

D-O Control & Power Board

Systemdokumentation v2.2

D-O Control & Power Board

Vollstaendige Nutzer-Anleitung.

Version 2.2 — 2026-04-23

Haftungsausschluss

D-O Control & Power Board ist ein DIY-Projekt. Aufbau, Verkabelung, Konfiguration und Betrieb umfassen Arbeiten, die bei unsachgemässer Ausführung Sach- oder Personenschaden verursachen koennen.

Mit der Nutzung dieser Dokumentation, Hard- und Software erkennst du an:

- Du bist alleine verantwortlich fuer alle Arbeiten an deinem System.
- Du verfuegst ueber die noetigen Grundkenntnisse oder holst dir qualifizierte Hilfe.
- Du pruefst Spannung, Polung, Sicherungen und Steckverbindungen vor jedem Einschalten.
- Du verwendest nur passende Komponenten.
- Du beachtest die oertlichen Vorschriften.

Der Autor und alle Beteiligten uebernehmen keinerlei Haftung fuer Schaeden durch falsche Verkabelung, ungeeignete Komponenten, unsachgemaeße Bedienung oder unzureichende Sicherheitsmassnahmen.

Die Dokumentation und Software werden "**wie besehen**" ohne jede Garantie bereitgestellt. Im Zweifel qualifizierte Hilfe einholen bevor am System gearbeitet wird.

Inhaltsverzeichnis

| | |
|---|-----------|
| Haftungsausschluss | 2 |
| Inhaltsverzeichnis | 3 |
| Einfuehrung — Was ist D-O Control & Power Board? | 7 |
| Hauptfunktionen | 7 |
| Schnelleinstieg | 8 |
| Board- und Sketch-Auswahl | 10 |
| Die beiden Boards im Ueberblick | 10 |
| Verfuegbare Sketche | 10 |
| Empfohlene Kombinationen | 11 |
| Upgrade-Hinweis fuer v3.4.1 | 11 |
| Sketch-Upload | 12 |
| Benoetigte Libraries | 12 |
| Arduino IDE Schritt fuer Schritt | 12 |
| Kommandozeilen-Alternative | 12 |
| IMU-Kalibrierung | 13 |
| Kalibrierung durchfuehren | 13 |
| Wann neu kalibrieren | 13 |
| Sender-Setup (FlySky) | 15 |
| 1. Empfaenger-Wahl (iBus-Ausgang spart Kabel) | 15 |
| 2. Empfaenger binden | 15 |
| 3. Kanalzuweisung in D-O | 16 |
| 4. CH3-Luecke schliessen | 16 |
| 5. Failsafe-Verhalten | 16 |
| SBUS-Support (v3.5.0+, Futaba / FrSky) | 17 |
| 1. Inverter bauen | 17 |
| 2. Verdrahtung: SBUS-Empfaenger an Mega | 17 |

| | |
|--|-----------|
| 3. SBUS im Menue aktivieren | 17 |
| 4. Fehlersuche | 18 |
| Sound-Files auf der SD-Karte | 19 |
| SD-Karten-Anforderungen | 19 |
| Dateiname-Format + Dateitypen | 19 |
| Track-Nummern-Karte (v3.4.1) | 20 |
| Wie der DFPlayer Dateien adressiert (v3.4.1+) | 20 |
| Lautsprecher-Anschluss | 20 |
| Konfigurationsmenue | 22 |
| Top-Level-Kommandos | 22 |
| Hauptmenue-Struktur | 23 |
| Compile-Feature-Flags (v3.4.1+) | 23 |
| Einen Wert bearbeiten | 24 |
| PID-Tuning-Leitfaden | 25 |
| Was jeder Wert bewirkt | 25 |
| Tuning-Vorgehen | 25 |
| Adaptive PID (optional) | 25 |
| Persoenlichkeit — Idle-Animationen und State-Reaktionen | 27 |
| Idle-Animationen | 27 |
| State-Reactions | 27 |
| Verhalten anpassen | 27 |
| Batterie-Monitoring | 28 |
| Spannungsteiler verdrahten | 28 |
| Default-Schwellen (v3.4) | 28 |
| Divider-Faktor kalibrieren | 28 |
| Feature einschalten | 29 |
| Fehlersuche | 31 |
| Migration von v1.4- / v1.5-Boards | 33 |
| Board-Zuordnung | 33 |
| Was sich softwareseitig aendert | 33 |

Solder-Jumper pruefen 33

Sicherheit **34**

Teil I

Erste Schritte

Einfuehrung — Was ist D-O Control & Power Board?

Das D-O Control & Power Board ist ein Arduino-basiertes Control- und Power-System fuer den selbstbalancierenden D-O-Droiden aus Star Wars: Der Aufstieg Skywalkers. Zwei aktuelle Board-Revisionen werden unterstuetzt: **V1.6** (flexibel, fuer PWM-Empfaenger und optional externen Arduino Nano fuer Sound) und **V1.7** (kompakt, iBus-only, Sound laeuft vollstaendig ueber den Mega).

Hauptfunktionen

- **Selbstbalance** ueber MPU6050 mit PID, kompatibel mit gaengigen Klonen (MPU6500/9250/6886).
- **FlySky iBus** mit 10 Kanaelen, alternativ klassische PWM-Empfaenger.
- **Vier Servos** fuer Mainbar und drei Kopfachsen.
- **DFPlayer Mini** Sound-System mit Persoenlichkeit: Greetings, Stimmungen, Idle-Animationen, Tilt-Warnungen.
- **Cytron MDD10A** Dual-Motortreiber (10 A pro Kanal).
- **Interaktives Serial-CLI** mit EEPROM-Konfiguration (PID-Tuning, Feature-Toggles).
- **Batterie-Monitoring** mit Low-Voltage-Schutz fuer 2x 2S LiPo in Reihe (4S, 14,8 V nominal).

Tipp: Lies das Sicherheitskapitel bevor du zum ersten Mal einschaltest. LiPo-Akkus und Servo-Blockierstroeme muss man ernst nehmen.

Schnelleinstieg

Dieses Kapitel bringt einen frisch aufgebauten D-O in rund 20 Minuten vom verkabelten Zustand auf die Raeder, angenommen Hardware ist montiert und verdrahtet. Wenn unterwegs etwas klemmt: ins Troubleshooting-Kapitel springen.

Raeder bei erstem Einschalten in der Luft. Die Balance-PID kann den Droiden waehrend Kalibrierung und Tuning heftig bewegen.

1. Libraries installieren

Arduino IDE oeffnen, Library Manager, installieren: **IBusBM** (Bolder Flight Systems) und **DFRobotDFPlayerMini** (DFRobot). Wire, Servo, EEPROM, SoftwareSerial kommen mit dem AVR-Core mit.

2. Board waehlen

Werkzeuge → Board → **Arduino Mega oder Mega 2560**. Passenden COM-Port waehlen.

3. Sketch oeffnen

D-O_ibus_v3.4/D_O_printed_droid_rc_ibus_v3.4.ino oeffnen.

4. Upload

Upload-Knopf druecken. "Done uploading" abwarten. Erster Upload dauert etwas, weil AVR-GCC kalt ist.

5. Serial Monitor bei 9600 Baud oeffnen

Es sollte === D-O Universal Controller v3.4.1 === erscheinen, gefolgt von Diagnose-Output.

6. IMU sofort kalibrieren

Innerhalb 3 Sekunden nach Boot D-O aufrecht und perfekt still hinstellen, dann c senden. 5 Sekunden still halten. Offsets werden im EEPROM gespeichert.

7. Power-Cycle

USB / Akku kurz trennen, dann wieder anstecken. D-O sollte aufrecht gestellt von selbst zu balancieren anfangen.

8. Empfaenger pruefen

m fuer Menue, "Show Current Status" auswaehlen, pruefen ob iBus gueltige Kanalwerte zeigt. Wenn Kanale 0 oder unstabil sind: Empfaenger nicht gebunden oder Verdrahtung falsch.

9. Default trimmen wenn noetig

Wenn der Droid nach vorne oder hinten driftet obwohl er physisch aufrecht steht: **Target Angle** im PID-Menue (Option 2) anpassen. Typischer Bereich -1,5 bis +1,5.

10. Testfahrt

Raeder auf den Boden, Abstand halten, Sender scharf. Arcade-Mixing ist Default (rechter Stick vor/zurueck + links/rechts).

Tipp: wenn die Boot-Meldungen zu schnell durchrauschen: `h` senden, dann wird die CLI-Hilfe erneut ausgegeben.

Board- und Sketch-Auswahl

Es gibt zwei aktuelle Board-Revisionen und drei Sketch-Optionen. Bevor du mit dem Verkabeln beginnst, waehle die Kombination, die zu deinem Materiallager passt. Dieses Kapitel zeigt, welcher Sketch fuer welches Board die beste Wahl ist und warum.

Die beiden Boards im Ueberblick

| Aspekt | V1.6 (Standard Control PCB) | V1.7 (Mini iBus Board) |
|-----------------------|---|--------------------------------|
| Formfaktor | Groesser, Platz fuer 10-Kanal-PWM-Empfaenger | Kompakt, nur iBus-Empfaenger |
| Empfaenger-Typen | iBus oder klassisch PWM | Nur iBus |
| Nano-Platz fuer Sound | Ja (optional) | Nein |
| DFPlayer-Anschluss | Mega D7/D8 (Standard) oder Nano D0/D1 per Solder-Jumper | Fest an Mega D7/D8 |
| Am besten fuer | Flexible Aufbauten, Upgrade-Pfade | Saubere Neu-Aufbauten mit iBus |

Verfuegbare Sketche

- **D-O_ibus_v3.4** (aktuell, empfohlen) - Universalcontroller mit DFPlayer am Mega in allen Modi. Unterstuetzt iBus (Mode 1) und klassische PWM-Empfaenger (Mode 0). Watchdog, IMU-Klon-Unterstuetzung, Adaptive PID, Dynamic Lean. Kein externer Nano noetig.
- **D-O_ibus_v2.1** (Legacy) - Vorgaenger von v3. Drei Setup-Modi (PWM only / Hybrid iBus+PWM / Pure iBus), unterstuetzt noch externen Nano fuer Sound ueber setup_type 0. Fuer bestehende Installationen mit Nano gedacht.
- **D_O_Nano_Sketch_v2** (Begleit-Sketch) - Laeuft auf einem externen Arduino Nano und macht nur Sound. Wird zusammen mit D-O_ibus_v2.1 verwendet, wenn der DFPlayer am Nano haengt statt am Mega. Mit v3.4 nicht noetig.

Empfohlene Kombinationen

| Board | Empfaenger | Bester Sketch | Warum |
|----------------------------------|---------------------------|-------------------------------------|--|
| V1.7 | iBus (FS-RX2A Pro, 10 CH) | D-O_ibus_v3.4 (Mode 1) | V1.7 unterstuetzt nur iBus, kein Nano-Slot. v3.4 ist aktueller Stand. |
| V1.6 | iBus (FS-RX2A Pro, 10 CH) | D-O_ibus_v3.4 (Mode 1) | Sauberste Verkabelung, DFPlayer am Mega, alle Features. |
| V1.6 | Klassisch PWM | D-O_ibus_v3.4 (Mode 0) | PWM-Empfaenger bleibt, kein Nano noetig, v3.1 steuert DFPlayer direkt am Mega an. |
| V1.6 | Bestehendes Nano-Setup | D-O_ibus_v2.1 D_O_Nano_Sketch_v2 | + Fuer Aufbauten mit bereits verdrahtetem Nano am Sound. v2.1 setup_type 0 erwartet das. |
| v1.4 / v1.5 (aeltere Revisionen) | wie V1.6 / V1.7 | wie V1.6 / V1.7 | Pin-kompatibel. Gleichen Sketch wie die entsprechende neue Revision waehlen. |

Upgrade-Hinweis fuer v3.4.1

v3.4.1 (Point-Release) bringt drei aus der AIO32-Schwester-Firmware portierte Features: DFPlayer Filename-basierte Adressierung (kein sequenzielles SD-Kopieren mehr noetig), optionale Madgwick-AHRS, und optionale Klassen-PID mit Anti-Windup. Dazu kommen SBUS-Support per neuer Menue-Option `r`, Runtime-Madgwick-Beta-Tuning per `b`, und das tote "Max Acceleration"-Feld ist aus dem EEPROM-Layout entfernt. Der Sketch-Ordner heisst weiter `D-O_ibus_v3.4` — damit vorhandene User-Downloads weiter passen.

v3.4.0 (zuvor) hatte die fehlende Deadband- und Expo-RC-Kurve zurueckgebracht, den Dynamic Lean Angle auf gefiltertes RC umgestellt, das RC-Shaping auf den Stellbereich geclampt und Idle-Animationen bei abgeschaltetem Empfaenger blockiert.

EEPROM-Layout-Aenderung: die Magic-Nummer wurde von `0xD043` (v3.4.0) auf **`0xD044`** (v3.4.1) hochgezogen. Beim **ersten Boot nach dem Upgrade von irgendeiner aelteren Version** erkennt der Sketch das alte Magic und setzt alle Einstellungen auf Defaults zurueck. IMU-Kalibrierung mit `c` neu laufen und Einstellungen ueber das Menue (`m`) neu setzen.

Sketch-Upload

Der Arduino-Sketch ist eine standard `.ino`-Datei — kein Build-System, keine externen Abhaengigkeiten ausser den unten gelisteten Libraries. Upload funktioniert aus der Arduino IDE oder per Kommandozeile mit `arduino-cli`.

Benoetigte Libraries

| Library | Quelle / Source | Zweck / Purpose |
|-------------------------------------|---|--|
| IBusBM | Bolder Flight Systems (Library Manager) | iBus receiver protocol on Serial1 |
| DFRobotDFPlayerMini | DFRobot (Library Manager) | MP3 playback via SoftwareSerial |
| Wire, Servo, EEPROM, SoftwareSerial | Arduino AVR core (included) | I2C, servo PWM, persistent config, DFPlayer link |
| avr/wdt.h | AVR libc (included) | Watchdog timer |

Arduino IDE Schritt fuer Schritt

- **IBusBM** und **DFRobotDFPlayerMini** aus dem Library Manager installieren.
- `D-O_ibus_v3.4/D_O_printed_droid_rc_ibus_v3.4.ino` oeffnen.
- Werkzeuge → Board → **Arduino Mega oder Mega 2560**.
- Werkzeuge → Port → der Port auf dem der Mega erscheint.
- Upload-Knopf druecken. "Done uploading." in der Statusleiste abwarten.
- Werkzeuge → Serieller Monitor. Baudrate auf **9600**.

Kommandozeilen-Alternative

Mit installiertem `arduino-cli`, konfiguriertem Arduino-AVR-Core und den beiden Libraries:

```
arduino-cli compile --fqbn arduino:avr:mega D-O_ibus_v3.4
arduino-cli upload --fqbn arduino:avr:mega --port COM5 D-O_ibus_v3.4
```

Hinweis: Pin D0 und D1 fuehren die Servo-Signale fuer Mainbar und Head 1. Das sind zugleich die Hardware-UART-Pins (Serial). Der Sketch nutzt `Serial` trotzdem fuer die CLI, was klappt weil der Mega Serial1-3 zusaetzlich hat. Bei seltsamem Servo-Zittern waehrend CLI-Nutzung: USB trennen und mit dediziertem BEC betreiben.

IMU-Kalibrierung

Die Accelerometer- und Gyro-Sensoren im MPU6050 haben kleine Fertigungs-Offsets (typisch wenige Zehntel Grad). Die Balance-PID kann "echt vertikal" nicht von "IMU-Chip leicht schief montiert" unterscheiden — deshalb messen wir die Offsets einmal und ziehen sie von jeder folgenden Messung ab. Ohne Kalibrierung kippt der Droid konstant nach vorne oder hinten um einen festen Betrag.

Kalibrierung durchfuehren

- Mega per USB oder Akku einschalten. Serial Monitor bei 9600 Baud.
- Innerhalb der **ersten 3 Sekunden** nach Boot den Droiden aufrecht und perfekt still hinstellen.
- Das Zeichen `c` senden.
- Der Mega gibt "Calibrating..." aus und sammelt ca. 5 Sekunden Mess-Samples.
- Wenn "Calibration complete." erscheint, sind die Offsets im EEPROM und ueberleben Power-Cycles.
- Alternativ: `m fuer Menue`, Option **8** (IMU Calibration) zu beliebiger Zeit.

Wann neu kalibrieren

- Nach Umbau oder Austausch des MPU6050-Moduls
- Wenn der Droid konstant nach vorne oder hinten driftet trotz sinnvollem Target Angle
- Nach dem Flashen einer Sketch-Version mit neuem EEPROM-Magic (Neuinstallation setzt Offsets zurueck)
- Selten: wenn du vermutest dass die IMU einen harten Stoss hatte und der interne Bias gewandert ist

Tipp: die Kalibrier-Samples werden 5 Sekunden gemittelt. Schon leichtes Zittern der Hand verzieht den Mittelwert. Droid auf stabilen Untergrund stellen, dann **Haende weg**, dann `c` senden.

IMU-Klon-Kompatibilitaet: v2.1 und v3.4 sprechen die MPU6050 per direkter I2C-Register-Ansteuerung an (keine externe Library). Sie scannen 0x68 und 0x69 und identifizieren den Chip ueber WHO_AM_I (MPU6050, MPU6500, MPU9250, MPU6886). Typische AliExpress-GY-521-Module laufen out-of-the-box. v1.1 und das Mega2560-Ur-Sketch sprechen nur 0x68 an und koennen Module mit AD0-High nicht nutzen.

Teil II

Kernfunktionen

Sender-Setup (FlySky)

D-O ist auf das FlySky-Oekosystem ausgelegt: ein 10-Kanal-Sender und ein 10-Kanal-iBus-faehiger Empfaenger. Der kompakte **FS-RX2A Pro** ist die empfohlene Wahl (10 CH, ca. 6 USD). Achtung: der aeltere **FS-iA6B** ist ein 6-Kanal-Empfaenger — er deckt die volle D-O-Kanalbelegung nicht ab (wir brauchen CH7-CH10 fuer Sound), und reicht allein nicht aus.

Wichtig: FlySky-Sender belegen CH3 standardmaessig **keinem** Stick oder Schalter. CH3 muss im Sender-Menue manuell zugewiesen werden, sonst reagiert der Mainbar-Servo nicht. Siehe Schritt 3 unten.

1. Empfaenger-Wahl (iBus-Ausgang spart Kabel)

Fast jeder moderne FlySky-10-Kanal-PWM-Empfaenger (FS-iA10B, FS-iA10, FS-R9B, FS-X6B etc.) hat zusaetzlich zu den Kanal-Pins einen **iBus-Ausgang**. Diese eine iBus-Leitung zum Mega (D19 / RX1) spart die sonst noetigen 10 einzelnen Signalleitungen. Zusaetzlich kannst du dann Setup-Mode 1 (iBus) im Sketch waehlen — sauberste Konfiguration. Der dedizierte FS-RX2A Pro ist kleiner und iBus-only aber funktional gleich.

2. Empfaenger binden

- **Bind-Taste** am Empfaenger gedruickt halten waehrend du ihn mit Strom verbindest.
- Taste loslassen. Die LED geht in Bind-Modus (typisch schnelles Blinken oder konstant blau).
- Am Sender **BIND-Modus** starten (meist System → RX Setup → Bind).
- Wenn der Sender Bind-Erfolg bestaetigt: BIND am Sender wieder verlassen.
- Die Empfaenger-LED sollte jetzt bei Signalempfang konstant leuchten, bei Signalverlust langsam blinken.

3. Kanaluweisung in D-O

| CH | Funktion / Function | FlySky Default Control |
|------|---|--|
| CH1 | Drive 1 (Steering in Arcade, Motor 1 in Tank) | Right Stick L/R |
| CH2 | Drive 2 (Throttle in Arcade, Motor 2 in Tank) | Right Stick U/D |
| CH3 | Mainbar Servo | not assigned — map to VrA or a switch |
| CH4 | Head Servo 1 (Pitch) | Left Stick U/D |
| CH5 | Head Servo 2 (Yaw) | Left Stick L/R |
| CH6 | Head Servo 3 (Roll) | VrB dial |
| CH7 | Sound Mute (2-pos) | SwA |
| CH8 | Sound Mode / Trigger (2-pos) | SwB |
| CH9 | Sound Mood (3-pos, neg/mid/pos) | SwC |
| CH10 | Sound Squeak (2-pos) | free — map to any unused control |

4. CH3-Luecke schliessen

An FlySky-Sendern (FS-i6, FS-i6X, FS-i6S) ist CH3 ab Werk unbelegt. Damit der Mainbar-Servo reagiert: **Functions** → **Aux. Channels** oeffnen und CH3 auf **VrA** legen, oder alternativ auf einen 2- oder 3-Positionen-Schalter. Nach Speichern im Monitor (Functions → Display) pruefen, dass CH3 jetzt auf die Bedienung reagiert.

5. Failsafe-Verhalten

v3.4 haelt die Drive-Kanaele auf Mitte wenn das iBus-Signal laenger als 500 ms ausfaellt, und balanciert weiter. Idle-Animationen werden bei Signalverlust unterdruickt. v2.1 loest stattdessen Emergency-Stop aus. In beiden Faellen: Sender-Failsafe auf Neutral-Sticks und Neutral-Schalter stellen, nicht auf Zufallswerte.

Tipp: wer Tank-Style fahren will (ein Stick pro Motor) oeffnet das Konfigurationsmenue, geht auf Driving Dynamics, setzt **Mixing Mode** auf 0. Arcade-Mode (1) ist Default und passt zur standard FlySky-Stick-Belegung.

SBUS-Support (v3.5.0+, Futaba / FrSky)

Ab v3.5.0 unterstützt der Mega-Sketch **SBUS** als Alternative zu FlySky iBus. SBUS ist das serielle RC-Protokoll von Futaba- und FrSky-Empfängern. Die Kanalbelegung auf D-O ist identisch zu iBus (CH1 = Drive 1, CH2 = Drive 2 usw.), sodass der Droid bei korrekter Verdrahtung unabhängig vom Protokoll gleich reagiert.

Hardware-Voraussetzung: die ATmega2560-UART kann ihre RX-Leitung nicht in Hardware invertieren. SBUS nutzt ein invertiertes Signal, deshalb ist ein **externer Inverter** zwingend zwischen SBUS-Ausgang des Empfängers und Mega-Serial1-RX (D19) erforderlich. Ohne ihn sieht der Mega keine Frames (packets = 0 in Menue-Option 9).

1. Inverter bauen

Zwei erprobte Varianten auf einem Stück Lochraster zwischen Empfänger und Mega:

- **NPN-Transistor-Variante:** 1 × BC547 (oder 2N3904), 1 × 10 kΩ (Kollektor-Pull-up nach +5V), 1 × 10 kΩ (Basis-Widerstand vom SBUS-Eingang). Kollektor = invertierter Ausgang zu Mega D19. Emitter nach GND. Klein, ~0,20 EUR Material.
- **74HC14-Variante:** Single-Gate-Schmitt-Trigger-Inverter-IC. Pin 1 = SBUS IN, Pin 2 = invertierter Ausgang zu Mega D19, Pin 7 = GND, Pin 14 = +5V. Schnellere Flanken, aber 6 unbenutzte Gates — nur lohnenswert wenn ohnehin ein 74HC14 da ist.

2. Verdrahtung: SBUS-Empfänger an Mega

- Empfänger-Versorgung (rot): an +5V-BEC.
- Empfänger-Masse (schwarz/braun): an gemeinsames GND.
- Empfänger-SBUS-Ausgang (weiss/gelb): in den Inverter-Eingang.
- Inverter-Ausgang: an Mega-Pin D19 (Serial1-RX).
- Inverter-Versorgung (+5V, GND): von der Mega-5V/GND-Schiene (kleiner Strom, kein BEC nötig).

3. SBUS im Menue aktivieren

- Arduino-IDE-Serial-Monitor bei 9600 Baud öffnen und `m` senden um das Konfig-Menue zu öffnen.
- `r` drücken (RC Protocol).
- `1` eingeben fuer SBUS. Der Sketch re-initialisiert Serial1 sofort auf 100000 Baud / 8E2.
- `s` drücken (Save and Exit) um die Wahl in EEPROM zu speichern.
- Prüfen: `m` erneut, dann Option 9 (Show Status). Oberste Zeile muss "RC Protocol: SBUS (100000 8E2, needs HW inverter)" zeigen. RC-Drive-Werte muessen innerhalb einer Sekunde auf Stick-Bewegung reagieren.
- Rollback: Menue → `r` → `0` → `s`.

4. Fehlersuche

| Symptom | Wahrscheinliche Ursache |
|-----------------------------------|--|
| Keine Reaktion, packets = 0 | Inverter falsch verdrahtet — SBUS IN → Inverter → D19 pruefen. Durchgangstest mit Multimeter. |
| Zufallswerte auf den Kanalen | Inverter fehlt (Droid sieht invertiertes Signal) oder falsche Baudrate — Option 9 muss 100000 8E2 zeigen. |
| Funktioniert kurz und stoppt | SBUS-Failsafe-Bit aktiv. Empfaenger-Bind, Sender-Strom, Antennenlage pruefen. |
| Funktioniert aber Kanale verkehrt | Inverter und Empfaenger invertieren beide (Doppel-Invertierung). Nur einen Inverter zwischen RX und Mega nutzen. |

Tipp: Beide Empfaengertypen kannst du in der Kiste behalten und durch Umstecken wechseln (Inverter-Pfad bei SBUS beruecksichtigen). Dem Mega ist es egal ob der aktuelle Sender FlySky oder FrSky ist, solange die Menue-Einstellung zum verdrahteten Empfaenger passt.

Sound-Files auf der SD-Karte

Der DFPlayer Mini spielt MP3-Dateien von einer MicroSD-Karte. Die Track-Nummerierung, der Ordnername und das Datei-Namensformat sind strikt — der DFPlayer liest Dateisysteme nicht flexibel. Regeln unten einhalten oder es kommt kein Ton.

SD-Karten-Anforderungen

- MicroSD-Karte, bis 32 GB.
- Formatiert als **FAT16** oder **FAT32**. Kein exFAT, kein NTFS.
- Im Root einen Ordner `/mp3/` anlegen.
- Alle Sound-Dateien in diesen Ordner, keine Unterordner.

Dateiname-Format + Dateitypen

Der DFPlayer identifiziert einen Track anhand der ersten **vier Ziffern** des Dateinamens. Alles nach dem Praefix ist frei — du kannst die Dateien so benennen, dass du tatsaechlich siehst, was drin ist. **MP3** und **WAV** werden unterstuetzt.

- Alle diese funktionieren und sind **Track 1**:
 - `0001.mp3`
 - `0001_startup.mp3`
 - `0001_battery_charged.wav`

Fuehrende Nullen sind Pflicht: `1.mp3` funktioniert **nicht**; es muss `0001.mp3` sein (oder irgendein Dateiname der mit `0001` beginnt).

Track-Nummern-Karte (v3.4.1)

| Track | Funktion / Function |
|-----------|---|
| 0001 | Startup sound — "battery charged" |
| 0002 | Default — "I am D-O" |
| 0003–0005 | Greetings (played on CH8 high, random pick) |
| 0006–0009 | Negative (played on CH9 low, random pick) |
| 0010–0014 | Positive (played on CH9 high, random pick) |
| 0015–0020 | Squeaky wheel (played on CH10 trigger, random pick) |
| 0021 | Tilt warning (auto, when angle > 15°) |
| 0022 | Recovery / relief (auto, after a tilt event) |
| 0023 | Low-battery warning (auto, below warning threshold) |
| 0024–0030 | Idle sounds (auto, when droid sits still) |
| 0031 | System ready (after boot) |
| 0032 | Signal lost (when receiver times out) |

Wie der DFPlayer Dateien adressiert (v3.4.1+)

Ab Sketch v3.4.1 (Flag `USE_PLAYMP3FOLDER = 1`, Default ON) fragt die Firmware den DFPlayer nach dem **Filename-Praefix**, nicht nach dem Dateisystem-Index. Du kannst Dateien in beliebiger Reihenfolge kopieren, eine einzelne Datei ersetzen oder neue Tracks dazulegen — kein Neuformatieren und Neukopieren der ganzen Karte noetig. Aeltere Sketche (v3.4.0 und davor) haben den FAT-Index verwendet, was striktes 0001-zuerst-Kopieren erforderte. Wenn du einen aelteren Sketch laeufst oder das Flag auf 0 setzt, gilt die alte Regel.

Lautsprecher-Anschluss

Das Board hat **zwei Lautsprecher-Ausgaenge**, jeweils mit eigenem + und --Anschluss. Beide werden **direkt vom DFPlayer Mini getrieben** — kein externer Verstaerker auf der Platine, und keiner noetig. Die zwei Ausgaenge sind parallel am Mono-Audio-Teil des DFPlayers, sie tragen also dasselbe Signal. Du kannst nutzen:

- **Einen Lautsprecher** an einem der beiden Ausgaenge — der andere bleibt leer.
- **Zwei Lautsprecher parallel** (je einer pro Ausgang), z.B. einen im Kopf und einen im Koerper. Beide spielen dasselbe Mono-Signal.

Empfohlene Daten pro Lautsprecher: **8 Ω, 2-3 W, 28-40 mm Durchmesser**. Zwei 8-Ω-Typen parallel ergeben 4 Ω Gesamtlast — noch im Rahmen, aber die Lautstaerke sollte dann nicht bis Maximum gehen. Bei parallelem Betrieb die +/- Markierung unbedingt beachten, sonst loeschen sich die beiden Lautsprecher akustisch aus.

Tipp: MP3-Bitrate 128-320 kbps. Niedrigere Bitraten gehen auch, aber der Sound wird eh durch einen kleinen Lautsprecher gepresst, 192 kbps ist ein guter Mittelwert. WAV-Dateien sind grösser, vermeiden aber Decoder-Eigenheiten bei sehr kurzen Effekten.

Konfigurationsmenue

Jeder Tuning-Parameter in v3.4.1 ist aus einem interaktiven Serial-Menue erreichbar. Den Arduino-IDE-Serial-Monitor bei 9600 Baud oeffnen und jederzeit `m` senden. Aenderungen bleiben in RAM bis du mit Option `s` speicherst — dann ueberleben sie Power-Cycles via EEPROM.

Top-Level-Kommandos

| Key / Taste | Function / Funktion |
|-------------|--|
| m / M | Open main configuration menu (anytime) |
| c / C | Run IMU calibration — only in the 3-second boot window after reset |
| h / H / ? | Show the top-level help text |

Status-, Reset- und Debug-Funktionen liegen im Menue (Option 9 fuer Status, Option 7 fuer Feature-Toggles usw.). Es gibt keine separaten Hotkeys dafuer.

Hauptmenue-Struktur

| Option | Inhalt / What it configures |
|--------|--|
| 1 | Setup Mode (0=PWM, 1=Serial RC — iBus or SBUS chosen via option r) |
| 2 | PID Configuration — KP, KI, KD, Target Angle, Max Integral |
| 3 | Adaptive PID — KP/KD values for slow / medium / fast speeds |
| 4 | Driving Dynamics — Ramp Rate, Max Lean Angle, Deadband, Expo Factor, Mixing Mode |
| 5 | Battery Settings — Warning V, Critical V, Voltage Divider Factor, Recovery Mode |
| 6 | Sound Settings — Volume (0-30), Sound Intervals, Idle Interval min/max |
| 7 | Feature Toggles — Mainbar Correction, Ramping, Adaptive PID, Dynamic Angle, Idle Actions, State Reactions, Battery Monitor, Servos, Watchdog |
| 8 | Run IMU calibration |
| 9 | Show current status (incl. active RC protocol + FAILSAFE flag on SBUS) |
| r | RC Protocol switch (iBus / SBUS) — runtime re-init, see SBUS chapter |
| b | Madgwick Beta tuning (only active when USE_MADGWICK_AHRS compile-flag=1) |
| m | Motor test & motor-swap / invert configuration |
| i | IMU axis test (live angle readout) |
| s | Save to EEPROM and exit |
| 0 | Exit without saving |

Compile-Feature-Flags (v3.4.1+)

v3.4.1 fuegt drei Compile-Flags am Anfang der .ino-Datei hinzu, portiert aus der AIO32-Schwester-Firmware. Die Defaults erhalten das v3.4.0-Verhalten byte-identisch, sodass ein Upgrade sicher ist; das Umstellen auf 1 aktiviert optionale Verbesserungen, die vor dem Freifahren auf einem Stand getestet werden sollten.

| Flag | Default | What it does |
|--------------------------|---------|--|
| USE_PLAYMP3FOLDER | 1 (ON) | DFPlayer addresses by filename (/mp3/NNNN.mp3) instead of FAT index — SD copy order becomes irrelevant |
| USE_PID_CONTROLLER_CLASS | 0 (OFF) | Replaces inline PID with class-style version: D-term first-order LPF + back-calculation anti-windup |
| USE_MADGWICK_AHRS | 0 (OFF) | Replaces complementary filter with Madgwick 6-DoF quaternion AHRS. Tune beta via menu option b |

Einen Wert bearbeiten

Wenn du ein Untermenue oeffnest, wird jeder Parameter mit aktuellem Wert und Min/Max-Bereich gezeigt. Enter druecken behaelt den aktuellen Wert, oder neue Zahl tippen und Enter. Flieskomma mit Punkt als Trenner (0.8, nicht 0,8).

Tipp: wenn du versehentlich eine Einstellung zerschossen hast und der Droid sich merkwuerdig verhaelt, ist der einfachste Weg: die EEPROM-Magic-Nummer im Sketch kurz aendern (Zeile `uint16_t magic`) und einmal flashen — das erzwingt einen sauberen Default-Reset. Dann zurueckaendern und nochmal flashen. Alternativ: mit Option **0** (Exit without Saving) aus dem Menue, wenn der Fehler noch ungespeichert ist.

PID-Tuning-Leitfaden

Das Balance-Verhalten steuert eine PID-Schleife: sie liest den Neigungswinkel aus der MPU6050 und steuert die Motoren so dass jede Abweichung vom **Target Angle** kompensiert wird. Drei Werte — KP, KI, KD — entscheiden wie aggressiv sie reagiert. Die Defaults (KP=25, KI=0, KD=0.8) sind ein vernuenftiger Startpunkt. Wenn der Droid schwingt, traeege wirkt oder driftet: nach unten scrollen und nachstellen.

Was jeder Wert bewirkt

| Gain | Wirkung / Effect | Symptom wenn zu niedrig / Symptom if too low | Symptom wenn zu hoch / Symptom if too high |
|-----------|--|---|---|
| KP (P) | How hard to push when tilted / Wie stark gegengesteuert wird | Droid falls over slowly / Droid kippt langsam | Rapid oscillation, buzzing motors / schnelles Schwingen |
| KI (I) | Corrects long-term drift / Korrigiert Langzeit-Drift | Droid slowly drifts forward or back / Droid driftet langsam | Wobble builds up over seconds / aufbauendes Schaukeln |
| KD (D) | Damping — resists sudden changes / Daempfung gegen ploetzliche Aenderungen | Overshoot after disturbance / Ueberschiessen nach Stoss | Jittery, nervous motion / zappeliges Verhalten |

Tuning-Vorgehen

- **Mit Defaults starten.** KP=25, KI=0, KD=0.8. Target Angle = -0.3 (kalibrierungsabhaengig).
- **Target Angle zuerst justieren.** Wenn der Droid konstant nach vorn/hinten lehnt waehrend er versucht aufrecht zu balancieren: Target Angle in 0.1-0.3-Schritten anpassen bis der Droid ungefaehr gerade steht.
- **KP fuer Autoritaet einstellen.** Wenn der Droid traeege wirkt und wenig Gegenkraft hat: KP in 2er-Schritten erhoehen (25, 27, 29, ...) bis Schwingen einsetzt, dann 20 % zurueck.
- **KD fuer Daempfung einstellen.** Wenn der Droid bei passendem KP noch schwingt: KD in 0.1-Schritten erhoehen (0.8, 0.9, 1.0, ...) bis das Schwingen verschwindet. Zu viel KD macht ihn zappelig.
- **Kleines KI nur bei Drift.** Wenn der Droid bei passendem KP+KD langsam in eine Richtung kriecht: KI auf 0.1 oder 0.2 stellen. Selten noetig — die meisten Aufbauten laufen mit KI=0.
- **Max Integral pruefen.** Bei 400 lassen. Nur senken wenn KI Aufstauverhalten verursacht.
- **Speichern.** Im Konfigurationsmenue $\$$ senden.

Adaptive PID (optional)

v3.4 kann je nach Bewegungsgeschwindigkeit zwischen drei PID-Werte-Saetzen umschalten: slow, medium, fast. Das hilft wenn ein einzelner Werte-Satz "stehend" und "Vollgas" nicht beide gut abdeckt. Erst die Default-Werte einstellen, nur dann an Adaptive drehen wenn der Droid im Stand gut, bei Fahrt aber instabil wird (oder umgekehrt).

Tipp: Tuning zuerst mit Raedern in der Luft testen. Heftiges Schwingen mit Raedern am Boden kann den Droiden durch den Raum schleudern und Sachen beschaedigen.

Persoenlichkeit — Idle-Animationen und State-Reaktionen

D-O in den Filmen ist ein nervoeser, neugieriger kleiner Droid — er macht Geraeusche, schaut sich um, reagiert auf Kippen. v3.4 bildet das mit zwei Systemen nach: **Idle-Animationen** wenn der Droid still steht, und **State-Reactions** ausgeloeset durch Ereignisse wie Kippen oder schwache Batterie. Beide sind per Default an und in den Feature-Toggles konfigurierbar.

Idle-Animationen

Wenn 3 Sekunden keine RC-Stick-Aktivitaet stattfindet, geht der Droid in Idle-Modus. Alle 5–15 Sekunden (konfigurierbar) wird eine zufaellige Aktion ausgeloeset:

- **70 % Wahrscheinlichkeit:** zufaelliger Idle-Sound (Tracks 0024–0030)
- **30 % Wahrscheinlichkeit:** zufaellige Kopf-Bewegung (nicken, sich umschauchen, Kopf schuettern)

In v3.4 sind Idle-Animationen unterdrueckt wenn das Empfaenger-Signal verloren ist oder seit Boot nie eins kam — kein D-O-Nicken mehr im Raum bei abgeschaltetem Sender. Emergency-Stop blockiert ebenfalls Idle-Servo-Bewegungen.

State-Reactions

| Trigger / Ausloeser | Sound | Details |
|------------------------------------|-------|---|
| Tilt warning (angle > 15°) | 0021 | Plays once per tilt event, 5 s cooldown |
| Recovery (tilt returned to normal) | 0022 | Plays after a tilt event when droid is balanced again |
| Low battery (below warning V) | 0023 | Plays once every 30 s while below threshold |
| Signal lost (receiver timeout) | 0032 | Plays once when iBus drops for > 500 ms |
| System ready (end of boot) | 0031 | Plays once after successful init |

Verhalten anpassen

Konfigurationsmenue mit `m` oeffnen, Option 6 (Sound Settings) und Option 7 (Feature Toggles):

- **idle_interval_min / idle_interval_max** (Option 6) — wie oft Idle-Aktionen feuern
- **min_sound_interval** (Option 6) — Mindestabstand zwischen zwei Sounds gegen Spam
- **Idle Actions**-Toggle (Option 7) — komplett aus wenn ein stiller Droid gewollt ist
- **State Reactions**-Toggle (Option 7) — Tilt/Recovery/Low-Batt-Sounds aus

Batterie-Monitoring

D-O laeuft mit **2 × 2S LiPo in Reihe = 4S insgesamt**: 16,8 V voll geladen, 14,8 V nominal, 12,0 V an der Tiefenentladungs-Grenze von 3,0 V pro Zelle. Der Mega hat ein optionales Spannungs-Monitoring, das vor zu tiefer Spannung warnt — aber das Feature braucht einen einfachen Spannungsteiler, der **ab Werk nicht auf der Platine bestueckt** ist. Den musst du selbst anloeten wenn du dieses Sicherheitsnetz willst.

Deshalb ist Battery Monitor in v3.4 per Default AUS. Ohne Spannungsteiler floatet Pin A15 und liefert Zufallswerte — das Feature wuerde sofort falsch-positive Kritisch-Meldungen ausloesen. Erst **nach** Anloeten des Teilers einschalten.

Spannungsteiler verdrahten

Zwei Widerstaende (10 kΩ und 3,3 kΩ) teilen die Batteriespannung auf unter 5 V runter, so dass der Mega-ADC sie sicher an Pin A15 messen kann:

```
Battery(+) -- [10k] -- A15 -- [3.3k] -- GND

Ratio: (10 + 3.3) / 3.3 = 4.03
At A15 = 5.0 V the battery is ~20.15 V (headroom above 4S max of 16.8 V)
At A15 = 3.67 V the battery is 14.8 V (4S nominal)
At A15 = 3.17 V the battery is 12.8 V (4S critical)
```

Default-Schwellen (v3.4)

| Config Field | Default | Meaning |
|------------------|---------|---|
| battery_warning | 13.6 V | ~3.4 V per cell, long-beep warning, every 30 s |
| battery_critical | 12.8 V | ~3.2 V per cell, stops motors and alarms |
| voltage_divider | 4.03 | Calibration factor for 10k / 3.3k divider |
| battery_monitor | false | Feature toggle — enable only after wiring the divider |
| battery_recovery | false | If true, motors resume after battery voltage climbs back above critical + 0.2 V |

Divider-Faktor kalibrieren

Reale Widerstaende haben 1-5 % Toleranz. Fuer genaue Werte die tatsaechliche Batterie-Spannung mit einem Multimeter messen waehrend der Droid an ist, dann **voltage_divider** im Battery-Menue so justieren dass die Mega-Anzeige uebereinstimmt. Beispiel: Multimeter 15,1 V, Mega zeigt 14,7 V — aktuellen Divider-Faktor mit $15,1 / 14,7 \approx 1,027$ multiplizieren.

Feature einschalten

- 10-k Ω - + 3,3-k Ω -Teiler von Batt+ nach GND anlöten, Mittelabgriff an A15.
- Konfigurationsmenue oeffnen (m) → Feature Toggles (7).
- **Battery Monitor** auf true.
- Optional **Battery Recovery** auf true wenn die Motoren nach kurzem Spannungseinbruch weiterlaufen sollen.
- Mit `s` speichern.
- Pruefen: Show Status sollte jetzt eine plausible Batteriespannung anzeigen.

Tipp: wenn du den Teiler nie einbaust: Battery Monitor aus lassen. Der Droid laeuft auch ohne — du verlierst nur das Safety-Net. Ein guenstiger LiPo-Alarm am Balance-Stecker macht dasselbe extern.

Teil III

Referenz & Anhaenge

Fehlersuche

Haeufige Probleme und die ueblichen Ursachen. Schritte der Reihe nach durchgehen.

Droid macht nichts / keine Serial-Ausgabe

- Falsches Board in der Arduino IDE (muss Mega 2560 sein, nicht Uno)
- Falscher COM-Port
- USB-Kabel nur zum Laden ohne Datenleitungen
- Serial-Monitor Baudrate falsch (muss 9600 sein)

Compile-Fehler zu IBusBM oder DFRobotDFPlayerMini

- Library nicht installiert — Library Manager oeffnen (Sketch → Bibliothek einbinden → Bibliotheken verwalten)
- Falsche Library-Variante — DFRobotDFPlayerMini muss die von DFRobot sein, kein Fork
- Fremde Dateien im Sketch-Ordner — nur die .ino der relevanten Version lassen

Droid balanciert nicht / kippt

- IMU nicht kalibriert — c innerhalb 3 s nach Boot senden, oder Option 8 im Menue
- Target Angle falsch — im PID-Menue in 0,1-Schritten anpassen bis der Droid gerade steht
- Motor falsch verdrahtet — Motor Swap / Motor Invert in den Feature Toggles nutzen
- MPU6050 falsch verdrahtet — SDA an D20, SCL an D21, nicht die analogen SDA/SCL-Pins
- IMU gedreht montiert — IMU Invert aktivieren wenn vorn/hinten verkehrt reagiert

Droid schwingt schnell

- KP zu hoch — im PID-Menue um 2-3 senken
- KD zu niedrig — im PID-Menue um 0,1 erhoehen
- Schmutzige USB-Versorgung — zum Tunen nur auf Akku laufen lassen

Keine RC-Reaktion

- Empfaenger nicht gebunden — Binding-Prozedur im Sender-Kapitel
- iBus-Signal am falschen Pin — muss D19 sein (Serial1 RX)
- Setup Mode falsch — Menue-Option 1, auf 1 (iBus) stellen wenn iBus-Empfaenger vorhanden
- iBus-Baudrate stimmt nicht — zwischen 9600 und 115200 umschalten

Mainbar-Servo reagiert nicht

- CH3 am Sender nicht belegt — FlySky-Default hat keine CH3-Zuweisung, siehe Sender-Setup

Kein Ton

- SD-Karte nicht FAT16/FAT32
- Dateinamen nicht im 0001.mp3-Format (muessen exakt vier Ziffern sein)

- Dateien nicht im /mp3/-Ordner
- DFPlayer-Verdrahtung vertauscht — Mega D7 liest, Mega D8 sendet via 1-k Ω -Widerstand
- Fehlender 1-k Ω -Widerstand auf TX-Leitung beschadigt DFPlayer-RX
- Lautsprecher lose oder falsch gepolt (8 Ω 2 W empfohlen)
- Volume steht auf 0 in den Sound Settings

Vor/Zurueck viel langsamer als Drehen

- Mixing Mode ist Tank aber du faehrst mit Arcade-Sticks — auf Arcade-Mode umstellen (Driving Dynamics, Option 1)

Batteriespannung zeigt Unsinn

- Spannungsteiler nicht angeloetet — A15 floatet ohne 10k- und 3,3k-Widerstaende
- Battery Monitor trotzdem an — ausschalten (Feature Toggles) bis der Teiler verbaut ist
- Divider-Faktor nicht kalibriert — Batterie mit Multimeter messen und voltage_divider anpassen

Droid startet zufaellig neu

- Watchdog feuert weil irgendwo im Loop was haengt — Watchdog testweise aus (Feature Toggles) um zu sehen wo der Hang sichtbar wird
- Brownout in der Stromversorgung — Motoren und Servos brauchen eigenen BEC, nicht die Mega-5-V-Schiene
- I2C-Error-Loop auf der IMU — SDA/SCL-Verdrahtung pruefen, anderes MPU6050-Modul probieren

Idle-Animationen laufen bei abgeschaltetem Sender

- Nur in v3.3.x und frueher — Upgrade auf v3.4, dort werden Idle-Aktionen bei Signalverlust geblockt

Migration von v1.4- / v1.5-Boards

Die aelteren v1.4- (Control PCB) und v1.5- (Mini iBus) Boards sind **pin-kompatibel** mit den aktuellen V1.6 und V1.7 — hardware-seitig muss nichts umverdrahtet werden. Du musst nur den Sketch aktualisieren und einige Verhaltens-Aenderungen kennen.

Board-Zuordnung

| Old board | Current equivalent | Recommended sketch |
|--------------------------|--------------------|---------------------------|
| v1.4 Control PCB (large) | V1.6 Standard PCB | D-O_ibus_v3.4 Mode 0 or 1 |
| v1.5 Mini iBus | V1.7 Mini iBus | D-O_ibus_v3.4 Mode 1 |

Was sich softwareseitig aendert

- **Sketch-Version.** Aeltere Installationen liefen mit v1.1 oder fruehen v2.x-Sketchen. v3.4 ist ein deutlicher Rewrite — Kapitel Board- und Sketch-Auswahl lesen.
- **EEPROM-Layout.** Erster Boot auf v3.4 setzt alle Einstellungen auf Defaults, weil das EEPROM-Magic auf 0xD043 hochgesetzt wurde. IMU kalibrieren (c) und Menue-Einstellungen neu eintragen.
- **Battery-Schwellen.** Waren mit "2 x 2S" beschriftet aber tatsaechlich fuer ein einzelnes 2S-Pack dimensioniert. v3.4 korrigiert das auf das reale 4S-Setup (13,6 V / 12,8 V).
- **Externer Nano entfaellt.** v1.4-Boards liefen typisch mit einem Arduino Nano fuer Sound. v3.4 steuert den DFPlayer direkt am Mega an, unabhaengig vom Mode. Nano abloeten oder unbestromt lassen — er ist nicht mehr Teil des Signalwegs wenn v3.4 geflasht ist.
- **Hybrid-Mode weg.** v2.1 hatte drei Setup-Types inklusive Hybrid (iBus CH1-6 + PWM CH7-10). v3.4 hat nur Mode 0 (alles PWM) und Mode 1 (alles iBus). Die meisten User wollen Mode 1.

Solder-Jumper pruefen

Wenn du von v1.4 mit externem Nano auf v3.4 mit Mega-nativer Soundausgabe wechselst: auf der Platine pruefen dass die Solder-Pads **P7** und **P8** gebroeckt sind — sie verbinden Mega D7/D8 mit dem DFPlayer. Fuer Mode 0 (PWM-Empfaenger) zusaetzlich **CH7**, **CH8**, **CH9**, **CH10** bruecken damit die Sound-Switches vom Empfaenger zum Mega kommen. Fuer Mode 1 (iBus) reichen P7 und P8 — die 10 Kanaele kommen ueber die einzelne iBus-Leitung.

Tipp: bei Unsicherheit ob ein Solder-Jumper gebroeckt ist: Platine unter helles Licht halten — gebroeckte Pads zeigen einen sichtbaren Loetkleks der beide Haelften verbindet. Unverloetete Pads haben sichtbare Luecke und Kupfertrenner.

Sicherheit

Der vollstaendige Haftungsausschluss steht auf Seite 2. Dieses Kapitel listet die praktischen Sicherheitsregeln fuer den Betrieb von D-O.

| Regel | Details |
|-----------------------|--|
| LiPo-Handhabung | LiPo-faehiges Ladegeraet. Nicht unbeaufsichtigt laden. Aufgeblaehete oder beschaedigte Packs ersetzen. |
| Polung | Jede Stromverbindung vor dem Einschalten verifizieren. Verkehrte Polung toetet den Mega sofort. |
| Spannung | 2x 2S LiPo in Reihe = 4S gesamt. 16,8 V voll, 14,8 V nominal, 12,0 V leer. Nicht unter 3,0 V pro Zelle entladen. |
| Servo-Stromversorgung | Servos ueber BEC (5-6 V, 3-5 A) speisen, NICHT ueber die 5-V-Schiene des Mega. |
| Motor-Stroeme | Cytron MDD10A vertraegt 10 A Dauerstrom pro Kanal. Bei Blockade auf Erwaermung achten. |
| Erstes Einschalten | Raeder in der Luft. PID kann den Droiden waehrend Kalibrierung / Tuning heftig bewegen. |
| Kalibrier-Haltung | D-O aufrecht und perfekt still halten, 5 Sekunden lang, waehrend IMU-Kalibrierung. |
| Umgebung | Bereich frei halten. Ein umkippende Droid kann auf Haende, Haustiere oder Kaffeetassen fallen. |