



Ⓧ Bedienungsanleitung

Hexapod Roboter-Board

Best.-Nr. 1589388

CE

	Seite
1. Einführung	3
2. Symbol-Erklärung	3
3. Bestimmungsgemäße Verwendung	4
4. Produktbeschreibung	4
5. Lieferumfang	5
6. Sicherheitshinweise	5
7. Benötigtes Zubehör zu Komplettierung des Roboters	7
8. Übersicht der Anschlüsse und Komponenten	8
9. Schematische Systemübersicht (Blockschaltbild)	12
10. Inbetriebnahme	13
11. Stromversorgung	14
12. Installation der Software und Firmware	15
13. Übertragen der Firmware	16
a) Treiber installieren	16
b) Arduino IDE installieren	16
c) Arduino IDE einrichten	17
14. RC-Servos für den Einbau vorbereiten	21
15. Hexapod-Mechanik und Grundkonfiguration	23
16. RC-Servos anschließen	26
17. Gamepad anschließen	28
18. Firmware-Konfiguration	32
19. Firmware übertragen	40
20. Steuern mit dem Gamepad	41
21. Beine kalibrieren	43
22. Verwenden von User-Boards	47
23. Demo-Programme	51
24. FAQ	55
25. Pinzuweisung der User-Boards	57
26. Reinigung	59
27. Entsorgung	59
28. Technische Daten	60

1. Einführung

Sehr geehrte Kundin, sehr geehrter Kunde,
wir bedanken uns für den Kauf dieses Produkts.

Dieses Produkt erfüllt die gesetzlichen, nationalen und europäischen Anforderungen.

Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, müssen Sie als Anwender diese Bedienungsanleitung beachten!



Diese Bedienungsanleitung gehört zu diesem Produkt. Sie enthält wichtige Hinweise zur Inbetriebnahme und Handhabung. Achten Sie hierauf, auch wenn Sie dieses Produkt an Dritte weitergeben.

Heben Sie deshalb diese Bedienungsanleitung zum Nachlesen auf!

Alle enthaltenen Firmennamen und Produktbezeichnungen sind Warenzeichen der jeweiligen Inhaber. Alle Rechte vorbehalten.

Bei technischen Fragen wenden Sie sich bitte an:

Deutschland: www.conrad.de/kontakt

Österreich: www.conrad.at
www.business.conrad.at

Schweiz: www.conrad.ch
www.biz-conrad.ch

2. Symbol-Erklärung



Das Symbol mit dem Ausrufezeichen im Dreieck weist auf wichtige Hinweise in dieser Bedienungsanleitung hin, die unbedingt zu beachten sind.



Das Pfeil-Symbol ist zu finden, wenn Ihnen besondere Tipps und Hinweise zur Bedienung gegeben werden sollen.

3. Bestimmungsgemäße Verwendung

Das Hexapod Roboter-Board ist ausschließlich für Ausbildung, Forschung und den privaten Einsatz im Hobby- bzw. Modellbaubereich und für die damit verbundenen Betriebszeiten ausgelegt.

Das Produkt ist kein Spielzeug, es ist nicht für Kinder unter 14 Jahren geeignet.

Das Produkt richtet sich an fortgeschrittene Anwender, die bereits Erfahrung mit Arduino und der Programmiersprache C/C++ und auch in der Elektronik sowie beim Aufbau von mechanischen Bausätzen besitzen.

Falls Sie das Produkt für andere Zwecke verwenden, als zuvor beschrieben, kann das Produkt beschädigt werden. Außerdem kann eine unsachgemäße Verwendung Gefahren wie z.B. Kurzschluss, Brand, Stromschlag etc. hervorrufen. Lesen Sie sich die Bedienungsanleitung und dieses Informationsblatt genau durch und bewahren Sie diese auf. Reichen Sie das Produkt nur zusammen mit der Dokumentation an dritte Personen weiter.

Dieses Produkt erfüllt die gesetzlichen, nationalen und europäischen Anforderungen. Alle enthaltenen Firmennamen und Produktbezeichnungen sind Warenzeichen der jeweiligen Inhaber. Alle Rechte vorbehalten.



Das Produkt darf nicht feucht oder nass werden. Die Elektronik ist für einen Betrieb bei einer Umgebungstemperatur zwischen 0 °C bis 40 °C ausgelegt.

Das Produkt ist kein Spielzeug, es ist nicht für Kinder unter 14 Jahren geeignet.

Beachten Sie alle Sicherheitshinweise dieser Bedienungsanleitung. Diese enthalten wichtige Informationen zum Umgang mit dem Produkt. Sie allein sind für den gefahrlosen Betrieb des C-Control Hexapod Roboter-Boards verantwortlich!

4. Produktbeschreibung

Das Hexapod Roboter-Board enthält alle wichtigen Komponenten, die zum Betrieb eines selbstgebaute Hexapod-Roboters mit Modellbau-Servos (RC-Servos) erforderlich sind.

Die Besonderheit dabei ist die Möglichkeit zur zusätzlichen Nutzung verschiedener Mikrocontroller-Boards wie Arduino UNO, SBC (Single Board Computer) wie Raspberry Pi oder auch NodeMCU. Auf diesen „User-Boards“ können eigene Programme zum Steuern des Roboters erstellt werden. Die Kommunikation zwischen User-Board und dem „Locomotion Controller“ erfolgt über ein serielles Protokoll, das die einfache Steuerung des Roboters erlaubt, ohne sich mit den komplexen Laufalgorithmen befassen zu müssen.

Das Hexapod Roboter-Board ist so ausgelegt, dass es individuelle Modifikationen und Erweiterungen erlaubt. Zudem enthält es noch einige zusätzliche Komponenten wie Taster, MicroSD-Kartenslot, Lautsprecher mit Verstärkerschaltung, I2C-Anschlüsse, IR-Empfänger, USB-Programmierschnittstelle für den Locomotion-Controller, ISP-Anschlüsse (SPI) und weitere Ein-/Ausgänge.

Die Programmierung des Hexapod Roboter-Boards erfolgt über die Open Source Arduino Plattform. Der Locomotion-Controller ist mit dem Arduino MEGA2560-Board kompatibel. Bei Bedarf kann die Firmware des Locomotion-Controllers individuell von Ihnen an den Roboter angepasst werden. Alle Beispielprogramme finden Sie unter www.conrad.com auf der jeweiligen Internetseite zum Produkt.

5. Lieferumfang

- 1x C-Control Hexapod Roboter-Board
- 2x Steckbrücke
- 1x Akkustecker XT30

Aktuelle Bedienungsanleitungen

Laden Sie aktuelle Bedienungsanleitungen über den Link www.conrad.com/downloads herunter oder scannen Sie den abgebildeten QR-Code. Befolgen Sie die Anweisungen auf der Webseite.



6. Sicherheitshinweise



Lesen Sie sich die Bedienungsanleitung aufmerksam durch und beachten Sie insbesondere die Sicherheitshinweise. Falls Sie die Sicherheitshinweise und die Angaben zur sachgemäßen Handhabung in dieser Bedienungsanleitung nicht befolgen, übernehmen wir für dadurch resultierende Personen-/Sachschäden keine Haftung. Außerdem erlischt in solchen Fällen die Gewährleistung/Garantie.

Sehr geehrte Kundin, sehr geehrter Kunde,

diese Sicherheitshinweise dienen nicht nur dem Schutz des Produkts, sondern auch Ihrer eigenen Sicherheit und der Sicherheit anderer Personen. Lesen Sie sich deshalb dieses Kapitel sehr aufmerksam durch, bevor Sie das Produkt in Betrieb nehmen!

- Das Produkt richtet sich an fortgeschrittene Anwender, die bereits Erfahrung mit Arduino und der Programmiersprache C/C++, in der Elektronik und beim Aufbau von mechanischen Bausätzen besitzen. Sollten Sie nicht über ausreichend Erfahrung verfügen, so wenden Sie sich bitte an einen erfahrenen Entwickler, einen umliegenden Maker-Club oder an unseren Support.
- Aus Sicherheits- und Zulassungsgründen ist das eigenmächtige Umbauen außerhalb der beschriebenen Möglichkeiten und/oder das Verändern des Produkts nicht gestattet.
- Das Produkt ist kein Spielzeug. Halten Sie es von Kindern und Haustieren fern.
- Lassen Sie das Verpackungsmaterial nicht achtlos liegen. Dieses könnte für Kinder zu einem gefährlichen Spielzeug werden.
- Schützen Sie das Produkt vor extremen Temperaturen, direktem Sonnenlicht, starken Erschütterungen, hoher Feuchtigkeit, Nässe, brennbaren Gasen, Dämpfen und Lösungsmitteln.
- Setzen Sie das Produkt keiner mechanischen Beanspruchung aus.
- Achten Sie beim Anschluss der RC-Servos und anderer Komponenten sowie deren Anschlussleitungen auf einen sicheren Kontakt. Lose oder wackelige Steckverbinder können Störungen oder Schäden hervorrufen.
- Sollten Lötarbeiten erforderlich sein, so achten Sie unbedingt darauf, dass beim Löten keine Kurzschlüsse entstehen.



- Achten Sie beim Einbau der Platine darauf, dass Bauteile oder Lötkontakte nicht mit metallischen Teilen in Berührung kommen und somit Kurzschlüsse entstehen. Dabei wird das Produkt beschädigt, Verlust von Gewährleistung/Garantie!
- Wenn kein sicherer Betrieb mehr möglich ist, nehmen Sie das Produkt außer Betrieb und schützen Sie es vor unbeabsichtigter Verwendung. Der sichere Betrieb ist nicht mehr gewährleistet, wenn das Produkt:
 - sichtbare Schäden aufweist,
 - nicht mehr ordnungsgemäß funktioniert,
 - über einen längeren Zeitraum unter ungünstigen Umgebungsbedingungen gelagert wurde oder
 - erheblichen Transportbelastungen ausgesetzt wurde.
- Gehen Sie vorsichtig mit dem Produkt um. Durch Stöße, Schläge oder dem Fall aus bereits geringer Höhe wird es beschädigt.
- Wenden Sie sich an eine Fachkraft, wenn Sie Zweifel über die Arbeitsweise, die Sicherheit oder den Anschluss des Produktes haben.
- Lassen Sie Wartungs-, Anpassungs- und Reparaturarbeiten ausschließlich von einem Fachmann bzw. einer Fachwerkstatt durchführen.
- Sollten Sie noch Fragen haben, die in dieser Bedienungsanleitung nicht beantwortet werden, wenden Sie sich an unseren technischen Kundendienst oder an andere Fachleute.

7. Benötigtes Zubehör zu Komplettierung des Roboters

Um einen lauffähigen Roboter zu erhalten, benötigen Sie zusätzlich zum Hexapod Roboter-Board noch folgende Zusatzkomponenten.

- 1x Hexapod-Mechanik (eigenes Design z.B. 3D-Druck oder einen fertigen Mechaniksetz, z.B. Conrad Best.-Nr. 1618958)
- 18x RC-Servos passend zur verwendeten Hexapod Mechanik. z.B. Conrad Best.-Nr. 1365926
- 1x PS2-kompatibles Gamepad mit Funkempfänger, z.B. Conrad Best.-Nr. 161300
- 1x NiMH-Akkupack mit 5 Zellen (Nennspannung 6 V/DC), Kapazität min. 2000 mAh
- 1x Ladegerät für den verwendeten NiMH-Akku
- 1x Arduino UNO (Conrad Best.-Nr. 191789), NodeMCU (Conrad Best.-Nr. 161301) oder Raspberry Pi (Conrad Best.-Nr. 1419716) User-Board (für die ersten Schritte empfiehlt sich die Verwendung eines Arduino UNO User)

8. Übersicht der Anschlüsse und Komponenten

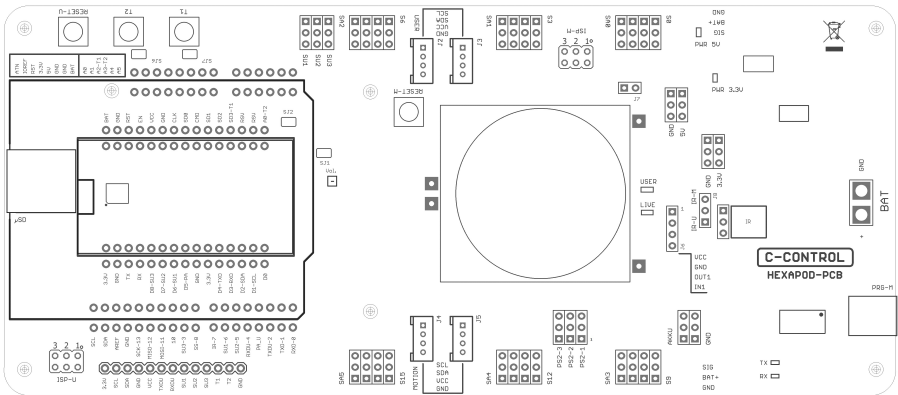


Bild 1

Legen Sie die das Hexapod Roboter-Board wie auf der Abbildung gezeigt vor sich hin, um sich einen Überblick über die Anschlussmöglichkeiten zu verschaffen.

➔ Auf www.conrad.com auf der Produktseite finden Sie zudem den Schaltplan zum Board.

Das Hexpod Roboter-Board verfügt über folgende Anschlüsse und Komponenten:

- BAT:** XT30-Stecker zum Anschluss des Akkus, der zur Stromversorgung des Boards dient. Betriebsspannung 4,5 bis 10 V/DC (+ = Pluspol; GND = Minuspol). Je nach verwendeten RC-Servos sollte als Stromversorgung ein NiMH-Akku mit 5 Zellen (Nennspannung 6 V/DC) und einer Kapazität von mindestens 2000 mAh zum Einsatz kommen.
- PRG-M:** USB-Anschluss zur Programmierung des Locomotion-Controllers.
- TX-LED:** Signalisiert bei der Übertragung eines Programms in den Locomotion-Controller die Sendeleitung (schnelles Blinken bei der Übertragung).
- RX-LED:** Signalisiert bei der Übertragung eines Programms in den Locomotion-Controller die Empfangsleitung (schnelles Blinken bei der Übertragung).
- AKKU und GND:** An diesen Pins kann die Akkuspannung abgegriffen werden (AKKU = Pluspol, GND = Minuspol). Die Pins sind nicht gegen Kurzschluss geschützt! Arbeiten Sie hier vorsichtig und verursachen Sie keinen Kurzschluss! Dies kann zur Zerstörung der Pins und des Hexapod Roboter-Boards führen! Dieser Anschluss steht für eigene Erweiterungen und Experimente zur Verfügung.

- J6: Hier stehen an den Pins VCC und GND stabilisierte 5 V/DC/1000 mA zur Verfügung. Dieser Anschluss dient für eigene Erweiterungen und Experimente (VCC = Pluspol, GND = Minuspol).
- OUT1 ist ein digitaler Ausgang des Locomotion-Controllers. Er kann über ein Kommando vom User-Board auf HIGH (+5 V/DC) oder LOW (0 V/DC) geschaltet werden. Die maximale Strombelastbarkeit beträgt 20 mA. Dieser Anschluss steht für eigene Erweiterungen und Experimente zur Verfügung.
- IN1 ist ein digitaler Eingang des Locomotion-Controllers. Er kann über ein Kommando vom User-Board einen digitalen Zustand lesen. Eine logische 1 wird zwischen 3,5 und 5 V/DC erkannt; eine logische 0 bei einer Spannung kleiner als 3,3 V/DC. Die maximale Eingangsspannung von 5 V/DC darf nicht überschritten werden! Dieser Anschluss steht für eigene Erweiterungen und Experimente zur Verfügung.
- IR: Der Infrarotempfänger „IR“ kann wahlweise mit dem Jumper „J8“ mit dem Locomotion-Controller (Jumperstellung IR-M) oder mit dem User-Board (Jumperstellung IR-U) verbunden werden. Der Empfänger ermöglicht es, ein 38 kHz Infrarotsignal z.B. einer Universalfernbedienung oder eines selbstgebaute IR-Senders zu empfangen.
- LIVE-LED: Die blaue „LIVE-LED“ signalisiert den Zustand des Locomotion-Controllers. Ist der Locomotion-Controller aktiv, blinkt diese LED in unterschiedlichen Zuständen.
- USER-LED: Die rote „USER-LED“ signalisiert, dass das PS2-kompatible Gamepad aktiviert ist und die Steuerkommandos vom User-Board nun ignoriert werden.
- 3.3V und GND: An diesen Pins stehen stabilisierte 3,3 V/DC/400 mA zur Verfügung. Dieser Anschluss steht für eigene Erweiterungen und Experimente zur Verfügung (3.3V = Pluspol, GND = Minuspol).
- 5V und GND: An diesen Pins stehen stabilisierte 5 V/DC/1000 mA zur Verfügung. Dieser Anschluss steht für eigene Erweiterungen und Experimente zur Verfügung (5V = Pluspol, GND = Minuspol).
- 3.3V PWR-LED: Die rote LED signalisiert, dass die 3.3 V-Spannungs-/Stromversorgung arbeitet. Sie leuchtet, sobald am BAT-Anschluss eine Stromversorgung angeschlossen ist.
- 5V PWR-LED: Die rote LED signalisiert, dass die 5 V-Spannungs-/Stromversorgung arbeitet. Sie leuchtet, sobald am BAT-Anschluss eine Stromversorgung angeschlossen ist.
- J7: Der Jumper „J7“ deaktiviert in der Locomotion-Controller-Firmware die Auswertung des PS2-kompatiblen Gamepads (der Roboter lässt sich dann nicht mehr über das Gamepad steuern). Die Gamepad-Steuerbefehle können dann nur noch über das User-Board ausgelesen werden!
- ISP-M: ISP-Anschluss („In-System-Programmierung“) des Locomotion-Controllers. Über diesen Anschluss kann der Locomotion-Controller über ein ISP-Programmiergerät programmiert werden. Es ist zudem möglich, diesen Anschluss zum Einbinden von eigenen Komponenten mit SPI-Schnittstelle zu verwenden.
- PS2-1 bis PS2-3: An diesen Pins kann ein PS2-kompatibles Gamepad, welches als Zubehör erhältlich ist, angeschlossen werden. Über diesen Controller kann der Roboter manuell gesteuert werden, ähnlich einem ferngesteuerten Auto.
- J2 und J3: Die beiden Buchsen sind mit dem I2C-BUS-Anschluss des Locomotion-Controllers verbunden und können für eigene Erweiterungen genutzt werden. Die Anschlüsse sind mit den „SEED-GROVE“ Komponenten kompatibel.
- J4 und J5: Die beiden Buchsen sind mit dem I2C-BUS-Anschluss des User-Boards verbunden und können für eigene Erweiterungen genutzt werden. Die Anschlüsse sind mit den „SEED-GROVE“ Komponenten kompatibel.
- RESET-M: Der Taster löst beim Drücken einen Hardware-Reset des Locomotion-Controllers aus.

- S0 bis S17: An diesen Anschlüssen werden die „Bein-Servos“ des Hexapods angeschlossen. Die Stiftleisten sind immer in Blöcken angeordnet, die dem jeweiligen Bein zugeordnet sind. Von der jeweiligen Beschriftung (S...) aus werden die Anschlüsse für „Coxa - Hüfte“, „Femur - Oberschenkel“ und „Tibia - Schienbein“ gezählt. Die Pins sind mit den gängigsten RC-Servos kompatibel. Achten Sie hier auf richtige Polung. Die Polung ist auf der Platine mit „SIG“ für Signalleitung, „BAT+“ für den Pluspol und „GND“ für den Minuspol beschriftet.
- SA0 bis SA5: Die analogen Eingänge SA0 bis SA5 können für eigene Erweiterungen in der Locomotion-Firmware verwendet werden. In der „Motion-Firmware“ können diese auch über eine Funktion vom User-Board abgefragt werden.
- SJ1: Der PCB-Jumper „SJ1“ verbindet den Audioverstärker mit dem Locomotion-Controller. In den meisten Praxisfällen bleibt diese Verbindung bestehen. Sie können den Jumper, falls nötig, mit einem kleinen Tapetenmesser vorsichtig durchtrennen. Mit einem Lötkolben und etwas Lötzinn können Sie die Verbindung wieder herstellen.
- Lautsprecher: Der Lautsprecher dient zur Ausgabe der Signaltöne und ist mit dem Audioverstärker verbunden. Die Signaltöne können vom Locomotion-Controller und auch vom User-Board generiert werden.
- Vol.: Der Trimmer „Vol.“ dient zum Einstellen der Lautstärke der Signaltöne des Locomotion-Controllers und des User-Boards. Drehen Sie den Trimmer in Richtung der Taster, so wird die Lautstärke erhöht; in der entgegengesetzten Richtung wird sie leiser. Zum Einstellen des kleinen Trimmers hat sich in der Praxis ein kleiner Uhrmacher-Schraubendreher herausgestellt. Seien Sie beim Einstellen vorsichtig, um den Trimmer nicht zu beschädigen! Grundeinstellung ist die Mittelstellung des Trimmers.
- T1 und T2: Die Taster „T1“ und „T2“ können mit dem User-Board genutzt werden. Sehen Sie sich hierzu die Software-Beispiele an.
- SJ6 und SJ7: Die PCB-Jumper „SJ6“ und „SJ7“ verbindet die Ausgänge der Taster „T1“ und „T2“ mit einem 22 kOhm Pullup-Widerstand, so dass im User-Board Programm keine Pullup-Widerstände aktiviert werden müssen. In den meisten Praxisfällen bleibt diese Verbindung bestehen. Sie können die Jumper, falls nötig, mit einem kleinen Tapetenmesser vorsichtig durchtrennen. Mit einem Lötkolben und etwas Lötzinn können Sie die Verbindung wieder herstellen.
- RESET-U: Der Taster „RESET-U“ führt einen Reset des User-Boards durch.
- SJ2: Der PCB-Jumper „SJ2“ verbindet den Taster „T2“ mit dem NodeMCU User-Board. Der verwendete Pin der NodeMCU ist der analoge Eingang und kann bei anderweitiger Verwendung vom Taster „T2“ über diesen Jumper abgetrennt werden. Im Normalfall bleibt diese Verbindung bestehen. Sie können den Jumper, falls nötig, mit einem kleinen Tapetenmesser vorsichtig durchtrennen. Mit einem Lötkolben und etwas Lötzinn können Sie die Verbindung wieder herstellen.
- SU1 bis SU3: Die Anschlüsse SU1, SU2 und SU3 sind mit den User-Board-Steckplätzen verbunden und können für eigene Anwendungen genutzt werden. Sie besitzen die gleiche Polung wie die der RC-Servo-Anschlüsse der Bein-Servos. Sehen Sie sich hierzu die Software-Beispiele an.
- μSD: Der Kartensteckplatz für eine MicroSD-Karte ist mit dem Arduino-UNO kompatiblen User-Board-Steckplatz verbunden und kann für eigene Anwendungen genutzt werden. Sehen Sie sich hierzu die Software-Beispiele an.
- ISP-U: ISP-Anschluss („In-System-Programmierung“) des „Arduino UNO“-kompatiblen User-Board-Steckplatzes. Über diesen Anschluss kann das User-Board über ein ISP-Programmiergerät programmiert werden. Es ist zudem möglich, diesen Anschluss zum Einbinden von eigenen Komponenten mit SPI-Schnittstelle zu verwenden.

User-Boards: Arduino-UNO:

Die beiden äußeren längeren Stiftheisten dienen zur Aufnahme eines „Arduino-UNO“-kompatiblen User-Boards. Alle Pins sind an Buchsenleisten daneben herausgeführt. Dies dient dazu, dass Sie einfach auf die Pins des Boards zugreifen können. Die Pinbelegung ist auf der Platine neben den Stiftheisten aufgedruckt und entspricht der des verwendeten Boards. Sehen Sie sich dazu die Software-Beispiele an.

NodeMCU:

Das beliebte WiFi-Board mit der Bezeichnung „NodeMCU“ und dem verwendeten ESP8266-WiFi-Chip kann auf den inneren Steckplatz als User-Board verwendet werden. Auch hier sind alle Pins auf den daneben liegenden Buchsenleisten herausgeführt, was das Experimentieren erleichtert. Sehen Sie sich dazu die Software-Beispiele an.

SBC (Single Board Computer):

An der Stiftheiste neben dem „ISP-U“ Anschluss kann z.B. ein Raspberry Pi 2 oder 3 oder ein beliebiger anderer SBC angeschlossen werden. Es ist über diesen Anschluss auch möglich, Funkmodule oder Bluetooth®-Module als User-Board zu verwenden. Diese Stiftheiste ist als universell zu betrachten und stellt eine Schnittstelle für eigene Erweiterungen und Entwicklungen dar.

→ Der Locomotion-Controller befindet sich unter dem Lautsprecher.

9. Schematische Systemübersicht (Blockschaltbild)

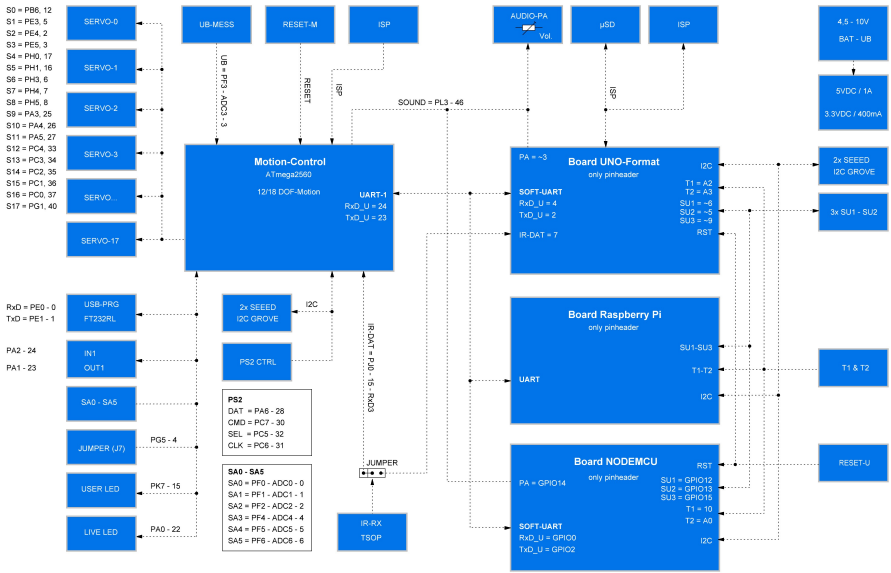


Bild 2

Das Diagramm zeigt schematisch die interne Verdrahtung und das Zusammenspiel der einzelnen Komponenten des Hexapod Roboter-Boards und dient als Hilfestellung bei der Programmierung.

10. Inbetriebnahme

Da es sich bei dem „Hexapod Roboter-Board“ um ein offenes Experimentier-Board (Development Board) handelt, können wir Ihnen in dieser Anleitung nur Tipps und Hinweise zum Aufbau Ihres eigenen Hexapoden geben. Für den fertigen lauffähigen Roboter sind Sie jedoch selbst verantwortlich!

Die folgende Abbildung zeigt einen Hexapod, der als Anhaltspunkt für Ihre eigene Konstruktion dienen kann. Dieser „Muster-Hexapod“ soll Ihnen das Abschätzen der Daten wie Robotergröße, Gewicht und die Auswahl der RC-Servos erleichtern.

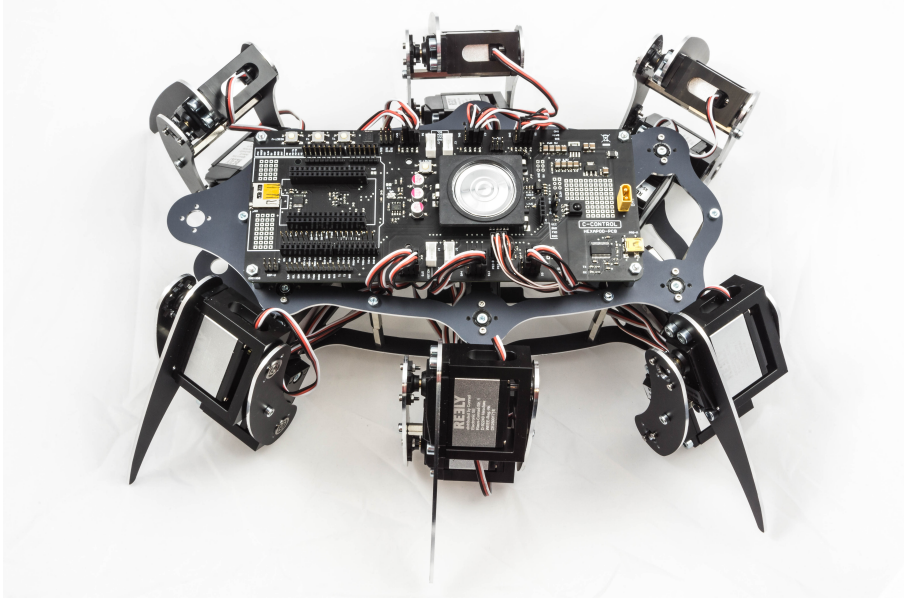


Bild 3

Technische Daten:

Akkupack:	5 Zellen NiMH, 3700 mAh (Anschlusskabel 2,5 mm ²)
Länge Körper:	280 mm
Höhe Körper ohne Platine:	65 mm
Breite Körper ohne Beine:	150 mm
Länge Coax:	50 mm
Länge Femur:	75 mm
Länge Tibia:	115 mm
Gewicht komplett:	ca. 2,7 kg
Stellkraft RC-Servo bei 6 V/DC:	8,5 kg
Anzahl RC-Servos:	18

11. Stromversorgung

Für einen sicheren und störungsfreien Betrieb des Hexapod Roboter-Boards ist die richtige Stromversorgung entscheidend. Das Board arbeitet mit einer Versorgungsspannung zwischen 4,5 und 10 V/DC, wobei eine optimale Stromversorgung von den verwendeten RC-Servos und Zusatzkomponenten abhängig ist.

Verwenden Sie z.B. analoge Standard RC-Servos, so ist in der Regel eine Versorgungsspannung von max. 6 V/DC zulässig (überprüfen Sie die Daten im Datenblatt der von Ihnen verwendeten RC-Servos nach).

Wir empfehlen, bei der Verwendung von Standard-RC-Servos mit einer Stellkraft von ≤ 8 kg und einer Versorgungsspannung von 6 V/DC einen 5zelligeren NiMH-Akkupack mit min. 2000 mAh zu verwenden. Bei zu geringer Kapazität des Akkupacks werden nur sehr kurze Laufzeiten des Roboters erzielt und es kann zum Reset des User-Boards bzw. des Locomotion-Controllers führen.

Sollten Sie für Ihren Roboter „High-Voltage RC-Servos“ verwenden (RC-Servos, die mit einem LiPo-Akku betrieben werden dürfen), so empfehlen wir einen 2S (7,4 V) LiPo-Akkupack mit min. 2000 mAh (Endladerate 20C).

Dem Lieferumfang des Hexapod Roboter-Boards liegt ein Akkustecker vom Typ XT30 bei. Dieser wird an dem von Ihnen ausgewählten Akku angelötet. Die Kabellänge zwischen Akku und Stecker darf 20 cm nicht überschreiten! Achten Sie auf die Polung und darauf, dass Sie beim Anlöten keinen Kurzschluss verursachen.



Akkus können im Kurzschlussfall sehr viel Strom abgeben, was zu Verbrennungen und Brandschäden sowie zu gefährlichen Verletzungen führen kann, außerdem zu einer Explosion des Akkus!

Bedenken Sie bei der Planung Ihres Hexapoden, dass dieser die RC-Servos, den Akku, die Mechanik, das Roboter-Board und Ihre zusätzlich verbauten Komponenten tragen muss!

Bedenken Sie auch, dass zu lange Beine einen sehr großen Hebel für die RC-Servos darstellen und die Stellkraft der RC-Servos zum Anheben und Laufen ausreichen muss. Werden die RC-Servos im Grenzbereich betrieben, so kann dies zur Überhitzung der Elektronik im Servo führen und diesen auf Dauer zerstören!



Bei der Verwendung von User-Boards, deren Stromversorgung kritisch ist (z.B. SBC oder leistungsstarke Arduino-UNO kompatible Boards mit ARM Controller), kann es sich zudem als sinnvoll erweisen, diese zusätzlich über eine Power-Bank zu versorgen.

In Bild 5 ist zu sehen, wie der Stecker am Hexapod Roboter-Board aufgesteckt wird.



Bild 4

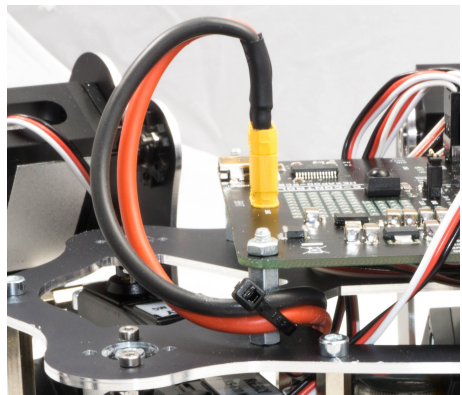


Bild 5

12. Installation der Software und Firmware

—> Die für das Hexapod Roboter-Board benötigten Software Pakete, Firmware und Tools stehen online als Download-Paket bereit. Dieses Paket wird regelmäßig aktualisiert und erweitert. Prüfen Sie daher gelegentlich, ob evtl. eine neue Version oder nützliche Erweiterungen zur Verfügung stehen.

Öffnen Sie Ihren Webbrowser und navigieren Sie entweder über unsere Shop-Website www.conrad.com auf die Produktseite des Hexapod Roboter-Boards, oder auf unsere Downloadseite.

Laden Sie sich hier das Paket „C-Control_Hexapod_Vx_x“ herunter (x_x bezeichnet hier die Version des Pakets).

Entpacken Sie die heruntergeladene ZIP-Datei auf Ihrer Festplatte.

Hier noch eine kurze Erklärung zu den enthalten Ordnern.

„Arduino“	Hier ist eine Textdatei mit dem Download-Link für die Software Arduino™ enthalten. Diese wird z.B. benötigt, um die Firmware ohne Updater aufzuspielen.
„Datasheets“	Datenblätter und Skizzen
„Driver“	Treiber für den FTDI USB-Chip
„Library-Demos“	Enthält die Arduino Library, Demos und die Locomotion-Firmware als ZIP-File. Wird in der Arduino IDE installiert werden.
„Schematic“	Schaltplan des C-Control Hexapod Roboter-Boards als PDF-Datei
„Terminal“	Kalibriersoftware um den Hexapod einzustellen
„Manual“	Enthält diese Anleitung als PDF-Datei

13. Übertragen der Firmware

Bei Auslieferung ist auf dem Hexapod Roboter-Board keine Firmware installiert. Die Firmware muss bei der Erstinbetriebnahme zunächst übertragen werden; dies erfolgt über die Arduino IDE.

Die nachfolgende Beschreibung zeigt Ihnen die Vorgehensweise zum Übertragen der Firmware. In den nachfolgenden Abschnitten werden Sie unterschiedliche Programme auf das Hexapod Roboter-Board übertragen müssen. Der Vorgang ist bis auf die Auswahl der jeweiligen Firmware identisch.



Beim Übertragen der Firmware muss das Hexapod Roboter-Board mit der Stromversorgung verbunden sein!

a) Treiber installieren

Verbinden Sie den „PRG-M“ Anschluss des Hexapod Roboter-Boards mit einem freien USB-Anschluss Ihres Computers. Windows wird nun versuchen, einen neuen Treiber zu installieren. Im Regelfall wird Windows die Treiber automatisch aus dem Internet herunterladen und installieren, da der Treiber für den verwendeten FTDI USB-Brückenchip im Windows-Treiberdownload verfügbar ist (der Computer muss hierfür mit dem Internet verbunden sein).

Sollte diese Methode nicht funktionieren, verweisen Sie bei der Installation manuell auf den Treiber. Der Treiber befindet sich im entpackten Download-Bundle, im Verzeichnis „Driver“.

Kontrollieren Sie nach der Installation des Treibers im Gerätemanager, ob der Treiber für das Hexapod Roboter-Board installiert wurde und welche „COM-Port-Nummer“ zugeteilt wurde. Wählen Sie diese dann in der Arduino IDE aus.

Die Treiber können Sie auch direkt unter <http://www.ftdichip.com> herunterladen.

b) Arduino IDE installieren

Laden Sie sich die neueste Version der Arduino IDE unter www.arduino.cc herunter. Es gibt die Arduino IDE als „Installer-Version“ und als „ZIP-Archiv“. Beide Versionen können verwendet werden.

Bei der Installer-Version installieren Sie die Arduino IDE wie auch von anderen Programmen gewohnt. Bei der ZIP-Version müssen Sie nach dem Download das ZIP-Archiv an die gewünschte Stelle auf Ihrem Computer entpacken. Danach starten Sie das Programm mit einem Doppelklick auf die Datei „arduino.exe“.

c) Arduino IDE einrichten



Bild 6

Starten Sie die Arduino IDE und folgen Sie den Anweisungen dieser Bedienungsanleitung.

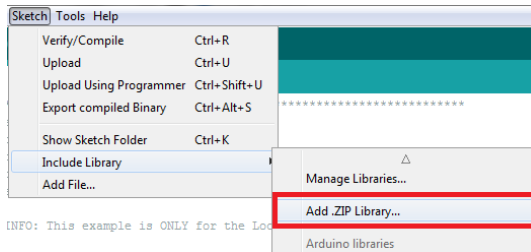


Bild 7

Installieren Sie nun die Hexapod Bibliothek. Diese befindet sich im Download-Bundle unter „\Library-Demos\C-Control-Hexapod.zip“. Wählen Sie dazu in der Arduino IDE den Menüpunkt „Sketch\Include Library\Add .ZIP Library...“ aus.

→ Sollten Sie später einmal ein Update der Bibliothek durchführen, so müssen Sie die alte Bibliothek zuvor löschen, da sonst die Arduino IDE einen Fehler meldet!

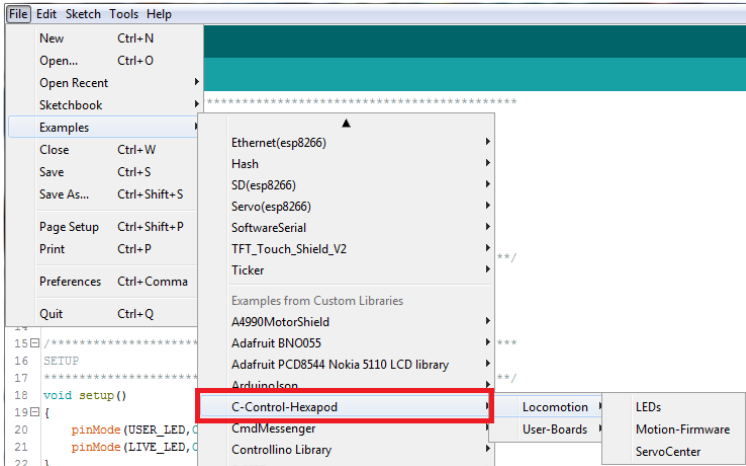


Bild 8

Nachdem Sie die Bibliothek installiert haben, finden Sie im Menüpunkt „\File\Examples\C-Control-Hexapod“ zwei Ordner mit den Namen „Locomotion“ und „User-Boards“ vor. Der Ordner „Locomotion“ enthält Programme für den Locomotion-Controller, der auf dem Hexapod Roboter-Board fest verbaut ist und für das Laufen zuständig ist. Der Ordner „User-Boards“ enthält Programme für die Arduino kompatiblen User-Boards wie Arduino UNO und NodeMCU.

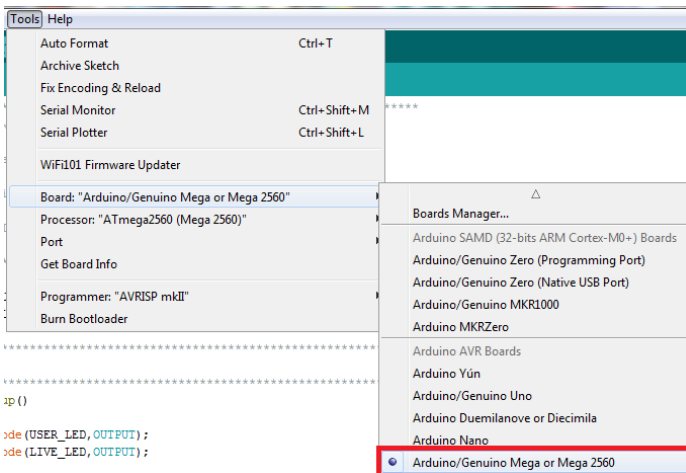


Bild 9

Wählen Sie den verwendeten Mikrocontroller unter dem Menüpunkt „Tools\Board\Arduino/Genuino Mega“ aus.

→ Der Locomotion-Controller ist mit dem Arduino MEGA2560 kompatibel.

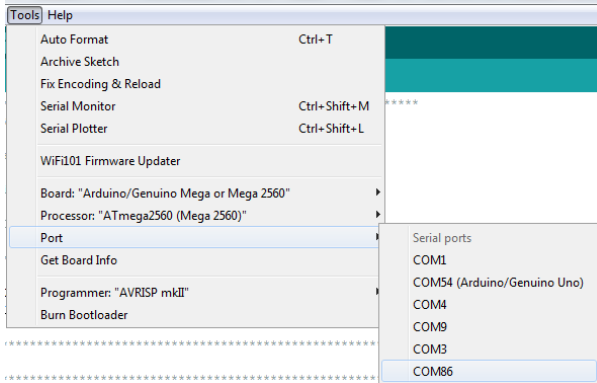


Bild 10

Wählen Sie den COM-Port unter „Tools\Port“ aus (wählen Sie den zuvor installierten COM-Port aus).

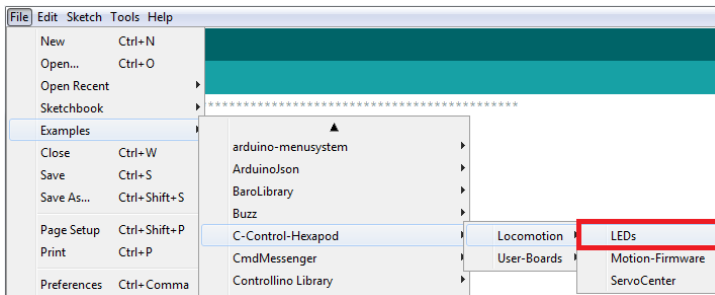


Bild 11

Um zu testen, ob alles funktioniert, übertragen Sie das Beispielprogramm „User-Board\LEDs“. Das Programm lässt nach erfolgreicher Übertragung die blaue „LIVE-LED“ und rote „USER-LED“ abwechselnd blinken.

Nach der Auswahl des „LEDs“-Programms (in Arduino auch Sketch genannt) öffnet sich ein neues Arduino IDE Fenster mit dem „LEDs“ Sketch.

Mit dem „Pfeil nach rechts“-Symbol (siehe Bild 12) im Menü übertragen Sie das Programm auf das Hexapod Roboter-Board. Alternativ können Sie auch den Shortcut „CTRL+U“ verwenden.



Dieser Vorgang gilt für alle zu übertragenden Programme (Sketches), die für den Locomotion-Controller bestimmt sind. Für die Beispiele der User-Boards ändern Sie nur das verwendete User-Board und den „COM-Port“.

Bild 12



Bitte lesen Sie sich den jeweiligen Anfangskommentar der Beispiele (Examples) vor dem Aufspielen durch. Dieser enthält wichtige Informationen zum Programm!



Sollte die Übertragung nicht klappen, so überprüfen Sie Ihre Einstellungen für Board und Anschluss und prüfen Sie zudem, ob das Board richtig im Gerätemanager eingetragen und installiert wurde und die Stromversorgung angeschlossen ist.

Hier finden Sie weitere Information zur Installation der Arduino IDE:

<http://arduino.cc/en/Guide/Windows>

14. RC-Servos für den Einbau vorbereiten

Bevor Sie die RC-Servos in Ihrer Mechanik verbauen, müssen Sie diese in die Mittelstellung bringen.

Packen Sie zunächst alle 18 RC-Servos aus und legen Sie sich die Zubehörkleinteile, die bei den RC-Servos enthalten sind, sauber sortiert nach Art (Schrauben, Servohörner etc.) beiseite, so dass diese nicht verloren gehen. Diese benötigen Sie erst zum Einbau in die Mechanik wieder!



Bild 13

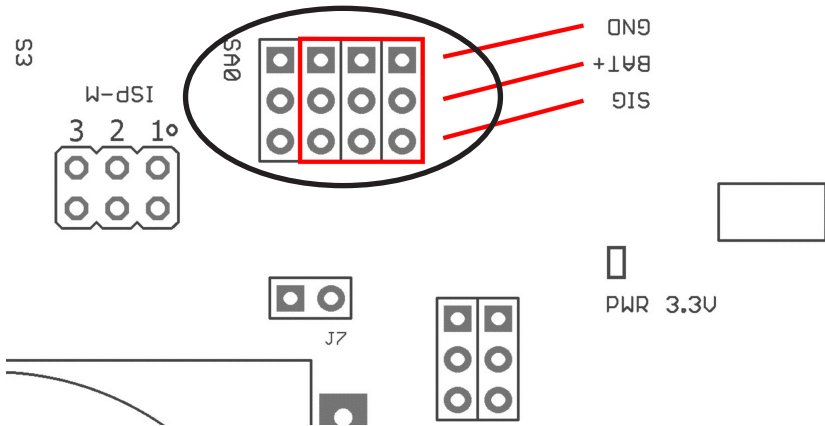


Bild 14

Schließen Sie die RC-Servos an den Stiftleisten wie in der Abbildung an (achten Sie auf die Polung!). Es werden alle Servoanschlüsse (S0 bis S17) des Roboter-Boards mit einer Pulsbreite von 1500 µs angesteuert.

Übertragen Sie zum Einstellen der RC-Servos den Sketch „ServoCenter“ auf das Hexapod Roboter-Board. Das Programm lässt nach dem Start die RC-Servos S0 bis S17 in die Mittelstellung fahren.

➔ Sie müssen nicht alle 18 RC-Servos gleichzeitig am Hexapod Roboter-Board anstecken. In der Praxis hat sich bewährt, dass es einfacher ist, nur einen Anschluss (z.B. S0) oder maximal 3 (S0 bis S2) zu verwenden.

Belassen Sie dieses Programm auf dem Hexapod Roboter-Board, bis der Roboter fertig zusammengebaut ist. Sollten Sie beim Zusammenbau ein Servo verdrehen, so können Sie dieses wieder anstecken und in Mittelstellung fahren lassen. Es hat sich in der Praxis bewährt, jedes einzelne Bein, welches fertig montiert wurde, mit allen 3 Servos am Board anzustecken und zu kontrollieren, ob die Servostellung noch korrekt ist. Wird das Servo beim Einbau aus der Mittelstellung gedreht, wirkt sich dies negativ auf die Laufeigenschaften aus bzw. kann dies Servo und Mechanik beschädigen.



Stecken Sie die RC-Servos nur an den jeweils ersten 3 Stifteleisten (im Abbildungsausschnitt rot bzw. mit einem Kreis gekennzeichnet) an. Der jeweils letzte Anschluss mit der Bezeichnung SA0 bis SA5 (im Abbildungsausschnitt SA0) ist für eigene Erweiterungen vorgesehen und führt kein Steuersignal! Die Anschlüsse SA0 bis SA5 sind mit den analogen Eingängen des Locomotion-Controllers verbunden. Die Anschlusszuordnung finden Sie in der Porttabelle im Anhang und in der Systemübersicht (Blockschaltbild).

Trennen Sie das Roboter-Board von der Stromversorgung, wenn Sie es längere Zeit nicht benutzen, um versehentliche Kurzschlüsse zu vermeiden!

→ RC-Servos besitzen eine Elektronik, einen Motor und ein Potentiometer. Das Potentiometer greift die Position des Servos ab und leitet diese an die Elektronik weiter. Die Elektronik im RC-Servo vergleicht die vorgegebene Stellung (Sollwert), in unserem Fall die Pulsbreite gegeben vom Locomotion-Controller mit der des Potentiometers im RC-Servo (Istwert) und regelt den Motor auf die Sollposition.

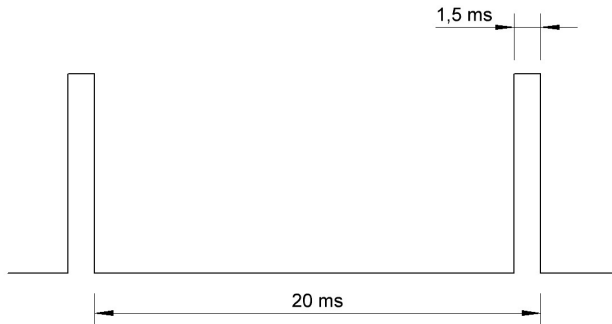


Bild 15

Der RC-Servo bekommt als Signal vom Mikrocontroller einen positiven Puls mit einer Pulsbreite zwischen $500 \mu\text{s}$ und $2500 \mu\text{s}$. Die gesamte Periodenlänge des Steuersignals beträgt 20 ms . Die Mittelstellung des RC-Servos beträgt eine Pulsbreite von $1,5 \text{ ms}/1500 \mu\text{s}$.

Die Grafik veranschaulicht das Puls/Periodenverhältnis wie es zur Ansteuerung eines RC-Servos nötig ist. In diesem Fall hat der Puls eine Länge von $1,5 \text{ ms}/1500 \mu\text{s}$, was der Mittelstellung des RC-Servos entspricht.

Wird die Pulsbreite kleiner $1500 \mu\text{s}$, so dreht das RC-Servo in die eine Richtung. Wird die Pulsbreite größer $1500 \mu\text{s}$, so dreht es in die andere Richtung. Der Drehwinkel ist dabei proportional zur Pulsbreite, d.h. je länger oder kürzer der Puls, desto mehr oder weniger dreht das RC-Servo in die entsprechende Richtung.

15. Hexapod-Mechanik und Grundkonfiguration

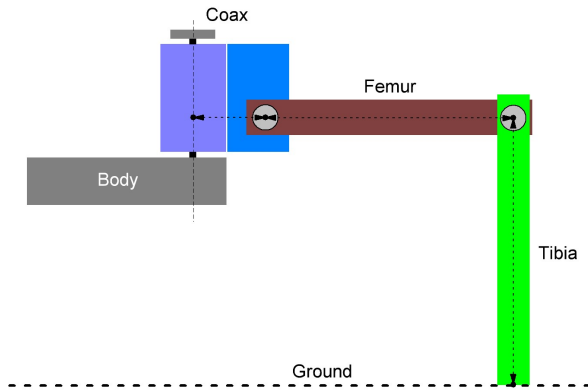


Bild 16

Die Abbildung zeigt die Grundkonfiguration eines Hexapod-Beins. Nachdem Sie die Servos im letzten Abschnitt in die Mittelstellung gebracht haben, können Sie nun Ihre Roboterbeine montieren.

Zu beachten ist, dass der Oberschenkel („Femur“) parallel zum Boden („Ground“ oben im Bild) ausgerichtet wird.

Der Drehpunkt des Schienbeines („Tibia“) steht parallel zum Körper („Body“). Wenn das Bein fertig eingestellt ist, ergibt es zwischen „Femur“ und „Tibia“ einen 90°-Winkel. Die Anordnung zwischen „Body“ und „Femur“ verlaufen in einer Linie parallel. Der Höhenunterschied spielt hierbei keine Rolle.

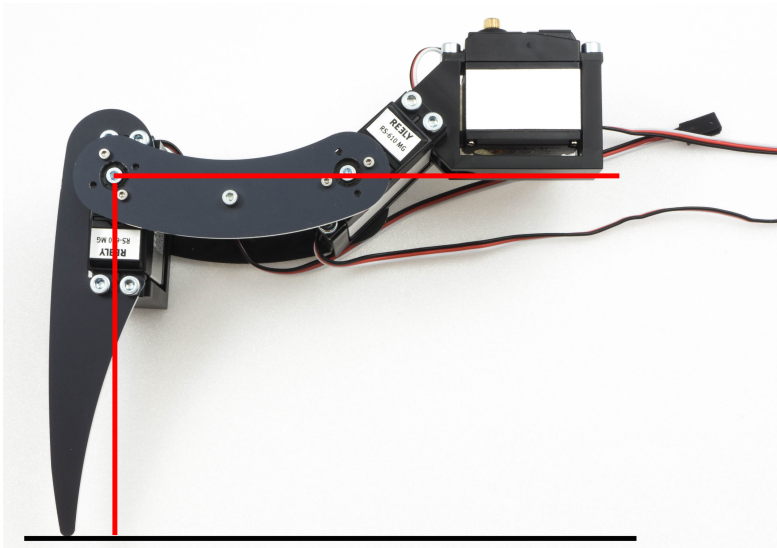


Bild 17

Die Abbildung zeigt zur realen Veranschaulichung ein Bein des Muster-Hexapoden. Trotz des abgewinkelten Tibia-Servo müssen am Ende „Femur“ parallel zum Boden und „Tibia“ parallel zum Körper stehen. „Femur“ und „Tibia“ ergeben am Ende einen 90° Winkel! Die RC-Servos selbst sind weiterhin in Mittelstellung!

Sollten Sie die Beine, bedingt durch die Zahnung der Servoscheibe bzw. der allgemeinen mechanischen Toleranzen, noch nicht zu genau ausrichten können, so stellt dies kein Problem dar. Stellen Sie das Bein so gut wie möglich mechanisch ein, wie in der Abbildung zu erkennen. Eine genaue Kalibrierung erfolgt nach dem Zusammenbau mit Hilfe der Software.

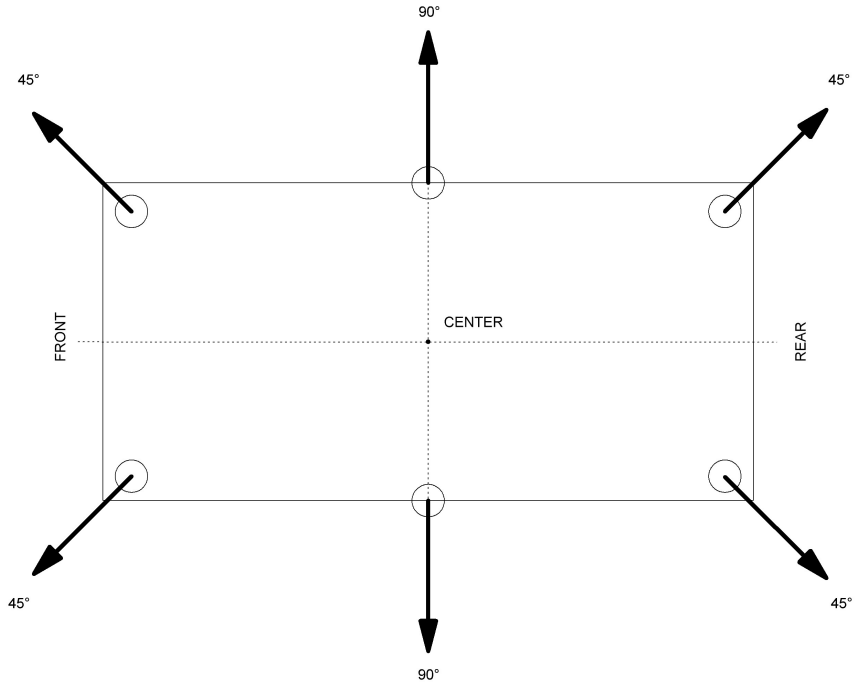


Bild 18

Die Skizze veranschaulicht den Hexapod-Körper von „oben“ betrachtet. Die zwei vorderen und die beiden hinteren Beine werden so eingebaut, dass sie in einem Winkel von 45° nach außen zeigen, ohne dabei die Mittelstellung des RC-Servos zu verdrehen.

Die beiden mittleren Beine (linke sowie rechte Seite) werden gerade nach außen (90°) eingebaut.

Die Abbildung zeigt zur realen Veranschaulichung die fertig ausgerichteten Beine des Muster-Hexapoden in der Grundkonfiguration.

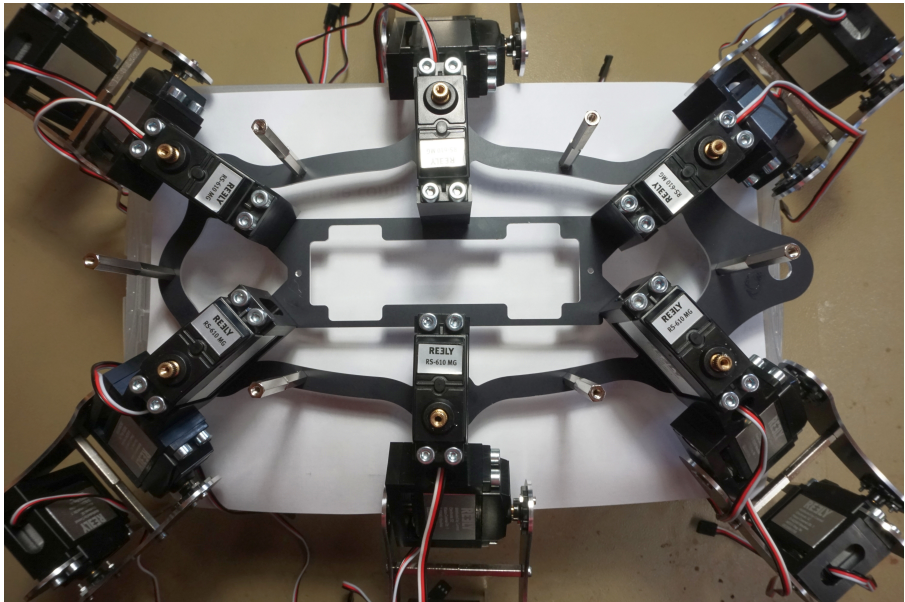


Bild 19

Stecken Sie die RC-Servos wie folgt am Hexapod Roboter-Board an:

Zuordnung RC-Servo zu Bein:

RR („right rear“ = hinten rechts)

S0 = Coax

S1 = Femur

S2 = Tibia

RM („right middle“ = Mitte rechts)

S3 = Coax

S4 = Femur

S5 = Tibia

RF („right front“ = vorne rechts)

S6 = Coax

S7 = Femur

S8 = Tibia

LR („left rear“ = hinten links)

S9 = Coax

S10 = Femur

S11 = Tibia

LM („left middle“ = Mitte links)

S12 = Coax

S13 = Femur

S14 = Tibia

LF („left front“ = vorne links)

S15 = Coax

S16 = Femur

S17 = Tibia

17. Gamepad anschließen

Um den aufgebauten Hexapod-Roboter zu steuern und die Einstellungen komfortabel und schnell zu überprüfen, ist es vorteilhaft, den Roboter mit dem Gamepad zu bedienen. Alternativ können Sie ihn auch gleich über ein User-Board steuern. Dies hat sich in der Praxis jedoch für die erste Inbetriebnahme als weniger komfortabel herausgestellt.

An das Hexapod Roboter-Board kann jedes handelsübliche PS2-kompatible Gamepad (kabelgebundene bzw. drahtlose Version) angeschlossen werden. Sie müssen dazu die Kontakte des Gamepads mit Kabel und RC-Servostecker (3polig, Rastermaß 2,54 mm) versehen.

Mit dem Gamepad stehen Ihnen nach der Inbetriebnahme sofort alle Funktionen des Locomotion-Controllers zur Verfügung und der Roboter kann ähnlich wie ein ferngesteuertes Auto bewegt werden, ohne zuvor ein eigenes Programm erstellen zu müssen.

Zudem können Sie später bei der Nutzung eines User-Boards mit Hilfe des Gamepads in den Ablauf Ihres Programms eingreifen. Es ist auch möglich, die Steuerbefehle (Joystickwerte und Taster) vom User-Board auszulesen und diese im eigenen Programm zu verwenden.

→ Wir empfehlen die Verwendung speziell für den Einsatz mit Mikrocontrollern vorbereiteten Funk-Gamepads, Conrad-Best.-Nr. 1613300, bestehend aus Funkempfänger und Controller.



Vergewissern Sie sich, dass der Empfänger richtig angeschlossen ist und die Stecker nicht verdreht auf den Pinleisten stecken, da sonst der Empfänger Schaden nehmen kann! Der Jumper „J7“ darf nicht aufgesteckt sein!



Bild 22

- 1 Gamepad (PS2-kompatibel)
- 2 Gamepad-Empfänger
- 3 Anschlussstecker zum Verbinden mit dem Hexapod Roboter-Board

Pinbelegung des PS2-kompatiblen Gamepads:

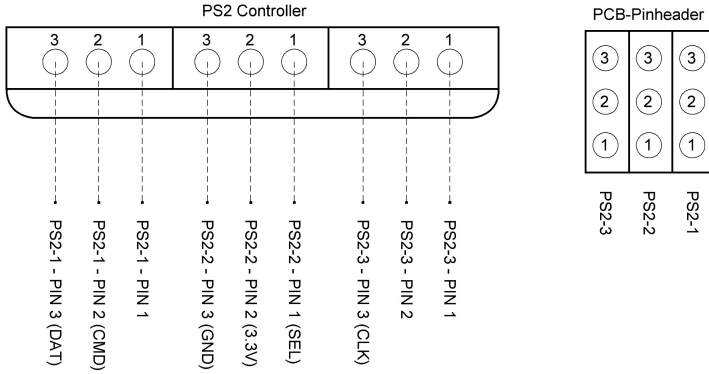


Bild 23

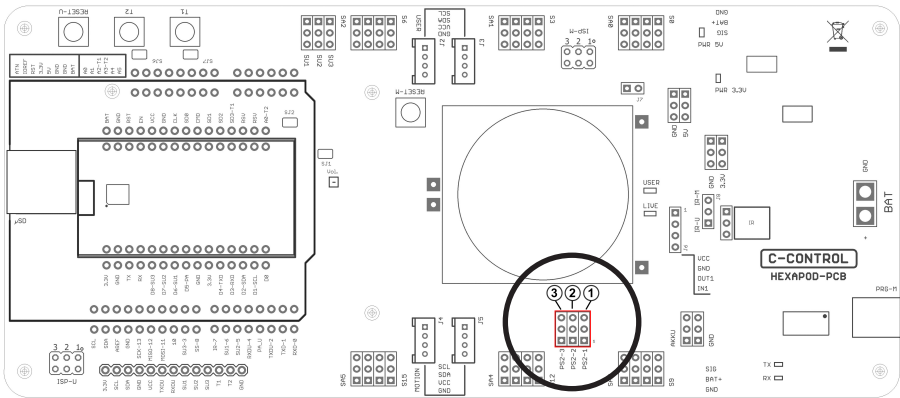


Bild 24

- 1 PS2-1
- 2 PS2-2
- 3 PS2-3



Pin 1 der drei jeweils 3poligen Stiftleisten (PS2-1, PS2-2 und PS2-3), liegt zum Rand der Platine hin.

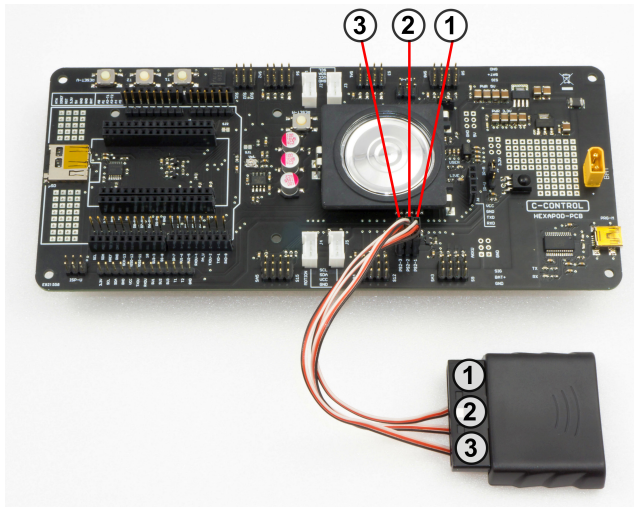


Bild 25

Der Empfänger wird mit den 3 Anschlusskabeln, wie in der Abbildung gezeigt, an das Hexapod Roboter-Board angeschlossen. Kabel (1) kommt auf Pinheader (1), Kabel (2) auf Pinheader (2), Kabel (3) auf Pinheader (3).



Die Kabelfarben können zu Ihrem verwendeten Gamepad abweichen! Halten Sie sich an die Pinbelegung beim Anschließen!

Vergewissern Sie sich, dass der Empfänger richtig angeschlossen ist und die Stecker nicht verdreht auf den Pinleisten stecken, da sonst der Empfänger Schaden nehmen kann!

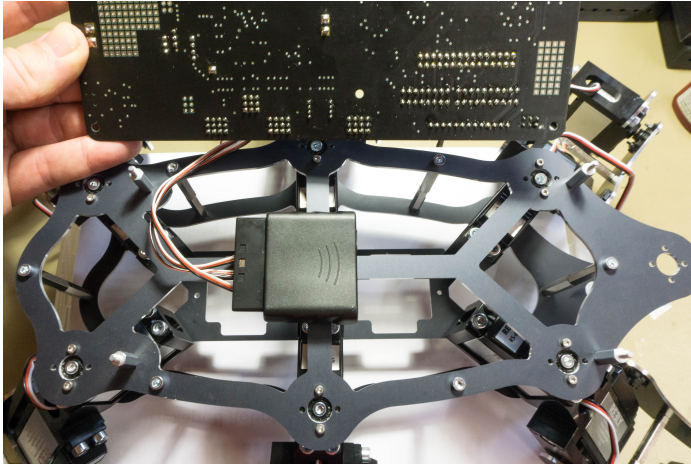


Bild 26

Beim Muster-Hexapod wurde der Empfänger unter dem Roboter-Board mit Servo-Tape (doppelseitiges Klebeband mit Schaumstoff zwischen den Klebeschichten) montiert.

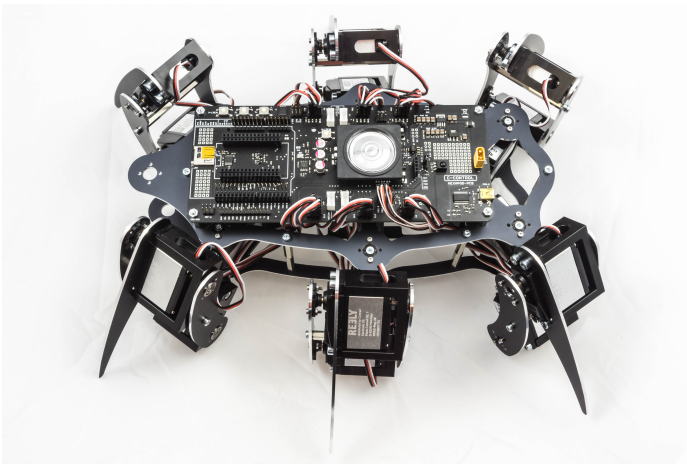


Bild 27

Durch die verdeckte Montage sieht der Muster-Hexapod sehr aufgeräumt aus. Sie können den Empfänger natürlich an einer beliebigen Stelle an Ihren Hexapod montieren. Die Bilder des Muster-Hexapoden sollen nur als Veranschaulichung dienen.

→ Montieren Sie den Empfänger nicht zu nahe an den RC-Servos! Diese können durch die Motoren Störungen im Empfänger erzeugen, was eine minimierte Reichweite zur Folge haben kann. Im schlimmsten Fall kommt keine Bindung zwischen Gamepad und Empfänger zustande.

18. Firmware-Konfiguration

Sie müssen zunächst die „Locomotion-Firmware“ auf die Parameter Ihres Hexapoden anpassen. Öffnen Sie dazu in der Arduino IDE den Sketch „Motion-Firmware“. Dieser Sketch enthält die komplette Firmware zur Steuerung des Hexapoden.

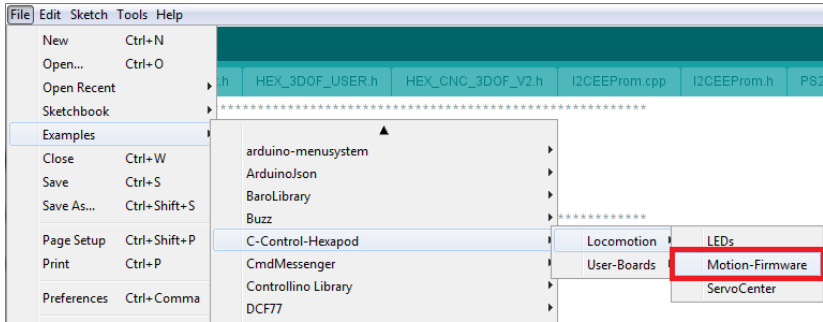


Bild 28

Folgenden Parameter müssen in den nachfolgenden Dateien angepasst werden:

Datei: Motion-Firmware.ino

In dieser Datei sind bereits verschiedene Mechaniken (Body-Frames) hinterlegt. Um Ihre Mechanik auszuwählen, entfernen Sie die Kommentarzeichen „//“ bei „//include HEX_3DOF_USER.h“. Die restlichen #include „HEX_xxx_xxx.h“ bleiben mit „//“ auskommentiert. Dadurch wird beim Kompilieren nur Ihre Header-Datei „HEX_3DOF_USER.H“ verwendet.

```
/*  
ROBOT-MECHANIC  
*/  
#include „HEX_3DOF_USER.h“  
//#include „HEX_CNC_3DOF_V2.h“
```


Datei: HEX_3DOF_USER.h

Min/Max Speed

Diese Parameter können Sie für die ersten Tests so belassen. Die Parameter geben die maximale und minimale Laufgeschwindigkeit an. SPEED_INIT ist die default Geschwindigkeit nach dem Einschalten. Die Parameter legen die Zykluszeit in Millisekunden fest. Aus diesem Grund ist der Wert von SPEED_MAX kleiner als der von SPEED_MIN.

```
#define SPEED_MIN      200
#define SPEED_MAX      25
#define SPEED_INIT     100
```

Schritthöhe

Diese Parameter können Sie für die ersten Tests so belassen. Die Parameter geben die normale und die erhöhte Schritthöhe an, also wie hoch ein Bein beim Laufen angehoben wird.

```
#define WALK_HEIGHT_NORMAL    40
#define WALK_HEIGHT_DOUBLE    100
```

RC-Servo Anschlüsse und Inverse-Funktion

In der folgenden Datei werden die RC-Servo Anschlüsse den Arduino Pins zugeordnet. Diese sind bereits in der Datei festgelegt und müssen für das Hexapod Roboter-Board nicht verändert werden. Mit den Definitionen mit der Endung „...Inv“ kann die Drehrichtung der RC-Servos umgedreht werden. Für die meisten Standard-RC-Servos können die vorgegebenen Einstellungen belassen werden. Drehen sich die RC-Servos beim Gehen in die falsche Richtung, so müssen die Werte von 1 auf 0 bzw. von 0 auf 1 abgeändert werden.

→ Ob die RC-Servos in die richtige Richtung drehen, können Sie sehr leicht herausfinden, indem Sie nach dem Übertragen der Firmware den Roboter mit Hilfe des Gamepads einschalten und aufstehen lassen. Fahren die Beine dabei nicht aus, sondern entgegengesetzt, so dass sich der Körper nicht anhebt, müssen die „Invers-Defines“ geändert werden.

```
/*
*****
Servo pin defines and inverse function
*****
// --> LEFT
#define cLFCoxaPin      36      // S15
#define cLFFemurPin     37      // S16
#define cLFTibiaPin     40      // S17
#define cLFCoxaInv      1
#define cLFFemurInv     1
#define cLFTibiaInv     1

#define cLMCoxaPin      33      // S12
#define cLMFemurPin     34      // S13
#define cLMTibiaPin     35      // S14
#define cLMCoxaInv      1
#define cLMFemurInv     1
#define cLMTibiaInv     1
*/
```

```

#define CLRCoxaPin      25      // S9
#define CLRFemurPin    26      // S10
#define CLRTibiaPin    27      // S11
#define CLRCoxaInv     1
#define CLRFemurInv    1
#define CLRTibiaInv    1

// --> RIGHT
#define CRFCoxaPin     6       // S6
#define CRFFemurPin    7       // S7
#define CRFTibiaPin    8       // S8
#define CRFCoxaInv     0
#define CRFFemurInv    0
#define CRFTibiaInv    0

#define CRMCoxaPin     3       // S3
#define CRMFemurPin    17      // S4
#define CRMTibiaPin    16      // S5
#define CRMCoxaInv     0
#define CRMFemurInv    0
#define CRMTibiaInv    0

#define CRRCoxaPin     12      // S0
#define CRRFemurPin    5       // S1
#define CRRTibiaPin    2       // S2
#define CRRCoxaInv     0
#define CRRFemurInv    0
#define CRRTibiaInv    0

```

Abmessung der Beine

Geben Sie bei „LEG DIMENSIONS“ die Maße Ihres Roboterbeines in Millimetern (mm) ein.

Messen Sie bei „cXXCoxaLength“ und „cXXFemurLength“ von Drehpunktmitte zur Drehpunktmitte und bei „cXXTibialLength“ von Drehpunktmitte bis zum Ende des Beines.

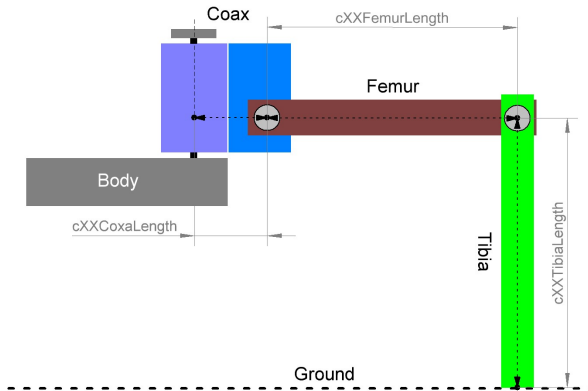


Bild 29



Messen Sie sehr sorgfältig. Eine falsche Angabe wird sich negativ auf das Laufverhalten auswirken!

```
/*  
LEG DIMENSIONS  
Universal dimensions for each leg in mm  
*/  
#define cXXCoxaLength      51  
#define cXXFemurLength    75  
#define cXXTibialLength   117
```

Min/Max Servo-Drehwinkel

Die Werte für „MIN-MAX ANGLES“ können Sie zunächst so wie vorgegeben belassen. Diese Werte begrenzen den Drehwinkel der RC-Servos, d.h. wie weit die Beine sich in einem mechanisch sicheren Bereich ohne Blockieren ansteuern lassen. Die bereits vorgegebenden Werte sind ein guter Mittelwert aus der Praxis. Sie können diese Werte später noch ändern und genauer auf Ihre verbauten RC-Servos anpassen.

Die Werte sind von der Grundkonfiguration eines Hexapod Beines aus zu betrachten und werden in 1/10 Grad angegeben. D.h. eine Angabe der Begrenzung von max. 300 und min. -300 lässt eine Ansteuerung von +/- 30° zu.

```
/******
```

```
MIN-MAX ANGLES
```

```
***** /
```

```
#define cCoxaMin      -300  
#define cCoxaMax      300  
#define cFemurMin     -1050  
#define cFemurMax     750  
#define cTibiaMin     -530  
#define cTibiaMax     900
```

Hexapod Body-Masse

Die Werte für „BODY DIMENSIONS“ sind für ein Body-Setup von 45° vorgegeben. D.h. die vorderen und hinteren Beine des Roboters stehen im 45° Winkel, wie in der Abbildung zu sehen, vom Body weg. Wünschen Sie andere Werte, so ändern Sie die Parameter entsprechend ab. Die Werte werden wieder in 1/10 Grad angegeben.

```

/*****
BODY DIMENSIONS
45° setup
*****/
#define cRRCoxaAngle1    -450
#define cRMCoxaAngle1    0
#define cRFCoxaAngle1    450
#define cLRCoxaAngle1    -450
#define cLMCoxaAngle1    0
#define cLFCoxaAngle1    450

```

Um die Body Offset-Werte zu ermitteln, benutzen Sie folgende Abbildung:

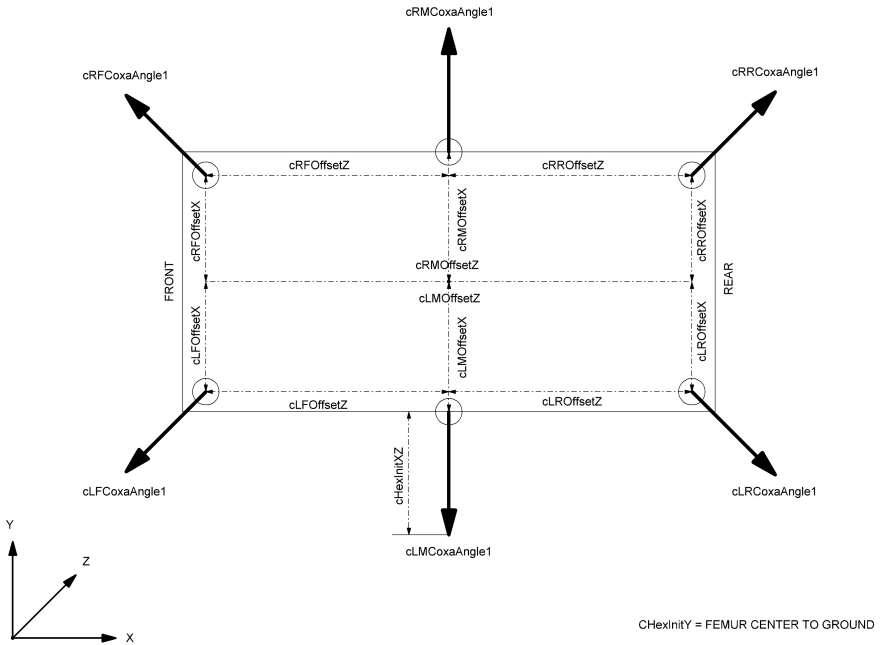


Bild 30

```

#define cRROffsetX -40 // Distance X from center of the body to the Right Rear coxa
#define cRROffsetZ 100 // Distance Z from center of the body to the Right Rear coxa
#define cRMOffsetX -55 // Distance X from center of the body to the Right Middle coxa
#define cRMOffsetZ 0 // Distance Z from center of the body to the Right Middle coxa
#define cRFOffsetX -40 // Distance X from center of the body to the Right Front coxa
#define cRFOffsetZ -100 // Distance Z from center of the body to the Right Front coxa

#define cLROffsetX 40 // Distance X from center of the body to the Left Rear coxa
#define cLROffsetZ 100 // Distance Z from center of the body to the Left Rear coxa
#define cLMOffsetX 55 // Distance X from center of the body to the Left Middle coxa
#define cLMOffsetZ 0 // Distance Z from center of the body to the Left Middle coxa
#define cLFOffsetX 40 // Distance X from center of the body to the Left Front coxa
#define cLFOffsetZ -100 // Distance Z from center of the body to the Left Front coxa

```

Die nun folgenden Parameter sind etwas schwieriger einzustellen und müssen teils experimentell ermittelt werden, bis der korrekte Wert gefunden ist.

Der Parameter „cHexInitXY“ gibt den Anstand von Drehpunkt des Coax-Servos zum Bein-Auftrittspunkt an.

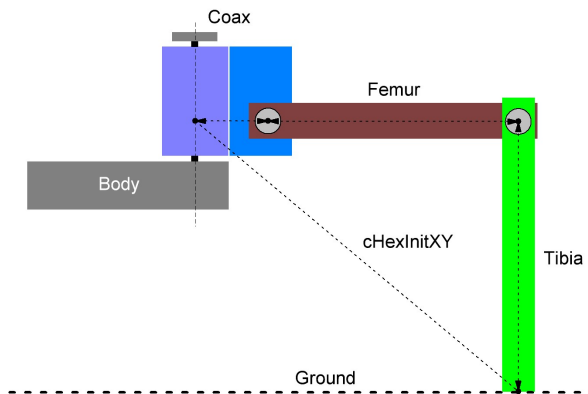


Bild 31

Abbildung (XZ) gibt den Abstand vom Coax-Servo-Drehpunkt zum Bein-Ende wieder.

Die Werte „cHexInitXZCos45“ und „cHexInitXZSin45“ werden aus dem Wert „cHexInitXZ“ ermittelt. Wenn Sie eine andere Beinkonstellaton verwenden möchten (z.B. dass die vorderen und hinteren Beine nicht wie hier vorgegeben im 45° Winkel stehen sondern z.B. im 60° Winkel), so ändern Sie cos(45) und sin(45) auf den entsprechenden Winkel z.B. 60 ab.

```

/*****
START POSITIONS FEET -> nach dem Aufstehen!
*****/
#define cHexInitXZ      110 // 110 distance Coax servo to Tibia end (ground contact)
#define cHexInitXZCos  cos(45) * cHexInitXZ
#define cHexInitXZSin  sin(45) * cHexInitXZ
#define cHexInitY      35 // Center Femur servo after start!

```

Der Parameter „MAX_BODY_Y“ gibt die maximale Höhe die der Body erreichen kann an. Je nach Bein-Abmessungen können Sie hier die Werte „g_abHexIntXZ“ und „g_abHexMaxBodyY“ noch anpassen.

Die drei Parameter bei „g_abHexIntXZ“ sind der Abstand vom Drehpunkt Coax-Servo zu den Bein-Enden, die am Boden stehen. Wird der Wert vergrößert, so gehen beim Erhöhen die Beine weiter weg vom Body. Die Angabe der Werte erfolgt in Millimetern! Es ist vom Vorteil wenn die Beine nicht zu weit vom Körper weg zeigen, sondern näher an den Körper sind, da hierdurch die Hebelwirkung auf die RC-Servos verringert wird!

```

/*****
Lets try some multi leg positions depending on height settings
*****/

```

```

#define CNT_HEX_INITS 3
#define MAX_BODY_Y    135 // max body height
#ifdef DEFINE_HEX_GLOBALS
    const byte g_abHexIntXZ[] PROGMEM = {cHexInitXZ, 125, 125};
    const byte g_abHexMaxBodyY[] PROGMEM = {35, 50, MAX_BODY_Y};
#else
    extern const byte g_abHexIntXZ[] PROGMEM;
    extern const byte g_abHexMaxBodyY[] PROGMEM;
#endif

```

Alle anderen Werte in dieser Datei müssen Sie nicht anpassen.

19. Firmware übertragen

Lesen Sie vor dem ersten Übertragen das Kapitel „Steuern mit dem Gamepad“, sofern Sie mit der Bedienung noch nicht vertraut sind.

Danach können Sie das Programm mit den neuen Parametern auf das Hexapod Roboter-Board übertragen. Sofern noch nicht alle Werte 100% stimmen, passen Sie die Werte weiter an Ihren Hexapod Body an.



Prüfen Sie vor dem Übertragen und Einschalten des Hexapod Roboter-Boards noch Folgendes:

- Sind alle RC-Servos richtig angeschlossen (Polung)?
 - Sind alle RC-Servos mechanisch in der Mitte?
 - Ist der PS2-Controller angeschlossen?
 - Ist die Stromversorgung ausreichend?
 - Steht genügend Fläche für den Roboter auch bei einem Fehlverhalten zur Verfügung?
- > Bedingt durch die Zahnung der Ruderhörner der RC-Servos stehen die Beine nach dem Zusammenbau mechanisch nicht 100% gerade. Lesen Sie dazu das Kapitel „Hexapod kalibrieren“, bevor Sie die Werte weiter anpassen. Mit der Kalibrierung können Sie die Beine exakt auf die Grundkonfiguration justieren. Erst danach macht es Sinn, die Parameter weiter anzupassen!

20. Steuern mit dem Gamepad

Nachdem Sie die „Motion-Firmware“ auf den Locomotion-Controller übertragen haben, können Sie den Roboter mit dem Gamepad steuern.

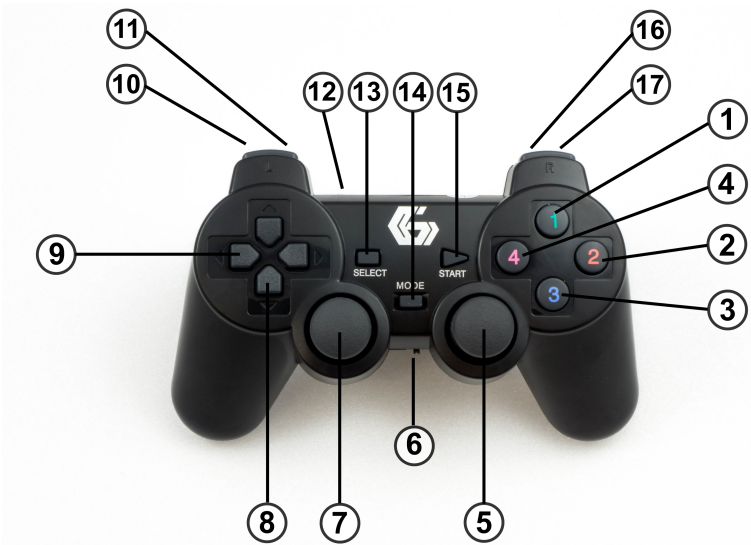


Bild 32

Schnellstart:

- Trennen Sie die Stromversorgung vom Hexapod Roboter-Board.
- Schalten Sie das Gamepad mit dem Schalter (6) ein. Die LED (12) „Grün“ und LED „Rot“ blinken.
- Stecken Sie das Hexapod Roboter-Board an der Stromversorgung (Akku) an. LED (12) „Grün“ und LED „Rot“ leuchten nach ein paar Sekunden dauerhaft. Der Sender ist nun mit dem Empfänger verbunden.
- Drücken Sie die Taste „START“ (15). Der Roboter piepst und die rote „USER-LED“ auf dem Hexapod Roboter-Board leuchtet dauerhaft.
- Schalten Sie den Roboter mit der Taste (3) ein. Der Roboter piepst und die blaue „LIVE-LED“ blinkt.
- Lassen Sie den Roboter mit der Taste (1) aufstehen.
- Bewegen Sie den Roboter mit den Daumenjoystick (7) vor/zurück und links/rechts. Mit dem Daumenjoystick (5) können Sie den Roboter auf der Stelle wenden.

Ausführliche Beschreibung der Gamepad-Funktionen:

- 1 Roboter aufstehen und hinsetzen lassen (Toggle Funktion)
 - 2 System-Reset
 - 3 Roboter ein-/ausschalten, nachdem das Gamepad mit (15) aktiviert wurde
 - 4 Balance-Mode ein-/ ausschalten
 - 5 Daumenjoystick rechts (auf der Stelle drehen, im Rotate- und Translate-Mode drehen und Höhe des Roboters ändern)
 - 6 Gamepad ein-/ausschalten
 - 7 Daumenjoystick links (vorwärts/rückwärts, im Rotate- und Translate-Mode Body vor/zurück sowie seitliche Bewegung)
 - 8 D-Pad auf/ab stellt die Höhe des Roboters ein (auf = höher, ab = niedriger)
 - 9 D-Pad links/rechts stellt die Laufgeschwindigkeit ein (links = langsamer, rechts = schneller)
 - 10 Obere Taste = Translate-Mode
 - 11 Untere Taste = Rotate-Mode
 - 12 Die LEDs zeigen die Stromversorgung und die Verbindung zum Empfänger an
 - 13 Gangart auswählen (Tripod-6, Tripod-8, Tripple-12, Tripple-16, Ripple-12, Wave-24). Bei jedem erneuten Betätigen des Tasters wird die Gangart, wie in der Liste beschrieben, durchgeschaltet. Beginnt die Liste von vorne (Tripod-6), so wird dies mit einer anderen Tonfolge durch das Hexapod Roboter-Board signalisiert
 - 14 Ohne Funktion
 - 15 Gamepad aktivieren
 - 16 Aktiviert doppelte Schritthöhe
 - 17 Aktiviert doppelte Schrittweite
- Alle Taster mit Toggle-Funktion (erneutes Betätigen wechselt den Zustand) lassen das Hexapod Roboter-Board in unterschiedlicher Tonfolge piepen. Drücken Sie die Taste zum ersten Mal, so quittiert das Hexapod Roboter-Board dies mit einem Piepton und beim erneuten Betätigen (Urzustand) mit einer Tonfolge. So können Sie leicht herausfinden, in welchem Zustand sich der Roboter befindet. Nach Ertönen der Tonfolge ist der immer der Urzustand (Startzustand) aktiv.

21. Beine kalibrieren

Um die Beine genau einzustellen, verwenden Sie das Programm „Terminal“, welches Sie im Ordner „Terminal“ im entpackten Download-Bundle finden. Das Ziel ist es, die Beine auf eine genaue Grundkonfiguration einzustellen. Beim Zusammenbau der Mechanik werden Sie die Beine aufgrund der Zahnung der Servohörner und der mechanischen Toleranzen nicht zu 100% in die Grundkonfiguration bringen können. Mit Hilfe des Tools können Sie diese Fehler ausgleichen.

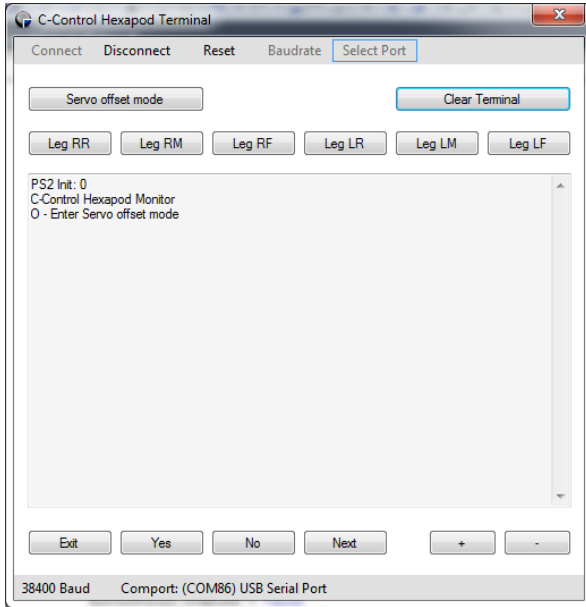


Bild 33

- Starten Sie das Programm mit der Datei „Terminal.exe“ und führen Sie die Kalibrierung durch. Stellen Sie den Roboter zuvor auf eine Erhöhung, so dass sich die Beine frei in der Luft bewegen können. Eine stabile Schachtel kann z.B. als Erhöhung dienen.
- Verbinden Sie das Hexapod Roboter-Board (PRG-M Anschluss) mit Ihrem Computer, wie bereits beim Aufspielen der Firmware durchgeführt.
- Schließen Sie den Akku am Hexapod Roboter-Board an.
- Starten Sie das Programm „Terminal.exe“.
- Wählen Sie jetzt den seriellen Anschluss aus (gleicher Anschluss wie bei der Programmierung). Klicken Sie dazu auf „Select Port“. Falls mehr als ein Gerät angezeigt wird, so stecken Sie die Hauptplatine einfach kurz aus und wieder am USB-Anschluss an. Das Gerät, welches dann kurz verschwunden war, ist das richtige.

- Wählen Sie unter „Baudrate“ 38400 Baud aus.
- Klicken Sie auf „Connect“. Der Roboter meldet im Terminalfenster, wie in Bild 33 gezeigt.
- Klicken Sie auf „Servo offset mode“, um in den Kalibriermodus zu gelangen.
- Der Roboter „wackelt“ nun mit dem RC-Servo, das zum Einstellen ausgewählt ist.
- Durch mehrmaliges Klicken auf „+“ und „-“ können Sie in kleinen Schritten das ausgewählte RC-Servo exakt einstellen.
- Haben Sie das RC-Servo fertig eingestellt, so können Sie mit einem Klick auf „Next“ das nächste RC-Servo einstellen, bis Sie alle 18 Servos eingestellt haben. Sie können auch mit einem Klick auf die Schaltflächen „Leg RR“, „Leg RM“, „Leg RF“, „Leg LR“, „Leg LM“ und „Leg LF“ gezielt ein Bein auswählen.
- Sind alle RC-Servos eingestellt, so klicken Sie auf „Exit“. Das Programm fragt nun, ob die Einstellungen gespeichert werden sollen. Klicken Sie auf „Yes“ um die Einstellungen zu speichern oder „No“ um die Einstellungen zu verwerfen.
- Mit „Clear Terminal“ können Sie den Inhalt des Terminalfensters löschen.

MAC und Linux

Sofern Sie einen MAC oder Linux benutzen, können Sie das Terminal nicht nutzen. Sie können jedoch die Beine trotzdem einstellen, indem Sie ein anderes Terminal-Programm Ihrer Wahl nutzen. Am einfachsten ist es, hier das Arduino Terminal zu nutzen. Dieses können Sie in der Arduino IDE aufrufen.

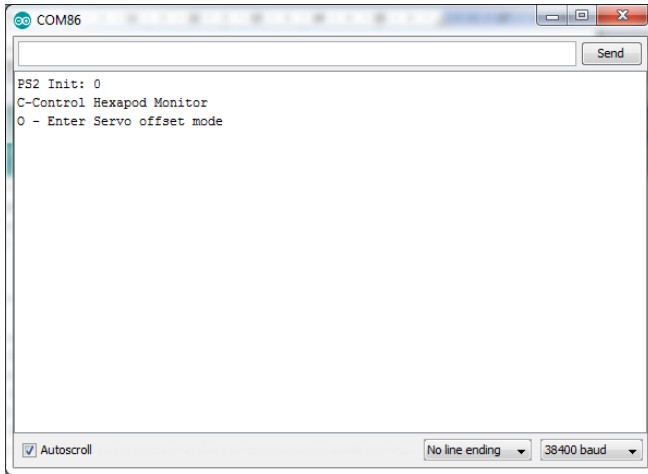


Bild 34

Die Abbildung zeigt das Arduino-Terminal, nachdem sich das Terminal mit dem Hexapod Roboter-Board verbunden hat.

Gehen Sie wie folgt vor:

- Verbinden Sie das Hexapod Roboter-Board (PRG-M Anschluss) mit Ihrem Computer, wie bereits beim Aufspielen der Firmware durchgeführt.
- Schließen Sie den Akku am Hexapod Roboter-Board an.
- Starten Sie das Terminal-Programm.
- Sofern Sie ein anderes Terminal verwenden als das Arduino-Terminal, wählen Sie den seriellen Anschluss aus (gleicher Anschluss wie bei der Programmierung).
- Wählen Sie als Baudrate 38400 Baud aus. Kein LF und CR bzw. im Arduino-Terminal „No line encoding“.
- Verbinden Sie das Terminal mit dem Roboter. Der Roboter meldet im Terminalfenster, wie in der Abbildung gezeigt. Das Arduino-Terminal verbindet sich sofort nach dem Öffnen!
- Drücken Sie einmal den Taster „RESET-M“ auf dem Hexapod Roboter-Board. Dies ist bei der Verwendung des Arduino-Terminals nicht nötig.
- Im Terminalfenster erscheint folgender Text.

```
PS2 Init: 0  
C-Control Hexapod Monitor  
0 - Enter Servo offset mode
```

- Senden Sie ein „O“ an das Hexapod Roboter-Board, indem Sie „O“ in die Sendezelle schreiben und danach Enter drücken. Der Roboter „wackelt“ nun mit dem RC-Servo, das zum Einstellen ausgewählt ist. Die Zeichen können klein oder groß geschrieben werden („o“ als auch „O“ ist möglich). Danach erscheint im Terminal folgende Meldung:

```
Serial Cmd Line:o<eol>
Find Servo Zeros.
$-Exit, +- changes, *-change servo
  0-5 Chooses a leg, C-Coxa, F-Femur, T-Tibia
Servo: RR Coxa(0)
```

Mit „\$“ verlassen Sie die Kalibrierung -> Exit

Mit „+“ oder „-“ können Sie in kleinen Schritten das ausgewählte RC-Servo exakt einstellen. Die Zeichen müssen so oft an das Hexapod Roboter-Board gesendet werden, bis das Bein richtig justiert ist.

Mit „*“ wählen Sie das nächste RC-Servo aus.

Mit einer Zahl von 0 bis 5 können Sie Beine einzeln anwählen.

0 = Leg RR

1 = Leg RM

2 = Leg RF

3 = Leg LR

4 = Leg LM

5 = Leg LF

- Sind alle RC-Servos eingestellt, senden Sie ein „\$“ Zeichen. Das Programm fragt nun, ob die Daten gespeichert werden sollen. Senden Sie „y“ für speichern oder „n“, um die Einstellungen zu verwerfen.

Wenn alle Beine fertig kalibriert sind, können Sie den Roboter mit dem Gamepad steuern. Im Gamepad-Modus ist der Roboter wie ein RC-Modell zu betrachten, das mit einer Funkfernsteuerung ferngesteuert wird.

Möchten Sie jedoch mehr aus Ihrem Hexapod-Roboter herausholen, so lesen Sie das nächste Kapitel über die Nutzung der User-Boards durch. Damit haben Sie die einfache Möglichkeit, eigene Programme zum Steuern des Roboters mit Hilfe der C-Control Hexapod-Bibliothek zu schreiben.

- Es kann sein, dass Sie diesen Vorgang nach einiger Zeit nochmals durchführen müssen. Grund hierfür ist, dass die RC-Servos nach einer gewissen Betriebszeit ein größeres Getriebespiel bekommen und sich die erste Kalibrierung nicht mehr als ideal erweist.

22. Verwenden von User-Boards

Das Hexapod Roboter-Board kann mit verschiedenen User-Boards ausgestattet werden, um die Funktionalität zu erweitern. Mögliche User-Boards sind z.B. Arduino UNO, NodeMCU oder kompatible Boards. Diese können direkt auf das Roboter-Board aufgesteckt werden.

Mit der im nächsten Kapitel beschriebenen Software-Bibliothek können Sie vom User-Board an den Locomotion-Controller Steuerbefehle senden oder Werte vom Locomotion-Controller lesen. Dazu aber später mehr im nächsten Kapitel „Demo Programme“.

Arduino UNO

Stecken Sie ein „Arduino UNO“ Board auf das Hexapod Roboter-Board wie in der folgenden Abbildung zu erkennen. Gehen Sie dabei vorsichtig vor und achten Sie darauf, dass beim Aufstecken die längeren Stiftleisten, welche in die Buchsen des „Arduino UNO“ Boards gesteckt werden, nicht verbogen werden. Achten Sie auch darauf, dass alle Stiftleisten-Pins in die Buchsen des „Arduino UNO“ Boards eingesteckt wurden.

Nach dem Aufstecken des Boards können Sie direkt zum Kapitel „Demo Programme“ springen.

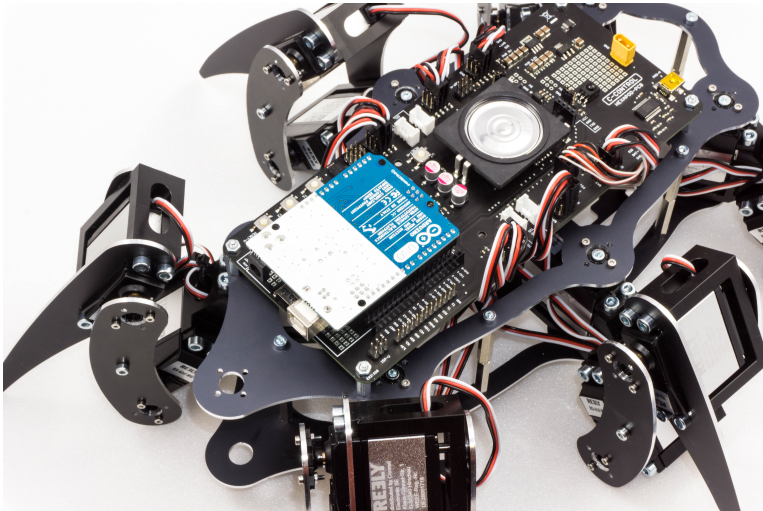


Bild 35

Die Abbildung zeigt ein aufgestecktes „Arduino UNO“ Board. Es ist möglich, fast alle Arduino UNO kompatiblen Boards zu verwenden.

NodeMCU WiFi Board

Das WiFi-Board mit der Bezeichnung „NodeMCU“ und dem verwendeten ESP8266 WiFi Chip kann auf dem inneren Steckplatz als User-Board verwendet werden.

Wenn Sie das Hexapod Roboter-Board mit WiFi erweitern möchten, können Sie dies ganz leicht mit Hilfe dieses User-Boards tun. In den Beispielen sind bereits Programme für das NodeMCU Board enthalten, die die Steuerung von einem Webbrowser aus zeigen.

Stecken Sie ein NodeMCU Board auf das Hexapod Roboter-Board wie in der folgenden Abbildung zu erkennen. Gehen Sie dabei vorsichtig vor und achten Sie darauf, dass beim Aufstecken die Stiftleisten nicht verbogen werden und alle Pins richtig eingesteckt sind.

Nach dem Aufstecken des Boards können Sie direkt zum Kapitel „Demo Programme“ springen.

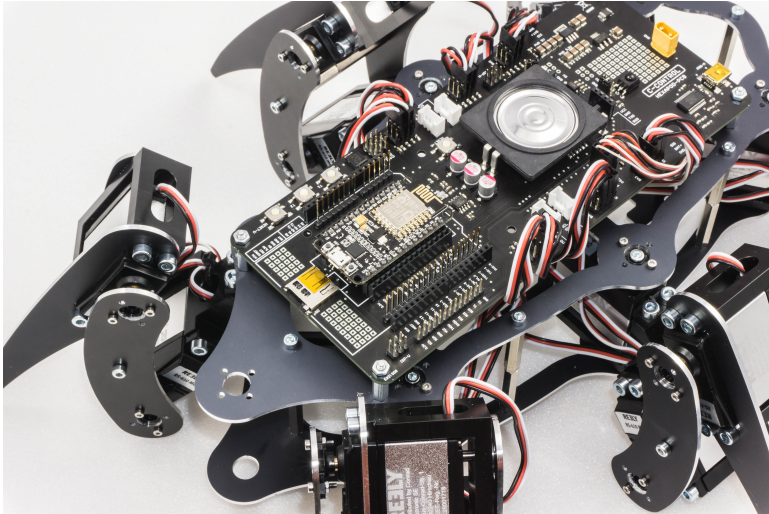


Bild 36

Die Abbildung zeigt ein aufgestecktes NodeMCU Board.



Achten Sie beim Kauf darauf, dass der Abstand der Stiftleisten stimmt! Es gibt zwei verschiedene Versionen auf dem Markt. Eine besitzt als USB-Chip einen Silabs CP2102, die andere Version einen CH340G. Die Version mit dem CH340G hat einen breiteren Pinabstand und passt nicht in das Hexapod Roboter-Board! Das zu verwendende NodeMCU Board muss einen Stiftleistenabstand von 23 mm besitzen.

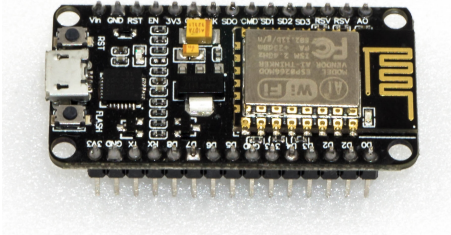


Bild 37

Single-Board-Computer (SBCs) und andere Boards

Das Hexapod Roboter-Board bietet die Möglichkeit über Platinenabstandshalter einen SBC wie z.B. einen Raspberry Pi 2/3 oder kompatible zu montieren. SBC mit einem anderen Lochabstand lassen sich mit einer selbstgebauten Adapterplatine leicht montieren. Hier bietet das Hexapod Roboter-Board genügend Spielraum für eigene Erweiterungen.

Die abgebildete Stiftleiste neben dem „ISP-U“ Anschluss ist ein universeller Anschlussport für SBC oder eigene Erweiterungen. Verwenden Sie für die Verdrahtung zwischen dem SBC und dem Hexapod Roboter-Board am besten sogenannte „Jumpwire“.

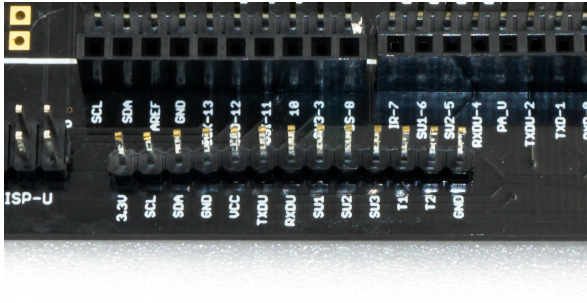


Bild 38

Pinbeschreibung:

3,3V	3,3 V/DC Versorgungseingang für die internen Pegelwandler des Hexapod Roboter-Boards (hier 3,3 V vom SBC anschließen). Verwenden Sie ein Board mit 5 V Pegel, dann schließen Sie hier 5 V statt den 3,3 V an. Hier die jeweilige Versorgungsspannung anschließen!
SCL	I ² C Taktleitung (hier SCL vom SBC anschließen, der I ² C-Bus steht dann an den Buchsen J2 und J3 zur Verfügung).
SDA	I ² C Datenleitung (hier SCL vom SBC anschließen, der I ² C-Bus steht dann an den Buchsen J2 und J3 zur Verfügung).
GND	Ground (Minuspol).
VCC	Ausgang +5 V/DC, 1000 mA vom Spannungsregler
TXDU	UART Sendeleitung vom Locomotion-Controller (hier die UART des SBC anschließen).
RXDU	UART Empfangsleitung vom Locomotion-Controller (hier die UART des SBC anschließen).
SU1 - SU3	Diese Anschlüsse führen direkt zu der Stiftleiste SU1, SU2, SU2. Diese Pins können z.B. für RC-Servos oder digitale Signale frei verwendet werden.
T1 und T2	Diese Pins sind mit den Tastern T1 und T2 verbunden. Die Taster sind extern mit Pull-Up Widerständen versehen und liefern ein High-Signal, wenn diese nicht betätigt sind. Die Pull-Up Widerstände können mit den Platinen-Jumpern SJ6 und SJ7 deaktiviert werden.
GND	Ground (Minuspol).

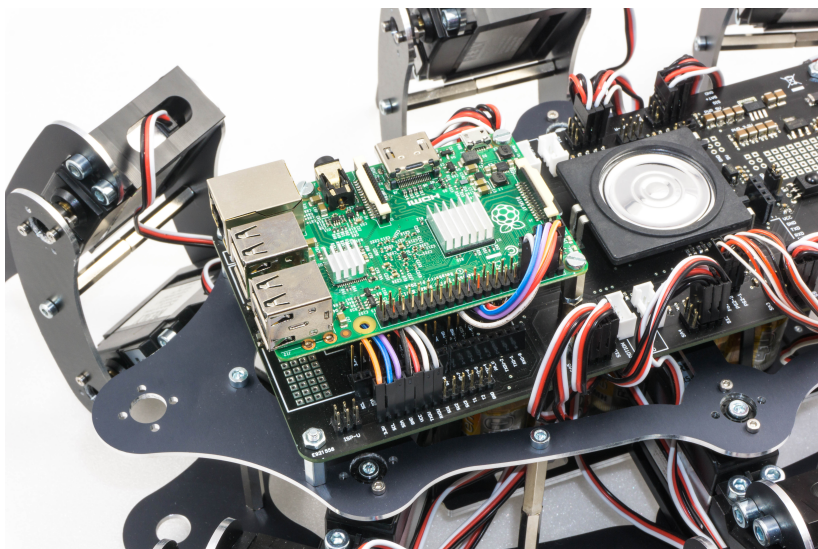


Bild 39

Die Abbildung zeigt den Muster-Hexapoden und einen auf 3 Platinen-Abstandshaltern mit 25 mm Länge montierten Raspberry Pi 2. Die Bohrungen zur Befestigung sind bereits im Hexapod Roboter-Board vorhanden. Die Verbindung zwischen Raspberry Pi und Hexapod Roboter-Board wurde mittels Steckbrücken (Jumpwire) hergestellt. Die Kabel der Steckbrücken wurden unter dem Raspberry Pi untergebracht, um eine saubere Verdrahtung zu erhalten.

- Es kann vorkommen, dass der SBC sehr empfindlich auf Spannungsschwankungen reagiert und ein sicherer Betrieb über den Akku, der auch die Bein-Servos versorgt, nicht möglich ist. Wir empfehlen in diesem Fall den SBC über eine kleine separate Powerbank zu versorgen. Hier reicht in der Praxis eine Akkukapazität der Powerbank von 2000 - 3000 mAh aus, um den SBC so lange zu versorgen, bis der Hexapod-Akku leer ist. Teils reicht es auch aus, die Stromversorgung direkt am SBC mit einem großen Kondensator 2200 μF (oder mehr) zu stützen. Grund für den instabilen Betrieb kann ein zu kleiner Akku mit hohem Innenwiderstand sein, oder RC-Servos mit sehr hohem Strombedarf bzw. großen Stromspitzen.

23. Demo-Programme

Um die User-Boards einfach und komfortabel zu verwenden, nutzen Sie die C-Control Hexapod Bibliothek. Diese wurde bereits am Anfang dieser Anleitung installiert. Die mitgelieferten Beispiele zeigen die Verwendung der User-Boards und die Funktionalität der Arduino Hexapod Bibliothek.

➔ Bei der Verwendung von User-Boards, welche nicht Arduino kompatibel sind (wie z.B. SBC) sehen Sie sich die „Hexapod_Lib.h“ Datei an. In dieser wird gezeigt, wie die Kommunikation zwischen dem User-Board und dem Locomotion-Controller funktioniert. Diese soll Ihnen als Beispiel zum Entwickeln einer eigenen Funktionsbibliothek dienen. Entpacken Sie dazu die ZIP-Datei „C-Control-Hexapod.zip“ aus dem Download-Bundle und sehen Sie sich den Quellcode der Bibliothek in einem Editor wie Notepad oder einer Entwicklungsumgebung Ihrer Wahl an.



Die mitgelieferten Beispiele sind für „Arduino UNO“ und „NodeMCU“ Boards, die mit der Arduino IDE programmiert werden, ausgelegt.

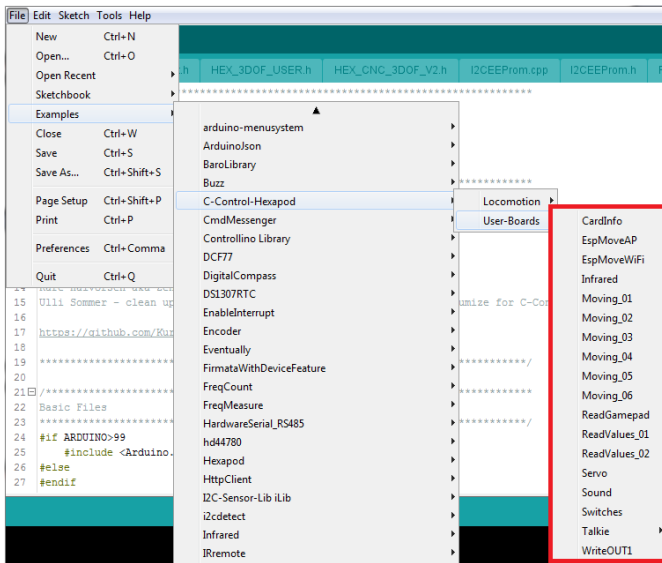


Bild 40

Die Arduino-Beispiele zeigen, wie die Bibliothek zu verwenden ist. Diese finden Sie in der Arduino IDE unter „Examples\C-Control-Hexapod...“. Sehen Sie sich in Ruhe alle Beispiele an und testen Sie diese auf Ihrem Hexapoden. Wenn Sie die Beispiele verstanden haben, können Sie loslegen und eigene Programme damit erstellen. Sehen Sie sich dazu auch die „Hexapod_Lib.h“ genauer an. Diese können Sie mit etwas Erfahrung sehr leicht mit eigenen Funktionen erweitern.

Kurzbeschreibung der mitgelieferten Beispiele (Locomotion):

Im Ordner „Locomotion“ finden Sie die Beispiele für den Locomotion-Controller (Arduino MEGA2560).

LEDs

Kleines Testprogramm, das die beiden LEDs (User und Live) abwechselnd blinken lässt.

Motion-Firmware

Firmware des Locomotion-Controllers.

ServoCenter

Dieses Programm bringt die RC-Servos in die Mittelstellung. Wird zum mechanischen Zusammenbau des Hexapod-Roboters benötigt.

Kurzbeschreibung der mitgelieferten Beispiele (User-Boards):

Im Ordner User-Boards finden Sie die Beispiele für Arduino UNO und NodeMCU bzw. compatible Boards.



Stellen Sie vor der Verwendung die Arduino IDE auf das richtige Arduino-Board um! Lesen Sie auch die Informationen in den Kommentaren zu den jeweiligen Beispielen!

CardInfo

→ Dieses Beispiel ist nur für Arduino UNO bzw. compatible Boards!

Sie können mit diesem Beispiel die Funktion des μ SD-Karten Lesers prüfen. Stecken Sie dazu eine MicroSD-Speicherkarte mit willkürlichem Inhalt in den Kartenslot und starten Sie das Beispiel. Im Terminal wird Ihnen der Inhalt der Speicherkarte angezeigt (Dateinamen). Es funktionieren alle original Arduino SD-Karten Beispiele. Es muss dabei nur der Pin für „CS“ (Chipselect) wie in diesem Beispiel abgeändert werden. Die Terminal-Baudrate beträgt 19200 Baud.

EspMoveAP

→ Dieses Beispiel ist nur für NodeMCU Boards!

Das Beispiel zeigt die Steuerung über einen HTTP-Server über eine direkte WiFi-Verbindung zum NodeMCU Board. Das Programm baut einen Access-Point auf, mit dem Sie sich z.B. mit einem Smartphone verbinden können. Die vergebene IP-Adresse wird im Terminal ausgegeben. Diese tippen Sie in Ihren Browser ein. Danach lädt dieser die Weboberfläche zum Steuern des Roboters. Sie können dann durch Antippen oder Klicken der Schaltflächen den Roboter über WiFi steuern. Das Beispiel kann leicht von Ihnen auf weitere Funktionalitäten erweitert werden. Die Terminal-Baudrate beträgt 115200 Baud.

EspMoveWiFi

→ Dieses Beispiel ist nur für NodeMCU Boards!

Das Beispiel zeigt die Steuerung über einen Webbrowser über eine WiFi-Verbindung zum NodeMCU Board. Das Programm baut eine Verbindung zu Ihrem Router auf und der Roboter ist dann innerhalb Ihres Netzwerkes erreichbar. Die vergebene IP-Adresse wird im Terminal ausgegeben. Diese tippen Sie in Ihren Browser ein. Danach lädt dieser die Weboberfläche zum Steuern des Roboters. Sie können dann durch Antippen oder Klicken der Schaltflächen den Roboter über WiFi steuern. Das Beispiel kann leicht von Ihnen auf weitere Funktionalitäten erweitert werden. Die Terminal-Baudrate beträgt 115200 Baud.

Infrared

→ Dieses Beispiel ist nur für Arduino UNO bzw. kompatible Boards!

Das Beispiel liest einen RC5 Infrarot-Code aus und gibt die Werte am Terminal aus. Stecken Sie den Jumper J8 dazu in Richtung der Beschriftung „IR-U“. Als IR-Sender kann eine Universalfernbedienung dienen, die auf RC5 eingestellt wurde (viele Philips-Geräte arbeiten mit RC5), welche für TV und Audiogeräte im Handel erhältlich sind. Die Terminal-Baudrate beträgt 19200 Baud.

Moving_01

Dieses Beispiel zeigt, wie Daten ohne die „Hexapod Bibliothek“ an den Locomotion-Controller gesendet werden. Dieses Beispiel eignet sich gut als Grundlage für die Verwendung von SBC. Es sollte jedoch bereits Erfahrung in der Programmierung vorhanden sein! Sehen Sie sich dazu die Funktion `SendData()` genauer an. Hier sehen Sie, wie Daten vom User-Board an den Locomotion-Controller gesendet werden.

Moving_02

Dieses Beispiel ist ideal für den Einstieg in die Programmierung der User-Boards und zeigt die Grundfunktionen, um den Hexapod mit Hilfe der Bibliothek zu steuern.

Moving_03

Dieses Beispiel zeigt die Verwendung der `ROBOT_MOVE()` Funktion.

Moving_04

Dieses Beispiel zeigt die Verwendung der `ROBOT_ROTATE_MODE()` und `ROBOT_TRANSLATE_MODE()` Funktion.

Moving_05

Dieses Beispiel zeigt die Verwendung der `ROBOT_SINGLE_LEG()` Funktion.

Moving_06

Dieses Beispiel zeigt, wie ein User-Programm mit Hilfe des Gamepads unterbrochen werden kann. Verwenden Sie bei diesem Beispiel das Terminal, um die Meldungen vom Roboter zu erhalten. Stellen Sie dazu den Roboter auf eine Erhöhung, so dass sich die Beine in der Luft bewegen, ohne den Boden zu berühren. So haben Sie die Möglichkeit, die Ausgaben beim Steuern des Roboters auf dem Terminal zu beobachten. Die Terminal-Baudrate beträgt 115200 Baud.

ReadGamepad

Das Beispiel zeigt, wie Sie die Steuerdaten des Gamepads vom Locomotion-Controller lesen können. Wenn Sie den Jumper „J7“ aufstecken, reagiert der Locomotion-Controller nicht mehr auf das Gamepad. Es ist dann möglich, die Daten nur auf dem User-Board zu verwenden und eigene Steuerungen zu realisieren. Die Terminal-Baudrate beträgt 115200 Baud.

ReadValues_01

Das Beispiel zeigt, wie Sie Daten zwischen Locomotion-Controller und User-Board ohne Bibliothek senden und einlesen können. Dieses Beispiel eignet sich gut als Grundlage für die Verwendung von SBC. Es sollte jedoch bereits Erfahrung in der Programmierung vorhanden sein! Die Terminal-Baudrate beträgt 115200 Baud.

ReadValues02

Dieses Beispiel zeigt, wie Daten vom Locomotion-Controller mit Hilfe der Bibliothek eingelesen werden. Die Terminal-Baudrate beträgt 115200 Baud.

Servo

Das Beispiel zeigt die Verwendung von zusätzlichen RC-Servos, welche über das User-Board gesteuert werden. Es werden die Anschlüsse SU1 bis SU3 dazu verwendet. Es ist auch möglich, die Anschlüsse SU1 bis SU3 für andere Komponenten wie Sensoren, Taster, Schalter etc. zu verwenden. Die Terminal-Baudrate beträgt 115200 Baud.

Sound

Das Beispiel zeigt eine einfache Tonausgabe über den Lautsprecher.

Switches

Das Beispiel zeigt wie, die User-Taster T1 und T2 abgefragt werden.

Talkie

Talkie zeigt, wie eine Sprachausgabe mit Hilfe der Talkie-Bibliothek realisiert werden kann. Talkie ist eine Implementierung des Texas Instruments Sprachsynthesizers ICs auf einem Arduino Board. Dieses IC wurde in den frühen 80igern sehr oft verbaut. Man fand diese Sprachausgabe oft in teuren Uhren, Spiele-Computern, Lerncomputern und vielen anderen Geräten. Der unverwechselbare Sound aus den 80igern ist damit auch auf dem Hexapod Roboter-Board verfügbar. Es sind in den Beispielen über 1000 vorgefertigte Phrasen enthalten, die zu Sätzen zusammgebaut werden können. Sollte Ihnen die Sprachausgabe zu leise sein, so können Sie die Lautstärke am Trimmer „Vol.“ auf dem Hexapod Roboter-Board erhöhen. Zudem können Sie noch eine kleine „Lautsprecherbox“ für den Lautsprecher bauen, dies erhöht die Lautstärke deutlich! Eine einfache kleine Schachtel kann hier wahre Wunder bewirken.

WriteOUT1

Dieses Beispiel zeigt, wie Sie die Aus-/Eingänge IN1 und OUT1 des Locomotion-Controllers lesen und schreiben können. Sie erhalten somit einen zusätzlichen Ein- und Ausgang zum User-Board. Die Pins sind direkt mit dem Locomotion-Controller verbunden und können maximal 20 mA Strom liefern. Die maximale Eingangsspannung für den Eingang beträgt 5 V/DC!

24. FAQ

Fehler beim Kompilieren der Beispiele

Sollte es zu Problemen beim Kompilieren der Firmware kommen: Die zur Verfügung gestellte Firmware sowie die Beispiele wurde mit der Arduino Version 1.8.3 entwickelt. Alternativ laden Sie diese unter www.arduino.cc herunter!

Tonausgabe zu leise

Prüfen Sie die Einstellung des Trimmers „Vol.“. Bauen Sie sich einen kleine „Lautsprecherbox“ zum Hexapod Roboter-Board Lautsprecher. Diese erhöht die Lautstärke und verbessert den Klang erheblich.

Es kommt keine Kommunikation mit SPI Geräten und dem User-Board zustande

Setzen Sie den Chip Select (CS) Pin des User-Boards auf „High“ und überprüfen Sie Ihre Anschlüsse am ISP-U Anschluss und ob die verwendete SPI Komponente mit Strom versorgt wird.

```
const int chipSelect = 8;
pinMode(chipSelect, OUTPUT);
digitalWrite(chipSelect, HIGH);
```

Das Gampad verbindet sich nicht

- Überprüfen Sie die Akkus im Sender.
- Ist der Empfänger richtig am Hexapod Roboter-Board angeschlossen?
- Ist die „Motion-Firmware“ auf das Hexapod Roboter-Board aufgespielt?
- Trennen Sie die Stromversorgung des Hexapod Roboter-Boards, schalten Sie das Gampad ein und stellen danach die Verbindung zur Stromversorgung des Hexapoden wieder her (Akku anstecken).
- Ist das Gamepad PS2-kompatibel?

Der Roboter reagiert nicht auf das Gamepad

- Überprüfen Sie die Akkus im Sender.
- Ist der Empfänger richtig am Hexapod Roboter-Board angeschlossen?
- Ist die „Motion-Firmware“ auf den Hexapod Roboter-Board aufgespielt?
- Trennen Sie die Stromversorgung des Hexapod Roboter-Boards, schalten Sie das Gampad ein und stellen danach die Verbindung zur Stromversorgung des Hexapoden wieder her (Akku anstecken).
- Ist das Gamepad PS2-kompatibel?
- Ist der Jumper J7 abgezogen?

Der Roboter läuft nicht sauber

- Sind alle Parameter richtig eingestellt?
- Wurden die Beine kalibriert?
- Sind die Akkus ausreichend stark und geladen?
- Sind die verbauten RC-Servos ausreichend stark?
- Ist der Roboter zu schwer?
- Sind die RC-Servos vor dem Einbau in Mittelstellung gestellt worden?
- Rutschen die Beine auf dem Untergrund?
- Sind alle Schrauben festgezogen?
- Wackelt die Mechanik?
- Ist die Mechanik zu schwergängig?

Wo finde ich den Schaltplan?

Der Schaltplan befindet sich im Ordner „Schematic“ im Download-Bundle bzw. kann unter www.conrad.com heruntergeladen werden.

Der Roboter führt nach dem Einschalten bzw. aufstehen einen Reset durch.

- Vergewissern Sie sich, dass der voll Akku geladen ist.
- Ist das Anschlusskabel des Akkus zu dünn ($< 1,5 \text{ mm}^2$) oder länger als 20 cm?
- Ist der Akkustecker sauber angelötet?
- Ist der Akku zu klein gewählt, dann kann er den nötigen Betriebsstrom für die RC-Servos nicht liefern.
- Blockiert ein oder mehrere Servos?
- Bewegt sich die Mechanik zu schwergängig?

25. Pinzuweisung der User-Boards

Locomotion-Controller (ATmega2560)

	Port	Arduino
S0	PB6	12
S1	PE3	5
S2	PE4	2
S3	PE5	3
S4	PH0	17
S5	PH1	16
S6	PH3	6
S7	PH4	7
S8	PH5	8
S9	PA3	25
S10	PA4	26
S11	PA5	27
S12	PC4	33
S13	PC3	34
S14	PC2	35
S15	PC1	36
S16	PC0	37
S17	PG1	40
SA0	PF0	A0

	Port	Arduino
SA1	PF1	A1
SA2	PF2	A2
SA3	PF4	A4
SA4	PF5	A5
SA5	PF6	A6
PS2_DAT	PA6	28
PS2_CMD	PC7	30
PS2_SEL	PC5	32
PS2_CLK	PC6	31
IR	PJ0	15
UB-MESS	PF3	A3
SOUND	PL3	46
Jumper J7	PG5	4
IN1	PA2	24
OUT1	PA1	23
RxD-U	PDs/RxD1	Serial1
TxD-U	PD3/TxD1	Serial1
LIVE-LED	PA0	22
USER-LED	PK7	69

User-Board Arduino

	Port	Arduino
T1	ADC2	A2
T2	ADC3	A3
SCK	PB5	13
MISO	PB4	12
MOSI	PB3	11
SS_SD	PB0	8
PA	PD3	3
IR-DAT	PD7	7
SU1	PD6	6
SU2	PD5	5
SU3	PB1	9
RxD_U	PD4	4
TxD_U	PD2	2

User-Board NodeMCU

	Port	Arduino
T1		10
T2	ADC0	A0
PA	D5	GPIO14
SU1	D6	GPIO12
SU2	D7	GPIO13
SU3	D8	GPIO15
RxD_U	D3	GPIO0
TxD_U	D4	GPIO2

26. Reinigung

Äußerlich sollte das Hexapod Roboter-Board nur mit einem weichen, trockenen Tuch oder Pinsel gereinigt werden. Verwenden Sie auf keinen Fall aggressive Reinigungsmittel oder chemische Lösungen, da sonst die Platine beschädigt werden könnte.

Schützen Sie die Platine vor Schmutz und Feuchtigkeit.

27. Entsorgung



Elektronische Geräte sind Wertstoffe und gehören nicht in den Hausmüll. Entsorgen Sie das Produkt am Ende seiner Lebensdauer gemäß den geltenden gesetzlichen Bestimmungen.

28. Technische Daten

Betriebsspannung.....	4,5 - 10 V/DC
Stromaufnahme nur Roboter-Board	ca. 80 mA
Anzahl der Servoanschlüsse	18 Locomotion-Controller + 3 User-Board
Programmierung.....	Arduino IDE (C/C++)
Programmierschnittstelle	USB
User-Board Steckplätze.....	3
I2C-Anschlüsse Locomotion-Controller	2
I2C-Anschlüsse User-Board	2
IR-Empfänger (*)	1x 38 kHz
Audio Verstärker mit Lautsprecher	1
MicroSD Kartensteckplatz (*)	1
Locomotion-Controller	1 (Atmel ATmega2560 - Arduino MEGA kompatibel)
Reset-Taster Locomotion-Controller.....	1
Reset-Taster User-Board.....	1
ISP-Anschluss Locomotion-Controller.....	1
ISP-Anschluss User-Board.....	1
PS2 kompatibler Gamepad Anschluss	1
Stromversorgung intern	3,3 V/DC und 5 V/DC
Strombelastbarkeit 3,3 V/DC	400 mA
Strombelastbarkeit 5 V/DC	1000 mA
Abmessung (L x B x H).....	200 x 88 x 23 mm
Platinen-Gewicht nur Platine	ca. 100 g
Arbeits-Temperaturbereich	0 °C bis +40 °C

(*) Nur verwendbar in Verbindung des Arduino UNO Steckplatz oder kompatiblen Boards!

Ⓓ Dies ist eine Publikation der Conrad Electronic SE, Klaus-Conrad-Str. 1, D-92240 Hirschau (www.conrad.com).

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktionen jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers. Nachdruck, auch auszugsweise, verboten. Die Publikation entspricht dem technischen Stand bei Drucklegung.

Copyright 2018 by Conrad Electronic SE.