pin TX0 = RX

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

→ IR-Compound-Eye + H-Bridge:

Die „H-Bridge“ kann zwei DC-Motoren gleichzeitig ansteuern. (Tech, Youtube, 2017) (QZ: 1-7)
Das „IR-Compound-Eye“ ist sehr Lichtempfindlich, da es selbst mit Infrarotstrahlen arbeitet. (Robotrus, 2018) (QZ: 16-21)

→ Lautsprecher + Servo:

Der Servo „Tower/MG995“ ist ein Servo mit einer hohen Momenten Leistung, deshalb benötigt er eine externe Energiequelle. (Funduino, 2018) (QZ: 38, 93) Der Lautsprecher zeigt Gemeinsamkeiten mit der SD-Karten-Erkennung, da dieser auch einen exakt vorbestimmten Ausgabe Pin benötigt und dieser wiederrum Modelabhängig ist. (GitHub, 2018)(QZ: 50, 52-54)
→ Mögliche Ausgabepins = 5|6|11|46

4. Beschreibung des Quellcodes

Der beiliegende Quellcode ist meiner Meinung nach sehr gut beschrieben. Die Kommentare verraten beinahe alles über das Programmierte. Trotzdem werde ich noch genauer auf die Bibliotheken, ihre Besonderheiten/Methoden und den ein oder anderen „Zeitfresser“ ansprechen. Die Pin-Belegung wurde teilweise von dem Mikrocontroller-Model oder von verwendeten Bibliotheken vorgegeben. (siehe < **TMRpcm.h** >)

Bibliotken und ihre Methoden:

→ <**Servo.h**> (Margolis, 2018) (QZ: 35)
Es kann ein „Servo-Objekt“ erzeugt werden mit dem man Methoden ausführen kann. (QZ: 36)
Zum Beispiel:
(QZ: 93)→ .attach(); Hier wird den Servo eine Steuerleitung zum Arduino-Board zugeteilt.
(QZ: 94)→ .write(); Hier wird den Servo ein Zielposition zugeteilt, die er anfährt. Dabei ist darauf zu achten, dass der Servo eine Verfahr Zeit hat.
(QZ: 210)→ .read(); Hier wird die aktuelle Position des Servos ausgelesen.
Eine Besonderheit von der „write()“-Methode ist es, dass eine externe Variable mit den Inhalt der runden Klammern beschrieben wird. Wenn die „read()“-Methode den aktuellen Wert auslesen will, „checkt“ sie nicht den aktuellen Winkel des Servos, sondern nur die externe Variable. Es kann zu Ablauffehlern kommen! (Mögliche Lösung: Verfahr Zeiten)

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

→ <**SD.h**> (SparkFun, 2018) (QZ: 52)

Es kann eine SD-Karte ausgelesen und Methoden ausgeführt werden.

Zum Beispiel:

(QZ: 99-105)→ SD.begin(CS); Hier wird der CS-Pin angesteuert, dadurch wird überprüft, ob eine SD-Karte vorhanden ist. *Siehe „Beschreibung der Schaltung“ →SD-Card-Reader

(QZ: 47-49)→ In diesen Quellcode Zeilen gehe ich näher auf einige Besonderheiten zur SD-Karten Formatierung und Liederkonvertierung ein. (Tech, YouTube, 2018) (audio.online-convert, 2018)

→ <**TMRpcm.h**> (GitHub, 2018) (QZ: 53)

Es kann ein „Lautsprecher-Objekt“ erzeugt werden mit dem man Methoden ausführen kann. (QZ: 54)

Zum Beispiel:

(QZ: 97)→ .speakerPin; Hier wird das Attribut „speakerPin“ des „Lautsprecher-Objektes“ eine Steuerleitung zum Arduino-Board zugeteilt.

*siehe „Beschreibung der Schaltung“ → Lautsprecher + Servo

(QZ: 98)→ .setVolume(5); Hier wird die Lautstärke des Lautsprechers festgelegt mit „5“. Der Minimalwert ist 0 und der Maximalwert ist 7.

(QZ: 197)→ .play(); Hier kann eine Datei von der SD-Karte abgespielt werden. Aber Achtung, je nach Stand der Bibliothek, die man herunterlädt, darf der Dateiname nicht länger als 8 Zeichen sein.

(QZ: 320)→ .isPlaying(); Hier kann überprüft werden, ob sich der Lautsprecher aktuell im Wiedergabemodus befindet, oder schon damit fertig ist.

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

Die Bibliothek hat eine Besonderheit, die man nicht übersehen sollte, denn wenn man einmal die Methode „play()“ ausführt, werden die PWM/Timer 1 ,2 und 3 ausgeschaltet, um Störgeräusche zu vermeiden.

On the Arduino Mega we have 6 timers and 15 PWM outputs:

Pins 4 and 13: controlled by timer0
Pins 11 and 12: controlled by timer1
Pins 9 and10: controlled by timer2
Pin 2, 3 and 5: controlled by timer 3
Pin 6, 7 and 8: controlled by timer 4
Pin 46, 45 and 44:: controlled by timer 5

Abbildung 6: „PWM-Timer-Pins“

Das hat zur Folge, dass diese Pins im Anschluss nutzlos sind. Dies Besonderheit hat sich bei meinem Projekt als echter „Zeitfresser“ erwiesen.

Hinweis: Ab der Quellcode Zeile „514“ befindet sich ein Methodenverzeichnis.

5. Bedienungsanleitung

Der „R2D2“ wird, wie schon erwähnt, über das Smartphone gesteuert. Die App die ich benutzte heißt „Arduino bluetooth“. Der Aufbau der App ist fast selbsterklärend. Nach der Installation und öffnen der App, wird man aufgefordert sein Bluetooth zu aktivieren. Jetzt stellt man die Verbindung zum Bluetooth-Empfänger her. (HC_05)

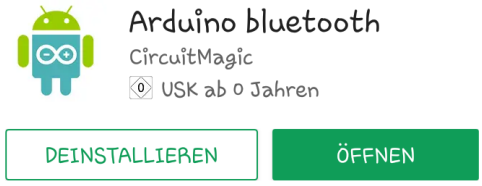


Abbildung 7: „Arduino bluetooth“

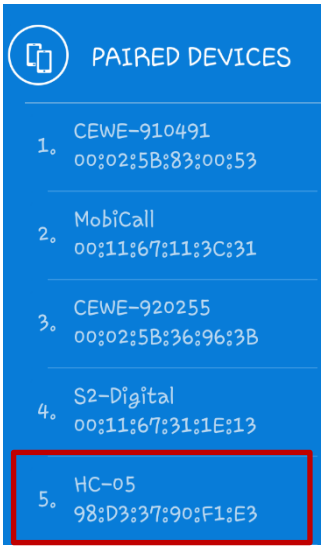


Abbildung 9: „Kopplungsmenü“

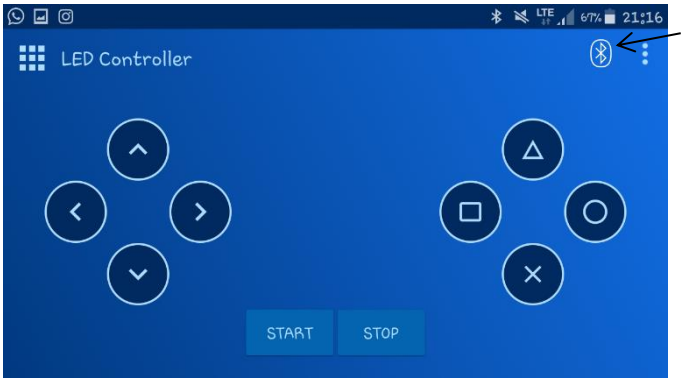
Abbildung 8: „Menü“

Man hat nun die Auswahl zwischen vier Menüpunkten. (QZ = 114-118)

Der Quellcode bezieht sich nur auf die Menüpunkte „Control pad“ und „Buttons“.

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

Menüpunkt „Control pad“: → siehe (QZ = 141-175)



Durch einen Klick auf das Bluetooth-Zeichen, kann die Verbindung getrennt werden.

Abbildung 10: „Control pad“

Bei einem Klick auf den Button mit den drei senkrecht zueinander laufenden Punkten, stellt man die Zeichen, je Button ein. Jede Taste ist mit einem Zeichen hinterlegt.

→ Beispiel: - Abbildung 11: „Button Einstellung“

Menüpunkt „Buttons“: → siehe (QZ = 176-188)

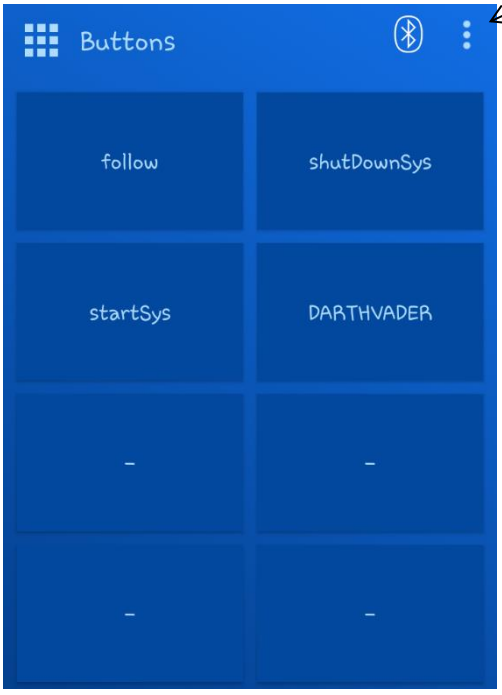


Abbildung 12: „Buttons“

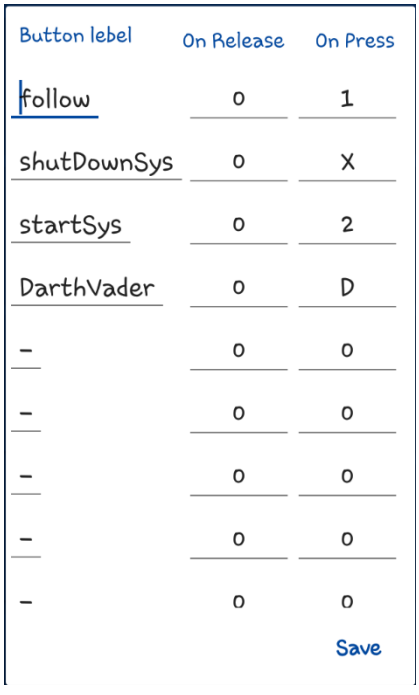


Abbildung 11: „Button Einstellung“

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

6. Quellenverzeichnis

6.1. Literaturverzeichnis

admin. (23. Februar 2018). *Bajdi.com*. Von http://www.bajdi.com/arduino-mega-2560-and-sd-card-modul/ abgerufen

audio.online-convert. (19. Januar 2018). Von https://audio.online-convert.com/convert-to-wav abgerufen

ChaosCorTech. (17. Januar 2018). *Thingiverse*. Von https://www.thingiverse.com/thing:1205821 abgerufen

Funduino. (15. Dezember 2017). Von https://funduino.de/tutorial-hc-05-und-hc-06-bluetooth abgerufen

Funduino. (15. März 2018). Von https://funduino.de/nr-20-rgb-led abgerufen

Funduino. (19. Januar 2018). Von https://funduino.de/nr-12-servo-ansteuern abgerufen

GitHub. (19. Januar 2018). Von https://github.com/TMRh20/TMRpcm/wiki abgerufen

Margolis, M. (20. Januar 2018). *ArduinoCC*. Von https://www.arduino.cc/en/Reference/Servo abgerufen

Robotrus. (15. Januar 2018). Von https://www.robot-r-us.com/sensor-infrared/ir-compound-eye.html abgerufen

SparkFun. (23. Februar 2018). *ArduinoCC*. Von https://www.arduino.cc/en/Reference/SD abgerufen

Tech, M. A. (15. Dezember 2017). *Youtube*. Von https://www.youtube.com/watch?v=Da4HY7HZ6h0 abgerufen

Tech, M. A. (20. Februar 2018). *YouTube*. Von https://www.youtube.com/watch?v=gi9mqIha8n0 abgerufen

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

6.2. Bilderverzeichnis

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

7. Anhang

7.1. Quellcode inkl. Zeilenzahlen

Projekt_R2D2

```
//Anschluesse H-Bright L298N/DC Motoren + Variablen:          Methodenverzeichnis: (Strg+L = 514)
const int ENA = 6;                  //Steueranschluss/ENA mit PWM (Rad in Fahrrichtung rechts) regelt die Umdrehungsgeschw.(min. 0 - max. 255)--> Spannungswert--> 0-5 Volt
const int IN1 = 22;                 //IN1+IN2 bestimmen den Drehsinn des Rades in Fahrrichtung rechts, einer von beiden muss auf "LOW", der andere auf "HIGH" eingestellt sein.
const int IN2 = 23;                 // " "
const int IN3 = 24;                 //IN3+IN4 bestimmen den Drehsinn des Rades in Fahrrichtung links, einer von beiden muss auf "LOW", der andere auf "HIGH" eingestellt sein.
const int IN4 = 25;                 // " "
const int ENB = 7;                  //Steueranschluss/ENB mit PWM (Rad in Fahrrichtung links) regelt die Umdrehungsgeschw.(min. 0 - max. 255)--> Spannungswert--> 0-5 Volt

int engineSpeed = 0;                //Eine Variable fuer die Umdrehungsgeschw.(min. 0 - max. 255) || siehe H-Bright--> ENA/ENB Anschluesse--> Beschreibung (Strg+L 1)

//Anschluesse Bluetooth-Empfaenger + Variablen:
//Der Bluetooth-Empfaenger hat neben GND und 5V noch 2 Kontaktstellen zum Microcontroller.--> (1.RX-PIN = TX0, 2.TX-PIN = RX0)

char bluetoothValue = ' ';          //Die Variable "bluetoothValue" ist sehr wichtig, denn sie speichert die Befehle vom Bediener.--> Variable zum vergleichen

//Anschluesse IR Compound Eye + Variablen:
const int TOP = A0;                 //Der Anschluess "TOP" entspricht den Infrarotsensoren oben.
const int LEFT = A1;                //Der Anschluess "LEFT" entspricht den Infrarotsensoren links.
const int BOTTOM = A2;              //Der Anschluess "BOTTOM" entspricht den Infrarotsensoren unten.
const int RIGHT = A3;               //Der Anschluess "RIGHT" entspricht den Infrarotsensoren rechts.
const int LEDS = 13;                //Der Anschluess "LEDS" entspricht den 4 Infrarot sendenten Leds, die in der Mitte des "IR Compound Eye" sitzen.

int oben = 0;                   //Die Variable "oben" hat das Ziel, den analogenWert am Pin, "TOP", abzuspeichern.
int links = 0;                  //Die Variable "links" hat das Ziel, den analogenWert am Pin, "LEFT", abzuspeichern.
int unten = 0;                  //Die Variable "unten" hat das Ziel, den analogenWert am Pin, "BOTTOM", abzuspeichern.
int rechts = 0;                 //Die Variable "rechts" hat das Ziel, den analogenWert am Pin, "RIGHT", abzuspeichern.

int durchschnittsAbstand = 0;       //Die Variable "durchschnittsAbstand" teilt den analogen-addierten Wert aller Sensoren durch 4.
const int MIN_ABSTAND = 280;        //Die Variable "MIN_ABSTAND" dient als min Grenzwert fuer die Verfolgung eines Objektes, bei ueberschreiten des Werte wird eine Objektverfolgung ausfuehren.--> DC Motoren und Kopfservo in Betrieb
const int MIN_ABSTAND2 = 380;       //Die Variable "MIN_ABSTAND2" dient als zweiter min Grenzwert fuer die Verfolung eines Objektes.
const int MAX_ABSTAND = 870;        //Die Variable "MAX_ABSTAND" dient als max Grenzwert fuer die Verfolgung eines Objektes, bei ueberschreiten des Werte entfernt sich der "R2D2" vom Objekt.
//const double multiplikator = 1.0;  //Die Variable "multiplikator" dient zur Vergroeßerung, des analogen Wertes, des rechten Infrarotsensors, da dieser von Werk aus schwaecher eingestellt ist.--> Kabelfehler beseitigt

//Anschluesse Servo + Variablen:
#include <Servo.h>                  //Die Bibliothek "Servo.h" wird aufrufen, um mit einen Servo arbeiten zu koennen.
Servo headServo;                    //Es wird ein Servo-Objekt("headServo") erzeugt. Im Anschluss wird der Anschlusspin/Steuerungspin festgelegt, desweitern werden Methoden, wie z.B. Stellwinkelverstellung ausgefuehrt.

const int PIN_SERVO = 8;            //Der Steurungspin "PIN_SERVO" eines Servos wird mit "8" festgelegt.
const int POS_MIN = 30;             //Der hier verbaute Servo hat nur einen Arbeitsbereich von 180°, deswegen werden zwei Grenzwerte festgelegt, die konstruktionstechnisch sinnvoll sind.--> 1."POS_MIN" = 30° und...
const int POS_MAX = 150;            //... 2."POS_MAX" = 150°.
int posCurrent = 90;                //Die Variable "posCurrent" speichert den aktuellen Stellwert des Servomotors. Zu Beginn wird "posCurrent" mit 90° initalisiert.--> Grundstellung
const int INITIAL_POS = 90;         //Die Variable "INITIAL_POS" ist die Grundstellung des Kopfservos.--> 90° entspricht der Mitte
const int STEP_RANGE = 3;           //Die Variable "STEP_RANGE" wird beim ausfuehren der Objektverfolgung auf die aktuelle Position("posCurrent") des Servos addiert und subtrahiert.

//Anschluesse MicroSD Card Adapter/Lautsprecher + Variablen:
//Sound Ausgabe bei Arduino Mega 2560:
//Wichtige Info/SD-Karte:--> SD-Karte formatieren--> (empfohlen ist "FAT" ->siehe arduino.cc/Notes on using SD cards)
//Wichtige Info/Sounds:--> Die Sounds muessen in 1.WAV 2.Change bit resolution = 8 Bit 3.Change sampling rate = 16000Hz 4.Change audio channels = mono umgewandelt werden.
//Wichtige Info/ModulePins:--> Bei den Arduino Mega 2560 muessen die Anschluesse so gewaehlt werden--> (1.CS Pin = 53, 2.SCK Pin = 52, 3.MOSI Pin = 51, 4.MISO Pin = 50, 5.VCC Pin = 5V, 6.GND Pin = GND)
//Wichtige Info/Lautsprecherpin:--> Bei den Arduino Mega 2560 muessen die Anschluesse so gewaehlt werden--> (1.GND Pin = GND, 2."Sound" Pin = 5|6|11|46)

#include <SD.h>                     //Die Bibliothek "SD.h" aufrufen, damit die SD-Karte erkannt wird.
#include <TMRpcm.h>                 //Die Bibliothek "TMRpcm.h" aufrufen, damit man einen Sound ausgeben kann.
TMRpcm lautsprecher;                //Es wird ein TMRpcm-Objekt("lautsprecher") erzeugt. Im Anschluss wird noch der Anschlusspin/Ausgabepin festgelegt, desweiteren werden Methoden, wie z.B. Lautstaerkenregelung ausgefuehrt.

const int CS = 53;                  //Der Anschlusspin "CS" des "MicroSD Card Adapter" ist als Identitaets Ueberpruefungs Verbindung der SD-Karte zu sehen.
//const int SCK = 52;               //Nur zur Vollstaendigkeit im Programm aufgelistet.--> hat keinen Einfluss auf das programmierte
//const int MOSI Pin=51;            // " "
//const int MISO Pin=50;            // " "

//Anschluesse RGB-LED + Variablen:
//Die RGB-LED ist eine besondere Led, denn sie kann in verschiedenen Farben leuchten. Die drei Hauptfarben Pins sind im Anschluss beschrieben.--> 1.Rot 2.Gruen 3.Blau
//Durch die Kompination von zwei oder mehreren aktiven Pins koennen mehrere Farben erzeugt werden. Beispiel: Rot + Gruen = Gelb
const int LED_RED = 9;              //Der Anschluess "LED_RED" hat die Faehigkeit, die Led rot leuten zu lassen.
const int LED_GREEN = 10;           //Der Anschluess "LED_GREEN" hat die Faehigkeit, die Led gruen leuten zu lassen.
const int LED_BLUE = 4;             //Der Anschluess "LED_BLUE" hat die Faehigkeit, die Led blau leuten zu lassen.
const int BREAK = 1000;             //Die Variable "BREAK", soll eine Verzoegerungszeit darstellen.
const int SHINE = 150;              //Die Variable "SHINE" bestimmt die Leuchtkraft der Led. Zahlenwert--> (min. 0 - max. 255)--> Spannungswert--> 0-5 Volt

//Laufvariable:
boolean run = true;                 //Die Variable "run" ist die Laufvariable des Programms. Sie ist die Laufbedingung fuer fast alle Schleifen!

void setup() {

  Serial.begin(9600);               //"Serial.begin()" dient als Ausgabehilfe, zur Problem Loesung oder zur Informations Ergaenzung. Ausgabeoberflaeche ist der "Serieller Monitor".--> (siehe oben rechts, die Lupe)

  //Anschluesse H-Bright L298N festlegen:
  pinMode(ENA, OUTPUT);             //Alle Schnittstellen/Pins des Mikrocontroller zur "H-Bright" sind "OUTPUTS", diese liefern die noetigen Informationen zur Raedersteuerung.(Ausnahme GND)
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);

  //Anschluesse IR Compound Eye festlegen:
  pinMode(TOP, INPUT);              //Alle Schnittstellen/Pins des Mikrocontroller zum "IR Compound Eye" sind "INPUTS", mit Ausnahme der "LEDS".
  pinMode(LEFT, INPUT);             //Die "INPUTS" liefern die noetigen Informationen fuer die Auswertung der IR-Sensoren und der "OUTPUT" versorgt die "LEDS" mit Energie.
  pinMode(BOTTOM, INPUT);
  pinMode(RIGHT, INPUT);
  pinMode(LEDS, OUTPUT);

```

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

```
  //Anschluess Servo festlegen:
  headServo.attach(PIN_SERVO);     //Mit der Methode "attach()" wird festgelegt, dass der "PIN_SERVO" die Steuerschnittstelle zwischen den Servo("headServo") und den Microcontroller ist.
  headServo.write(INITIAL_POS);    //Mit der Methode "write()" wird den Servo-Objekt, "headServo", eine Anfahrposition "INITIAL_POS" vorgegeben. Es handelt sich um die Start/Grundstellung.

  //Anschluss Lautsprecher + Identitaetskontrolle:
  lautsprecher.speakerPin = 5;                    //Hier wird den TMRpcm-Objekt, "lautsprecher", der Ausgabepin "5" zugeteilt.
  lautsprecher.setVolume(5);                      //Das  TMRpcm-Objekt, "lautsprecher", soll die Lautstaerke "5" erhalten.--> "setVolume(5)"--> Erfahrungswert(Min.0 - Max.7)
  if (!SD.begin(CS)) {                            //Idaentitaetspruefung mit der Methode "SD.begin()": Besteht keine Verbindung von "CS" zur SD-Karte, dann....
    Serial.println("SD Auslesefehler");           //...soll am Serieller Monitor "SD Auslesefehler" ausgegeben werden und...
    return;                                       //..."true" zurueck geliefert werden.

  } else {                                        //Wenn aber eine Verbindung besteht, dann...
    Serial.println("SD anwesend");                //...soll am Serieller Monitor "SD anwesend" ausgegeben werden.
  }

  //Anschluess RGB-LED festlegen:
  pinMode(LED_RED, OUTPUT);        //Alle Schnittstellen/Pins des Mikrocontroller zur "RGB-LED" sind "OUTPUTS", diese liefern die noetigen Informationen zur Leuchtfarbe und Leuchtkraft.(Kathode GND)
  pinMode(LED_GREEN, OUTPUT);
  pinMode(LED_BLUE, OUTPUT);

}
void loop() {
  //Fuer die Steuerung des "R2D2" wird eine Bluetooth Verbindung mit den Smartphone hergestellt. Hierzu muss eine App installiert werden, in diesen Fall
  //ist es die App "Arduino Bluetooth". Nach den erfolgreichen verbinden via Bluetooth (Bluetooth-Modulename = HC-05), koennen nun Befehlen versendet und ausgelesen werden.
  //In der App steht einen das Menu mit den vier Wahloptionen zur Verfuegung.--> 1."Led Controller" 2."Terminal" 3."Control pad" 4."Buttons"
  //Wir nutzen 3."Control pad" und 4."Buttons". Hier koennen Tasten wie z.B "X,O,>,<" usw. mit Ziffern, Buchstaben oder Zeichen hinterlegt
  //werden, die nach den druecken--> versenden--> am Mikrocont. als Byte empfangen werden. Je nach Ziffer, Buchstabe oder Zeichen wird ein spezielles Unterprogramm ausgefuehrt.

  //Regestrierung von neuen Befehlen, durch den Bediener:
  if (Serial.available()) {                       //Wenn ein neues Byte regestriert wurde (Serial.available())...
    bluetoothValue = Serial.read();               //..., soll mit der Hilfe der Methode "Serial.read()", dieser ausgelesen und in die Variable "bluetoothValue" gespeichert werden.
    Serial.print("bluetoothValue: ");
    Serial.println(bluetoothValue);               //Ausgabe von "bluetoothValue: (Byte)" am Serieller Monitor mit anschließenden Zeilenumbruch.
  }

  //Bevor der "R2D2" irgendeinen Befehl entgegen nehmen kann, muess das System hochgefahren werden. Der Button "startSys" in der App sendet, wenn er gedrueckt wird, eine '2'.
  if (bluetoothValue == '2') {                    //Wenn "bluetoothValue" gleich '2'--> (R2D2 hochfahren/App: Taste--> "startSys") ist, dann...
    startSystem();                                //... wird die Methode "startSystem()" ausgefuehrt.--> (Strg+L = 196)
    changeRun();                                  //Die Methode "changeRun()" aendert die Laufbedingung(run). Hier--> run = true (Strg+L = 325)

    while (run) {                                 //In der while-Schleife werden die Befehle des Bediener entgegen genommen und ausgefuehrt, solange "run" gleich "true" ist.

     //Regestrierung von neuen Befehlen, durch den Bediener:
     if (Serial.available()) {                    //Wenn ein neues Byte regestriert wurde (Serial.available())...
       bluetoothValue = Serial.read();            //..., soll mit der Hilfe der Methode "Serial.read()", dieser ausgelesen und in die Variable "bluetoothValue" gespeichert werden.
       Serial.print("bluetoothValue: ");
       Serial.println(bluetoothValue);            //Ausgabe von "bluetoothValue: (Byte)" am Serieller Monitor mit anschließenden Zeilenumbruch.
     }

     //Ab hier wird ueberprueft, ob ein Befehl vom "Control pad" vorliegt(case 1 - case 7), oder eine Taste von "Buttons" Menue gedrueckt wurde(case 8 - case 10).
     switch (bluetoothValue) {                    //In der switch-case Kontrollstruktur heißt, die zu ueberpuefende Variable "bluetoothValue". Hier wird verglichen, ob das gesendete Zeichen einen Befehl zugeordnet ist.

       case 'f'://case 1                          //Wenn "bluetoothValue" gleich 'f'--> (Vorwaertsfahren/App: Pfeil nach oben) ist, dann...
         engineSpeed = 255;                       //...wird "engineSpeed" mit "255" beschrieben. Anmerkung(*Dieser Zeile dient zur besseren Uebersicht in der Loop, insbesonders fuer den Programmierer).
         engineRotation("front");                 //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "front" ausgefuehrt, diese sorgt fuer die Vorwaertsbewegung. (Strg+L = 329)
         break;                                   //Fall dieser case zutrifft, soll die switch-case verlassen werden.

       case '<'://case 2                          //Wenn "bluetoothValue" gleich '<'--> (Linksfahren/App: Pfeil nach links) ist, dann...
         engineSpeed = 255;                       //...wird "engineSpeed" mit "255" beschrieben. Anmerkung(*siehe oben " ").
         engineRotation("left");                  //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "left" ausgefuehrt, diese sorgt fuer die Linksbewegung.
         break;                                   //...switch-case wird verlassen.

       case 'b'://case 3                          //Wenn "bluetoothValue" gleich 'b'--> (Rueckwaertsfahren/App: Pfeil nach unten) ist, dann...
         engineSpeed = 255;                       //...wird "engineSpeed" mit "255" beschrieben. Anmerkung(*siehe oben " ").
         engineRotation("back");                  //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "back" ausgefuehrt, diese sorgt fuer die Rueckwartsbewegung.
         break;                                   //...switch-case wird verlassen.

       case'>'://case 4                           //Wenn "bluetoothValue" gleich '>'--> (Rechtsfahren/App: Pfeil nach rechts) ist, dann...
         engineSpeed = 255;                       //...wird "engineSpeed" mit "255" beschrieben. Anmerkung(*siehe oben " ").
         engineRotation("right");                 //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "right" ausgefuehrt, diese sorgt fuer die Rechtsbewegung.
         break;                                   //...switch-case wird verlassen.

       case'0'://case 5                           //Wenn "bluetoothValue" gleich '0' --> (keine Bewegungsausfuehrung/App: jede Entlastung einer Taste ist mit '0' beschrieben) ist, dann...
         engineSpeed = 0;                         //...wird "engineSpeed" mit "0" beschrieben. Anmerkung(*siehe oben  " ").
         engineRotation("stand");                 //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "stand" ausgefuehrt, diese sorgt fuer keine Bewegung.
         break;                                   //...switch-case wird verlassen.

       case'K'://case 6                           //Wenn "bluetoothValue" gleich 'K'--> (Kopf dreht sich nach rechts/App: Kreis-Taste--> (Playstation-Controller)) ist, dann...
         servoRotation("plus");                   //...wird die Methode "servoRotation" mit den Uebergabewert "plus" ausgefuehrt, diese sorgt fuer eine Kopfbewegung nach rechts des "R2D2". (Strg+L = 493)
         break;                                   //...switch-case wird verlassen.

       case'V':// case 7                          //Wenn "bluetoothValue" gleich 'V'--> (Kopf dreht sich nach links/App: Viereck-Taste--> (Playstation-Controller)) ist, dann...
         servoRotation("minus");                  //...wird die Methode "servoRotation" mit den Uebergabewert "minus" ausgefuehrt, diese sorgt fuer eine Kopfbewegung nach links des "R2D2". (Strg+L = 493)
         break;                                   //...switch-case wird verlassen.

       case'1'://case 8                           //Wenn "bluetoothValue" gleich '1'--> (Objektverfolgung aktiviert/App: Taste--> "follow") ist, dann...
         followObject();                          //...wird die Methode "followObject()" ausgefuehrt. (Strg+L = 374)
         break;                                   //...switch-case wird verlassen.
```

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

```
      case'D'://case 9                    //Wenn "bluetoothValue" gleich 'D'--> (Darth Vader Panikmodus/App: Taste--> "DarthVader") ist, dann...
        panicMode();                      //....wird die Methode "panicMode()" ausgefuehrt. (Strg+L = 448)
        break;                            //...switch-case wird verlassen.

      case'X'://case 10                   //Wenn "bluetoothValue" gleich 'X'--> (Programm herunterfahren/App Taste--> "shutDownSys") ist, dann...
        shutDownSystem();                 //...wird die Methode "shutDownSystem()" ausgefuehrt. (Strg+L = 483)
        changeRun();                      //Die Methode "changeRun()" aendert die Laufbedingung(run). Hier--> run = false--> while-Schleife wird im Anschluss verlassen--> Bediener kann keine Befehle mehr geben, außer er startet das System erneut.
        break;                            //...switch-case wird verlassen.


    }
  }
}
}//ENDE LOOP

void startSystem() {                      //In der Methode "startSystem", soll das Hochfahren des System simuliert werden. Hilfsmittel: Soundausgabe, RGB-LED, Servo und DC Motoren
 lautsprecher.play("19.wav");             //Der "lautsprecher" soll nun den Sound mit den Dateinamen "19" ausgeben.--> "play("19.wav")"--> Nach den Dateinamen wird noch ein ".wav" hinzugefuegt.
 lightCheck();                            //Die Methode "lightCheck()", wird aufgerufen. (Strg+L = 264)
 sound();                                 //Die Methode "sound()", wird aufgerufen. (Strg+L = 320)

 lautsprecher.play("7.wav");              //Der "lautsprecher" soll den Sound mit den Dateinamen "7" ausgeben.
 sound();                                 //Wiedergabezeit-->...

 shine("gelb");                           //Jetzt wird die Methode "shine()" mit den Uebergabewert "gelb" ausgefuehrt, diese sorgt dafuer das die RGB-LED gelb leuchtet. (Strg+L = 287)
 headServo.write(POS_MIN);                //"headServo" soll die Position "POS_MIN" anfahren.
 delay(2000);                             //Eine Verzoegerungszeit von 2000ms = 2Sekunden, damit der Servo die Position anfahren kann.--> Verfahrzeit = 2sek
 run = true;                              //Die Laufvariable muss hier einmal auf "true" gesetzt werden, damit nach einen herunter-, und anschließenden hochfahren des Systems ein festen Laufbedinungswert hat.--> "changeRun()" ist keine Loesung

 do {                                     //In der do/while-Schleife wird nun ein Test von Elektrischen Bauteile simuliert. Erste Schleife--> lange Servobewegung || POS_MIN-POS_MAX
   if (headServo.read() == POS_MIN) {     //Wenn der "headServo" die Position von "POS_MIN" erreicht, dann...
     headServo.write(POS_MAX);            //..soll "headServo" die Position "POS_MAX" anfahren.
     delay(2000);                         //Verfahrzeit = 2sek
     shine("rot");                        //"shine("rot")"--> RGB-LED leuchtet rot.

   } else if (headServo.read() == POS_MAX) {  //Wenn aber der "headServo" die Position von "POS_MAX" erreicht, dann...
     lautsprecher.play("18.wav");         //..soll der "lautsprecher" den Sound mit den Dateinamen "18" ausgeben.
     sound();                             //Wiedergabezeit-->...

     //Mit POS_MIN und POS_MAX wurde die maximal erlaubten Drehwinkel angefahren, nun soll noch die Genauigkeit des Servos durch zwei kleinere Winkel ueberprueft werden.|| 50° und 130°
     headServo.write(50);                 //Der "headServo" soll die Position "50" Grad anfahren.
     delay(2000);                         //Verfahrzeit = 2sek
     shine("gruen");                      //"shine("gruen")"--> RGB-LED leuchtet gruen.
     changeRun();                         //Die Methode "changeRun()" aendert die Laufbedingung(run). Hier--> run = false (Strg+L = 325)
   }
 } while (run);                           //In der do/while-Schleife wird der "R2D2" hochgefahren, solange "run" gleich "true" ist.--> Erste Schleife
 changeRun();                             //Die Methode "changeRun()" aendert die Laufbedingung(run). Hier--> run = true

 do {                                     //In der do/while-Schleife wird nun ein Test von Elektrischen Bauteile simuliert. Zweite Schleife--> kurze Servobewegung mit anschließender Grundstellung.|| 50°-130°-90°
   if (headServo.read() == 50) {          //Wenn der "headServo" die Position von "50" Grad erreicht, dann...
     headServo.write(130);                //...soll er die Position "130" Grad anfahren.
     delay(1500);                         //Verfahrzeit = 1.5sek
     shine("tuerkis");                    //"shine("tuerkis")"--> RGB-LED leuchtet tuerkis.

   } else if (headServo.read() == 130) {  //Wenn aber der "headServo" die Position von "130" Grad erreicht, dann...
     lautsprecher.play("9.wav");          //...soll der "lautsprecher" den Sound mit den Dateinamen "9" ausgeben.
     sound();                             //Wiedergabezeit-->...
     headServo.write(INITIAL_POS);        //Der "headServo" soll die Position "INITIAL_POS" anfahren.--> Grundstellung
     delay(1500);                         //Verfahrzeit = 1.5sek
     shine("lila");                       //"shine("lila")"--> RGB-LED leuchtet lila.

   } else if (headServo.read() == INITIAL_POS) { //Wenn aber der "headServo" die Position "INITIAL_POS" erreicht, dann...
     lautsprecher.play("6.wav");          //...soll der "lautsprecher" den Sound mit den Dateinamen "6" ausgeben.
     sound();                             //Wiedergabezeit-->...
     shine("blau");                       //"shine("blau")"--> RGB-LED leuchtet blau.--> Betriebsfarbe
     changeRun();                         //Die Methode "changeRun()" aendert die Laufbedingung(run). Hier--> run = false
   }
 } while (run);                           //In der do/while-Schleife wird der "R2D2" hochgefahren, solange "run" gleich "true" ist.--> Zweite Schleife

 //Als letztes werden die DC-Motoren auf ihre Funktion ueberpueft. Ein kurzes links und rechts fahren wird simuliert.
 delay(2000);                             //Verzoegerungszeit = 2sek

 engineSpeed = 230;                             //"engineSpeed" wird mit "230" beschrieben. Anmerkung:(*Dieser Zeile dient zur besseren Uebersicht in der Loop, insbesonders fuer den Programmierer).
 engineRotation("left");                        //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "left" ausgefuehrt, diese sorgt fuer die Linksbewegung. (Strg+L = 329)
 delay(2000);                                   //Verfahrzeit = 2sek

 engineSpeed = 230;                             //"engineSpeed" wird mit "230" beschrieben. Anmerkung:(*siehe oben).
 engineRotation("right");                       //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "right" ausgefuehrt, diese sorgt fuer die Rechtsbewegung. (Strg+L = 329)
 delay(2000);                                   //Verfahrzeit = 2sek

 engineSpeed = 0;                               //"engineSpeed" wird mit "0" beschrieben. Anmerkung:(*siehe oben).
 engineRotation("stand");                       //Jetzt wird die Methode "engineRotation()" mit den Uebergabewert "stand" ausgefuehrt, diese sorgt fuer keine Bewegung.
}

void lightCheck() {                             //In der Methode "lightCheck" wird die Funktionen der RGB-LED ueberprueft.
 shine("rot");                                  //Jetzt wird die Methode "shine()" mit den Uebergabewert "rot" ausgefuehrt, diese sorgt dafuer das die RGB-LED rot leuchtet. (Strg+L = 287)
 delay(BREAK);                                  //Kurze Verzoegerungszeit von "BREAK" in Millisekunden. || 1000msek = 1sek

 shine("gruen");                                //RGB-LED--> Leuchtfarbe: gruen
 delay(BREAK);                                  //Verzoegerungszeit von "BREAK" in Millisek.

 shine("blau");                                 //RGB-LED--> Leuchtfarbe: blau
 delay(BREAK);                                  //Verzoegerungszeit von "BREAK" in Millisek.
```

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

```
  shine("gelb");                          //RGB-LED--> Leuchtfarbe: gruen
  delay(BREAK);                           //Verzoegerungszeit von "BREAK" in Millisek.

  shine("tuerkis");                       //RGB-LED--> Leuchtfarbe: gruen
  delay(BREAK);                           //Verzoegerungszeit von "BREAK" in Millisek.

  shine("lila");                          //RGB-LED--> Leuchtfarbe: gruen
  delay(BREAK);                           //Verzoegerungszeit von "BREAK" in Millisek.

  shine("null");                          //Die RGB-LED leuchtet nicht mehr.
  delay(BREAK);                           //Verzoegerungszeit von "BREAK" in Millisek.
}

void shine(String color) {                //Die Methode "shine" bekommt einen String uebergeben, der den Befehl fuer die Leuchtfarbe der RGB-Led beinhaltet.--> leichter nachvollziehbar
  analogWrite(LED_RED, LOW);              //Um keine ungewollten Farbkombinationen zu erhalten, werden im ersten Schritt alle Pins der Led kurzzeitig auf "LOW" gesetzt.--> kein leuchten R-
  analogWrite(LED_GREEN, LOW);            //G-
  analogWrite(LED_BLUE, LOW);             //B-

  if (color == "rot") {                   //Wenn "color" gleich "rot" ist, dann soll die Led rot leuchten.
    analogWrite(LED_RED, SHINE);          //Die Methode "analogWrite()" setzt einen Pin, in diesen Fall "LED_RED", auf den Variablenwert von "SHINE" und der Led wird ein Rotanteil hinzugeschalten.(0-5 Volt) R+

  } else if (color == "gruen") {          //Wenn aber "color" gleich "gruen" ist, dann soll die Led gruen leuchten.
    analogWrite(LED_GREEN, SHINE);        // " " ..."LED_GREEN"...ein Gruenanteil wird hinzugeschalten. G+

  } else if (color == "blau") {           //Wenn aber "color" gleich "blau" ist, dann soll die Led blau leuchten.--> Betriebsfarbe
    analogWrite(LED_BLUE, SHINE);         // " " ..."LED_BLUE"...ein Blauanteil wird hinzugeschalten. B+

  } else if (color == "gelb") {           //Wenn aber "color" gleich "gelb" ist, dann soll die Led gelb leuchten.
    analogWrite(LED_RED, SHINE);          // " " ..."LED_RED" + "LED_GREEN"...ein Rotanteil und ein Gruenanteil wird hinzugeschalten.R+ G+ --> Gelb
    analogWrite(LED_GREEN, SHINE);

  } else if (color == "tuerkis") {        //Wenn aber "color" gleich "tuerkis" ist, dann soll die Led tuerkis leuchten.
    analogWrite(LED_GREEN, SHINE);        // " " ..."LED_GREEN" + "LED_BLUE"...ein Gruenanteil und ein Blauanteil wird hinzugeschalten.G+ B+ --> Tuerkis
    analogWrite(LED_BLUE, SHINE);

  } else if (color == "lila") {           //Wenn aber "color" gleich "lila" ist, dann soll die Led lila leuchten.
    analogWrite(LED_BLUE, SHINE);         // " " ..."LED_BLUE" + "LED_RED"...ein Blauanteil und ein Rotanteil wird hinzugeschalten.B+ R+ --> Lila
    analogWrite(LED_RED, SHINE);

  } else if (color == "null") {           //Wenn aber "color" gleich "null" ist, dann soll die Led nicht leuchten.
    analogWrite(LED_RED, LOW);            //Die Pins: "LED_RED", "LED_GREEN", und "LED_BLUE" werden auf "LOW" geschalten.--> R- G- B-
    analogWrite(LED_GREEN, LOW);
    analogWrite(LED_BLUE, LOW);
  }
}

void sound() {                            //In der Methode "sound" wird sichergestellt, dass es wahrend der Soundausgabe kein weiteres Bauteil mit großen Energieverbrauch laeger aktiv sind und es somit zu keinen Spannungsabfall kommt.--> sonst Stoergeraesche
  do {                                    //Die do/while-Schleife wird so lange ausgefuehrt, bis...
  } while (lautsprecher.isPlaying());     //...der "lautsprecher" die Wiedergabe beendet.--> Datei is komplett abspielt --> "isPlaying()" liefert eine "0" oder auch "false"
}

void changeRun() {                        //In der Methode "changeRun" wird die Laufbedingung "run" geaendert.
  run = !run;                             //Zum Beispiel, wenn run = true--> nach dieser Zeile run = false;
}

void engineRotation(String movement) {    //Die Methode "engineRotation" bekommt einen String uebergeben, der den Befehl fuer die Bewegung der Motoren beinhaltet.--> leichter nachvollziehbar

  if (movement == "front") {              //Wenn "movement" gleich "front" ist, dann soll der "R2D2" nach vorne fahren.
    analogWrite(ENA, engineSpeed);        //Bestmoegliche Einstellungen herrausgefunden durch Tests. || z.B (engineSpeed < 170)--> Motoren "quaelen sich"--> engineSpeed > 170
    analogWrite(ENB, engineSpeed);        //Fuer weitere Informationen: siehe (//Anschluesse H-Bright L298N/DC Motoren + Variablen:)--> (Strg+L = 1)
    digitalWrite(IN1, HIGH);              //
    digitalWrite(IN2, LOW);               //Aus IN1+IN2--> gegen Uhrzeigersinn
    digitalWrite(IN3, HIGH);              //
    digitalWrite(IN4, LOW);               //Aus IN3+IN4--> gegen Uhrzeigersinn

  }  else if (movement == "left") {       //Wenn aber "movement" gleich "left" ist, dann soll der "R2D2" nach links fahren.
    analogWrite(ENA, engineSpeed);        //Bestmoegliche Einstellungen herrausgefunden durch Tests. || z.B "linkes Rad" fixiert && "rechtes Rad" drehen--> langsame Drehung, um das fixierte Rad.
    analogWrite(ENB, engineSpeed);        //Fuer weitere Informationen: siehe (//Anschluesse H-Bright L298N/DC Motoren + Variablen:)--> (Strg+L = 1)
    digitalWrite(IN1, HIGH);              //
    digitalWrite(IN2, LOW);               //Aus IN1+IN2--> gegen Uhrzeigersinn
    digitalWrite(IN3, LOW);               //
    digitalWrite(IN4, HIGH);              //Aus IN3+IN4--> mit den Uhrzeigersinn--> aus IN1+IN2+IN3+IN4 schnelle Drehung

  } else if (movement == "back") {        //Wenn aber "movement" gleich "back" ist, dann soll der "R2D2" nach hinten fahren.
    analogWrite(ENA, engineSpeed);        //Bestmoegliche Einstellungen herrausgefunden durch Tests. || z.B (engineSpeed < 170)--> Motoren "quaelen sich"--> engineSpeed > 170
    analogWrite(ENB, engineSpeed);        //Fuer weitere Informationen: siehe (//Anschluesse H-Bright L298N/DC Motoren + Variablen:)--> (Strg+L = 1)
    digitalWrite(IN1, LOW);               //
    digitalWrite(IN2, HIGH);              //Aus IN1+IN2--> mit den Uhrzeigersinn
    digitalWrite(IN3, LOW);               //
    digitalWrite(IN4, HIGH);              //Aus IN3+IN4--> mit den Uhrzeigersinn

  } else if (movement == "right") {       //Wenn aber "movement" gleich "right" ist, dann soll der "R2D2" nach rechts fahren.
    analogWrite(ENA, engineSpeed);        //Bestmoegliche Einstellungen herrausgefunden durch Tests. || z.B "linkes Rad" drehen && "rechtes Rad" fixiert --> langsame Drehung, um das fixierte Rad.
    analogWrite(ENB, engineSpeed);        //Fuer weitere Informationen: siehe (//Anschluesse H-Bright L298N/DC Motoren + Variablen:)--> (Strg+L = 1)
    digitalWrite(IN1, LOW);               //
    digitalWrite(IN2, HIGH);              //Aus IN1+IN2--> mit den Uhrzeigersinn
    digitalWrite(IN3, HIGH);              //
    digitalWrite(IN4, LOW);               //Aus IN3+IN4--> gegen Uhrzeigersinn--> aus IN1+IN2+IN3+IN4 schnelle Drehung

```

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

```
} else if (movement == "stand") {      //Wenn aber "movement" gleich "stand" ist, dann soll der "R2D2" nicht fahren.
  analogWrite(ENA, engineSpeed);       //Bestmoegliche Einstellungen herrausgefunden durch Tests. Achtung "engineSpeed" hat hier den Wert "0"!
  analogWrite(ENB, engineSpeed);       //Fuer weitere Informationen: siehe (//Anschluesse H-Bright L298N/DC Motoren + Variablen:)--> (Strg+L = 1)
  digitalWrite(IN1, LOW);              //
  digitalWrite(IN2, LOW);              //Aus IN1+IN2--> keine Bewegung
  digitalWrite(IN3, LOW);              //
  digitalWrite(IN4, LOW);              //Aus IN3+IN4--> keine Bewegung

}
}

void followObject() {                  //In der Methode "followObject()" werden die IR-Sensoren ausgewertet, durch die Aktoren Servo und DC-Motor wird eine Objectverfolgung in der X und Y Achse erzeugt.
 readSensor();                         //Der Erste Schritt ist die Sensoren auszuwerten, das wird in der Methode "readSensor()" erledigt.(Strg+L = 417)

 //X-Achsen Regelung = Servo Regelung:
 if (durchschnittsAbstand > MIN_ABSTAND) {  //Im zweiten Schritt wird auf eine Bewegung des Objekts reagiert. Allerdings muss sich das Objekt auch im Erfassungsbereich ("durchschnittsAbstand" > "MIN_ABSTAND") der Sensoren liegen.
   shine("rot");                       //Dann soll die RGB-LED rot leuchten.

   if (links > rechts) {               //Wenn der Wert des linken IR-Sensor("links") groeßer als der Wert des rechten IR-Sensors("rechts") ist, dann...
     servoRotation("minus");           //...wird die Methode "servoRotation()" mit den Uebergabewert "minus" aufgerufen. (Strg+L = 493)

   } else if (links < rechts) {        //Wenn aber der Wert des linken IR-Sensor("links") kleiner als der Wert des rechten IR-Sensors("rechts") ist, dann...
     servoRotation("plus");            //...wird die Methode "servoRotation()" mit den Uebergabewert "plus" aufgerufen.
   }
   //Y-Achsen Regelung = DC-Motoren Regelung:   Im dritten und letzten Schritt wird mit hilfe der DC-Motoren eine noch schnellere Verfolgung in der X-Achse realisiert. Desweiteren soll auch die Y-Achsen Regelung implementiert werden.
   if (posCurrent < 60) {              //Ist "posCurrent" < "60"--> Servo naehert sich den Grenzwert, dann...
     engineSpeed = 210;                //....wird "engineSpeed" mit "210" beschrieben und...   --> engineSpeed = 210--> Erfahrungswert
     engineRotation("right");          //....der "R2D2" dreht sich kurz nach rechts.

   } else if (posCurrent > 120) {      //Ist "posCurrent" > "120"--> Servo naehert sich den Grenzwert, dann...
     engineSpeed = 210;                //....wird "engineSpeed" mit "210" beschrieben und...   --> engineSpeed = 210--> Erfahrungswert
     engineRotation("left");           //....der "R2D2" dreht sich kurz nach links.
   }
   delay(30);                          //Verfahrzeit = 30msek

 } else {                              //Wenn das Objekt nicht im Erfassungsbereich liegt, dann... --> Erfassungsbereich muss > 0 | Erfas. = "durchschnittsAbstand" - "MIN_ABSTAND"
   shine("blau");                      //....soll die RGB-LED rot leuchten.
 }

 //Y-Achsen Regelung = DC-Motoren Regelung:--> Desweiteren soll auch die Y-Achsen Regelung implementiert werden.
 if (durchschnittsAbstand > MIN_ABSTAND && durchschnittsAbstand < MIN_ABSTAND2) {          //Wenn "durchschnittsAbstand" > MIN_ABSTAND && "" < MIN_ABSTAND2, dann...
   engineSpeed = 180;                                                                     //....wird"engineSpeed" mit "180" beschrieben und...   --> engineSpeed = 180--> Erfahrungswert
   engineRotation("front");                                                               //....der "R2D2" faehrt kurz nach vorne.

 } else if (oben > MAX_ABSTAND && links > MAX_ABSTAND && unten > MAX_ABSTAND && rechts > MAX_ABSTAND) {       //Wenn aber die einzelnen Werte der IR-Sensoren groeßer als der "MAX_ABSTAND" sind, dann...
   engineSpeed = 180;                                                                                       //....wird "engineSpeed" mit "180" beschrieben und...   --> engineSpeed = 180--> Erfahrungswert
   engineRotation("back");                                                                                  //....der "R2D2" faehrt kurz nach hinten.

 } else {                                                                                                   //Wenn aber die einzelnen Werte der IR-Sensoren kleiner als der "MIN_ABSTAND" sind, dann...
   engineSpeed = 0;                                                                                         //....wird "engineSpeed" mit "0" beschrieben und...
   engineRotation("stand");                                                                                 //....der "R2D2" bleibt stehen.
 }
}

void readSensor() {                          //In der Methode "readSensor" werden die einzelnen Sensoren ausgelesen und in Variablen gespeichert.--> Auswertung

 digitalWrite(LEDS, HIGH);                   //Die 4 Leds am "IR Compound Eye" werden auf "HIGH" gesetzt und leuchten.--> Infrarotstrahlen--> nicht wahrzunehmen mit den menschlichen Auge
 delay(5);                                   //Ein kurze Wartezeit von 5 Millisekunden, dass die Infrarotstrahlen genuegend Zeit haben, vom reflektierten Objekt zurueck auf die Sensoren zu strahlen.

 oben = analogRead(TOP);                     //Hier findet die Auswertung der analogen Eingaenge des Microcontroller statt. Jeder Eingangswert wird in die dazugehoerige Variable gespeichert.
 links = analogRead(LEFT);                   //Die Methode "analogRead()" kann Spannungen zwischen 0 und 5 Volt wahrnehmen, dass entspricht Werte zwischen 0 und 1023.
 unten = analogRead(BOTTOM);
 rechts = analogRead(RIGHT);
 durchschnittsAbstand = (oben + links + unten + rechts) / 4; //Nach der Auswertung der Eingaenge, wird der "durchschnittsAbstand" aller IR-Sensoren ermittelt.--> Bildung des Mittelwertes
 digitalWrite(LEDS, LOW);                                    //Die 4 Leds am "IR Compound Eye" werden auf "LOW" gesetzt und leuchten nun nicht mehr.
 //Hilfreiche Ausgabe, zum testen der einzelnen Sensoren:
 /*Serial.print("Oben:");                                    //Ausgabe der aktuellen analogWerte am "Serieller Monitor".
   Serial.print(oben);
   Serial.print(", ");

   Serial.print("Links:");
   Serial.print(links);
   Serial.print(", ");

   Serial.print("Unten:");
   Serial.print(unten);
   Serial.print(", ");

   Serial.print("Rechts:");
   Serial.print(rechts);

   Serial.print("\n");
 */
}

void  panicMode() {                          //In der Methode "panikMode", soll eine reale Begegnung von "R2D2" mit "Darth Vader" simuliert werden. Hilfsmittel: Soundausgabe, LED, Servo und DC Motoren
 delay(2000);                                //Verzoegerungszeit = 2sek
 lautsprecher.play("DVcoming.wav");          //Der "lautsprecher" soll nun den Sound mit den Dateinamen "DVcoming" ausgeben.
 if (lautsprecher.isPlaying()) {             //Mit der if-Abfrage wird sichergestellt, dass die Verzoegerungszeit exakt mit der Soundausgabe beginnt.--> SDCard laedt teilweise
   delay(16000);                             //Verzoegerungszeit = 16sek
 }
```

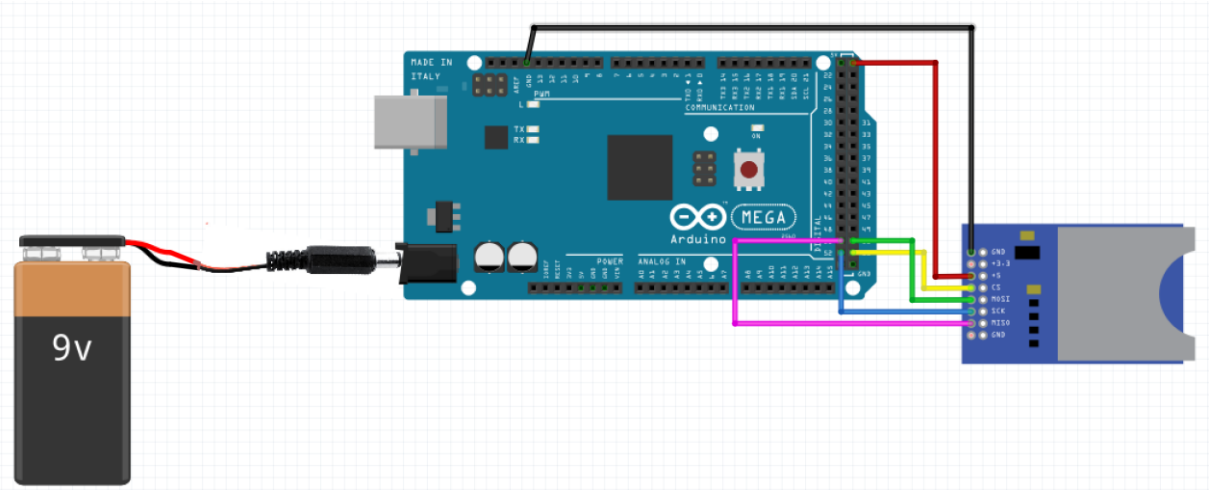
Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

```
  //"R2D2" schaltet in den Ueberlebensprogramm, da er den Atem von "Darth Vader" hoert.
  shine("rot");                                 //RGB-LED--> Leuchtfarbe: rot
  delay(5000);                                  //Verzoegerungszeit = 5sek

  headServo.write(50);                          //Die Servo Bewegung auf "50" und "130" Grad soll ein "Umgebung checken und ueberlegen" darstellen.
  delay(1500);                                  //Verfahrzeit = 1.5sek
  headServo.write(130);
  delay(1500);                                  //Verfahrzeit = 1.5sek
  sound();                                      //Wiedergabezeit-->... insgesamt ca.30sek

  delay(300);                                   //Verzoegerungszeit = 0.3sek
  engineSpeed = 255;                            //"engineSpeed" wird mit "255" beschrieben und...   --> engineSpeed = 255--> max Geschwindigkeit
  engineRotation("front");                      //...der "R2D2" faehrt mit Vollgas geradeaus.
  headServo.write(INITIAL_POS);                 //Der Servo faehrt in die Grundstellung
  delay(3500);                                  //Verfahrzeit/Fluchtzeit = 3.5sek

  engineSpeed = 0;                              //"engineSpeed" wird mit "0" beschrieben und...
  engineRotation("stand");                      //...der "R2D2" bleibt stehen.

  //"R2D2" ist veraengstigt und schaltet sich kurz ab
  delay(2000);                                  //Verzoegerungszeit = 2sek
  lautsprecher.play("scared.wav");              //Der "lautsprecher" soll nun den Sound mit den Dateinamen "scared" ausgeben.
  sound();                                      //Wiedergabezeit-->...
  shutDownSystem();                             //Die Methode "shutDownSystem()" wird aufgerufen. (Strg+L = 483)

  shine("blau");                                //Simulation beendet--> "R2D2" in Standardbetrieb--> Betriebsfarbe
}

void shutDownSystem() {                         //In der Methode "shutDownSystem", soll das Herunterfahren des System simuliert werden. Hilfsmittel: Soundausgabe, LED, Servo und DC Motoren
  headServo.write(INITIAL_POS);                 //Der "headServo" faehrt in die Grundstellung
  delay(2000);                                  //Verzoegerungszeit = 2sek
  engineRotation("stand");                      //"engineRotation()" wird mit den Uebergabewert "stand" ausgefuehrt, diese sorgt fuer keine Bewegung.
  lautsprecher.play("22.wav");                  //Der "lautsprecher" soll nun den Sound mit den Dateinamen "22" ausgeben.
  sound();                                      //Wiedergabezeit-->...
  shine("null");                                //RGB-LED leuchtet nicht mehr.
  delay(3000);                                  //Verzoegerungszeit = 3sek
}

void servoRotation(String movement) {           //Die Methode "servoRotation" bekommt einen String uebergeben, der den Befehl fuer die Drehbewegung des Servos beinhaltet.--> leichter nachvollziehbar

  if (movement == "minus") {                    //Wenn "movement" gleich "minus" ist, dann soll der "R2D2" seinen Kopf nach Links, also Richtung/bis "POS_MIN" drehen.
    posCurrent -= STEP_RANGE;                   //Der aktuellen Position des Servos "posCurrent" wird die "STEP_RANGE" subtrahiert.

    if (posCurrent < POS_MIN) {                 //Ist der Wert von "posCurrent" kleiner als der Grenzwert "POS_MIN", dann...
      posCurrent = POS_MIN;                     //...wird "posCurrent" mit den konstanten Grenzwert von "POS_MIN" beschrieben.--> kein ueberfahren des Grenzwertes
    }

  } else if (movement == "plus") {              //Wenn aber "movement" gleich "plus" ist, dann soll der "R2D2" seinen Kopf nach Rechts, also Richtung/bis "POS_MAX" drehen.
    posCurrent += STEP_RANGE;                   //...wird "STEP_RANGE" mit "posCurrent" addiert.--> posCurrent entspricht der Servoposition in Grad.

    if (posCurrent > POS_MAX) {                 //Ist der Wert von "posCurrent" groeßer als der Grenzwert "POS_MAX", dann...
      posCurrent = POS_MAX;                     //...wird "posCurrent" mit den konstanten Grenzwert von "POS_MAX" beschrieben.--> kein ueberfahren des Grenzwertes
    }
  }
  headServo.write(posCurrent);                  //Nach einigen if-Abfragen wird die Lage des Servos korrigiert.--> X-Achsen Nachstellung mit der Methode "write()", die neue Position von "posCurrent" wird angefahren.
  delay(25);                                    //Verfahrzeit = 25msek
}
//-----------------------------------------------------------------
//-----------------------------------------------------------------
//Methoden finden:
//startSystem()--> (Strg+L 196)
//lightCheck()--> (Strg+L 264)
//shine()--> (Strg+L 287)
//sound()--> (Strg+L 320)
//changeRun()--> (Strg+L 325)
//engineRotation()--> (Strg+L 329)
//followObject()--> (Strg+L 374)
//readSensor()--> (Strg+L 417)
//panicMode()--> (Strg+L 448)
//shutDownSystem()--> (Strg+L 483)
//servoRotation()--> (Strg+L 493)

```

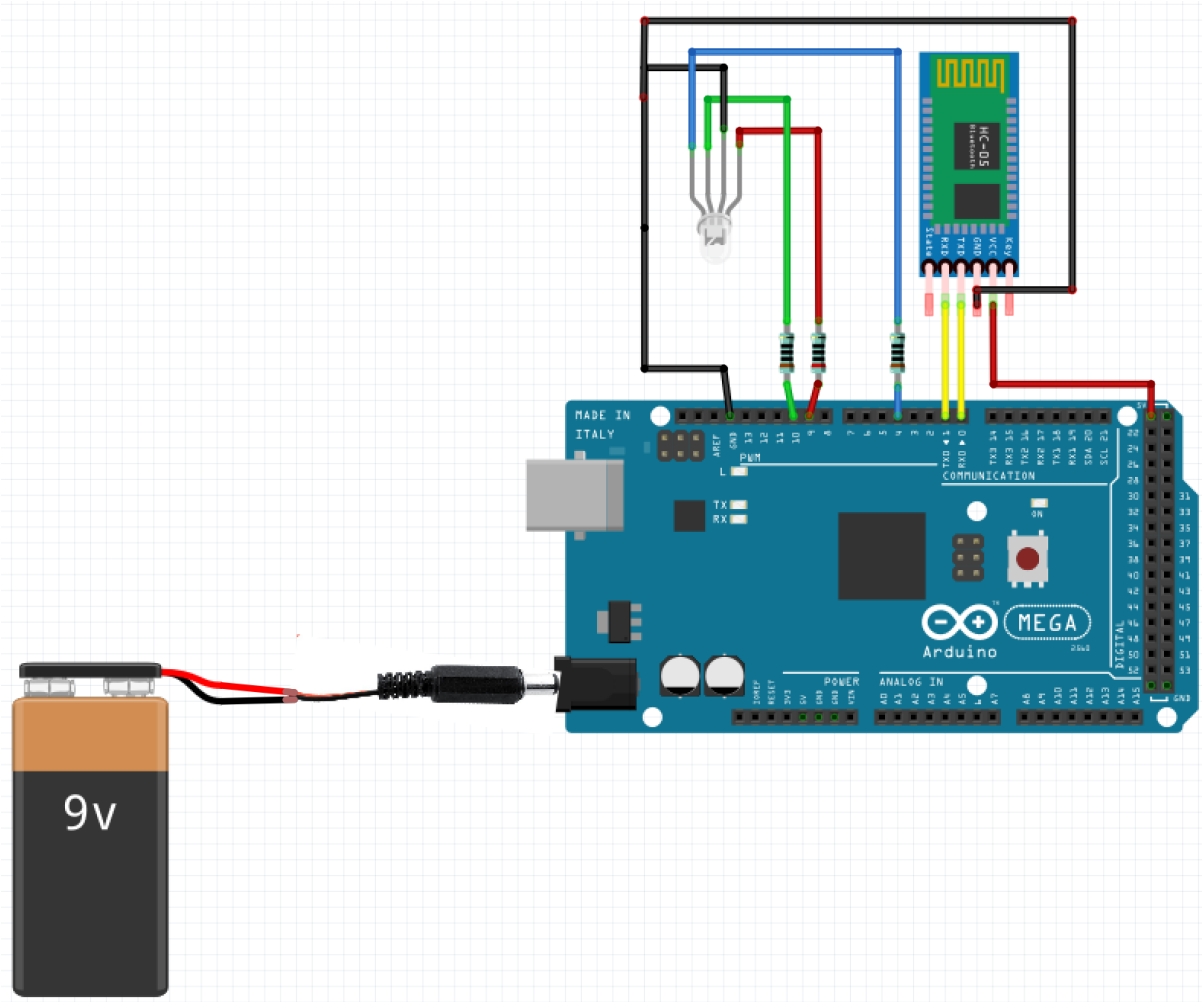
Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

7.2. Fritzing

→ SD-Card-Reader:



→ HC-05 + RGB LED:

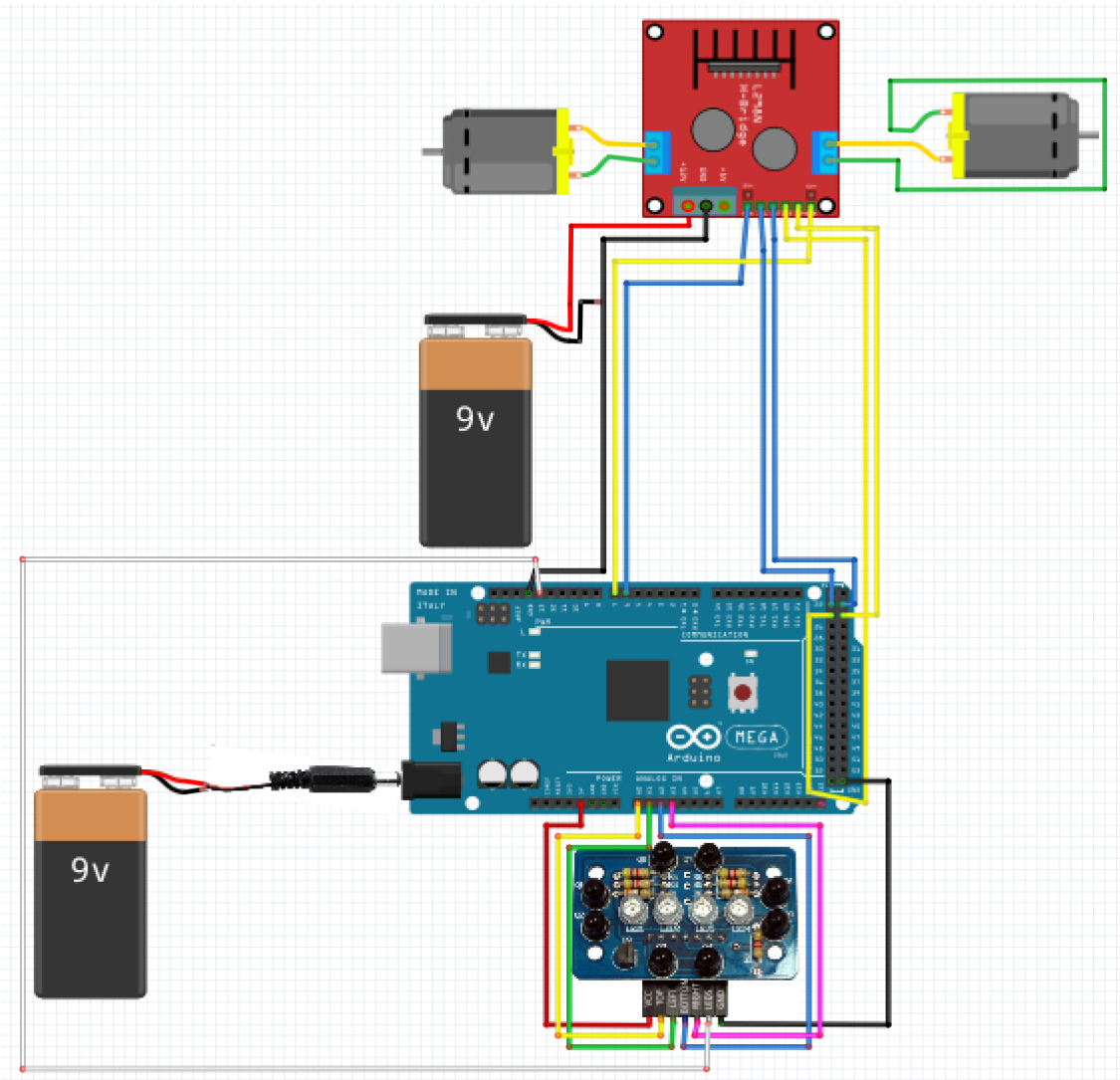


Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

→ IR-Compound-Eye + H-Bridge:

→ Rechter Motor/Steuerleitungen

→ Linker Motor/Steuerleitungen



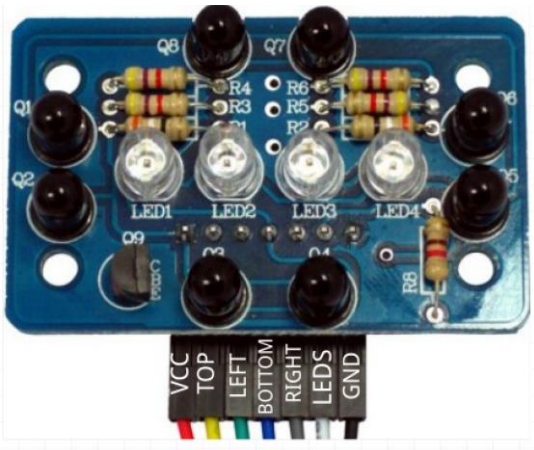
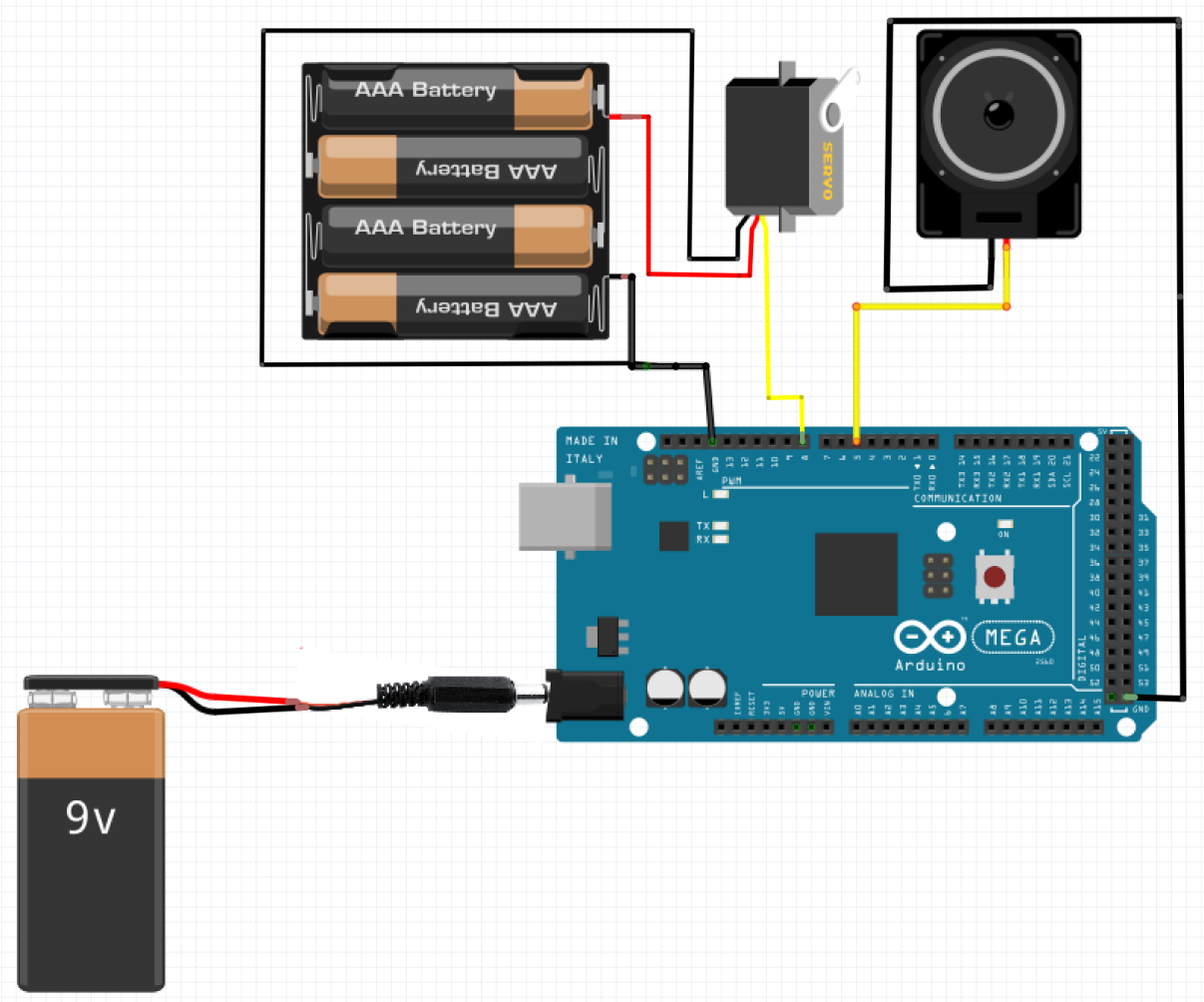


Abbildung 13: „Nahaufnahme des IR-Compound-Eye“

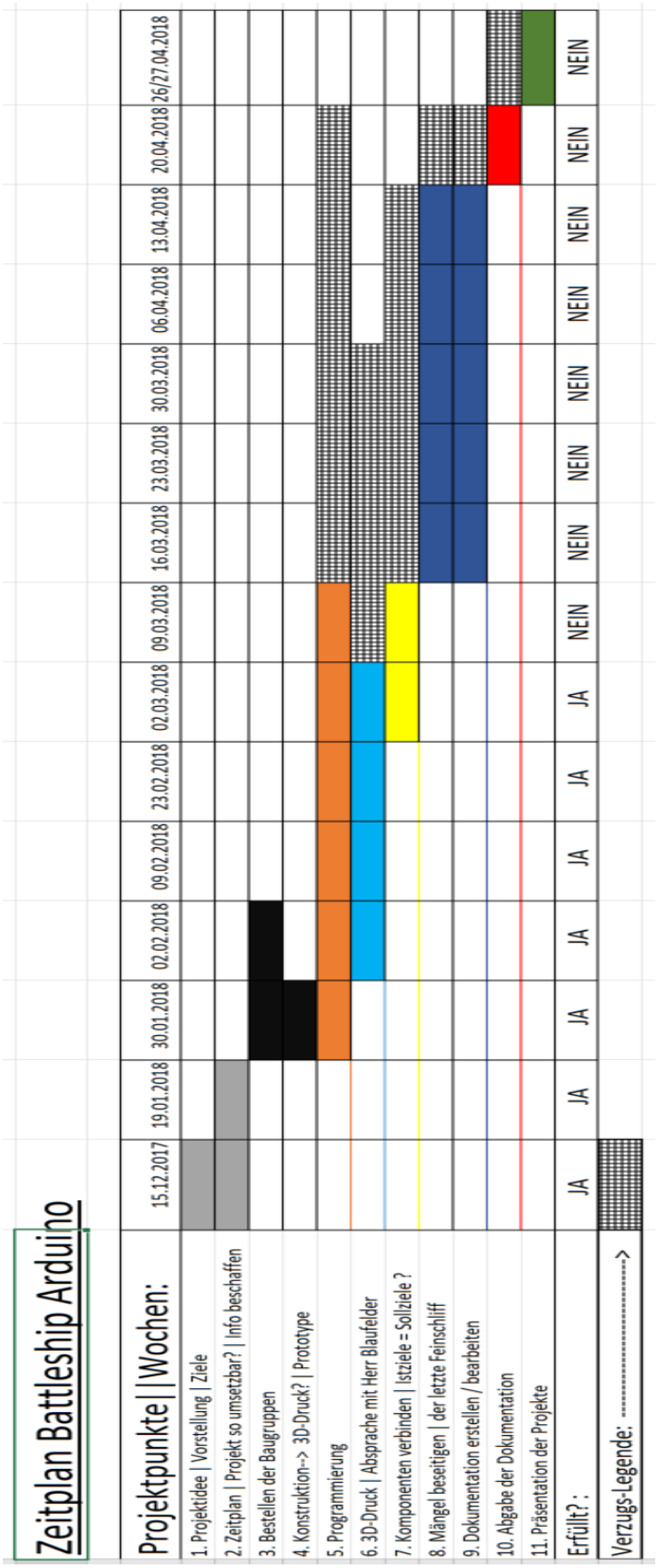
Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

→ Lautsprecher + Servo:



Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

7.3. Zeitplanung



Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024

7.4. Erklärung mit Unterschrift

Ich versichere durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt, alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen, als solche kenntlich gemacht und mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Herzogenaurach den 24.4.2018

_____

Knauer Johannes

Staatlichen Beruflichen Schulzentrum
Herzogenaurach-Höchstadt a. d. Aisch
Friedrich-Weiler-Platz 2
91074 Herzogenaurach
Tel.: 09132/8023
Fax: 09132/8024