

# Altes Wiki

- [Elektronik Grundlagen](#)
- [Caeser \(Basestation\) v2](#)
- [Bossix \(Mainboard\) v2](#)
- [Asterix](#)
- [PCBS](#)
- [PCB BOM](#)

# Elektronik Grundlagen

Auf dieser Seite sollen ein paar Grundlagen zur Entwicklung von Elektronik im und um den Roboter gesammelt werden. Diese Seite ist noch im Aufbau.

## Schematic

In der Schematic werden die verwendeten Bauteile und deren Verbindungen festgelegt. Jedes Bauteil wird dabei durch ein Symbol dargestellt. Die Verbindungen zwischen den Symbolen werden durch Linien, welche als Wire bezeichnet werden, dargestellt. Da direkte Verbindungen nicht immer möglich sind und zu einem unübersichtlichen Plan führen können, können die Kabel mit einem Label gekennzeichnet werden. Das Label gibt dem Kabel einen Namen und verbindet alle Kabel mit dem gleichem Namen.

## Layout

Im Layout wird das PCB erstellt. Die Umrandung, sowie der Aufbau, können angegeben werden. Jeder Komponente wird ein Footprint zugewiesen. In einem ersten Schritt werden die Komponenten auf dem Board platziert ohne dabei Verbindungen herzustellen. Dabei wird darauf geachtet, dass Verbindungen möglichst gut gelegt werden können. Wenn alle Komponenten auf dem PCB platziert sind, können die Verbindungen hergestellt werden. Die Platzierung kann auch dabei noch angepasst werden, falls etwas nicht passt.

## Kicad

KiCad ist das von uns genutzte CAD (Computer Aided Design) Programm zur Erstellung von PCBs. Wir haben KiCad gewählt, da es Open Source ist, auf jeder Plattform läuft (Windows, Mac, Linux) und alle, für die LuBots wichtigen, Features unterstützt.

Alternativen:

- Altium Designer
  - Industrie Standard
  - Für Studenten kostenlos nutzbar

- Nur Windows
- **Eagle**
  - Integration mit Fusion360 (Hardware)
  - Für hobby sehr eingeschränkt kostenlos nutzbar
  - Für Studenten weniger eingeschränkt kostenlos nutzbar
  - Manche Footprints stimmen nicht

KiCad besteht aus mehreren Komponenten. Die wichtigsten sind der Schematic Editor und der PCB Editor. In diesen können die Schematic und das Layout gemacht werden. Begleitend gibt es einen Symbol Editor und einen Footprint Editor, womit Symbole für die Schematic und Footprints für das Layout erstellt und verändert werden können. Weitere nützliche Komponenten sind der Gerber Viewer, der Image Converter und die Calculator Tools. Der Gerber Viewer kann erstellte fertigungsdateien (gerber format) visualisieren, wodurch diese überprüft werden können. Mit dem Image Converter können beliebige Grafiken in Footprints und Symbole umgewandelt werden. Die Calculator Tools beinhalten eine Reihe von nützlichen Rechnern zum Beispiel: E-Series calculator, Transmission Line calculator, Via Size calculator und Track Width calculator. Die umrandung von Schematics kan mit dem Drawing Sheet Editor angepasst werden und zu guter letzt gibt es noch den Plugin and Content Manager. Dieser bietet die möglichkeit externe Plugins, Bibliotheken und Themes zu installieren.

Die Benutzung von kiCad ist [hier](#) genauer erläutert.

# Caesar (Basestation) v2

Auf dem Robocup in Bangkok haben wir gemerkt, dass der Funk miserabel war (Teilweise über 0.5s ohne Packetempfang am Roboter und Kein Packetempfang an Caesar). Daher wurden ein neuer Transceiver und Verstärker gewählt. Zusätzlich soll Version 2 autonom nutzbar sein und über Ethernet auch die Daten der Vision und dem Gamecontroller verarbeiten.

ceasar\_v2.drawio.svg

## Komponenten

### 2.4GHz Funk

Für die Kommunikation mit den Robotern wurde der SX1280 Transceiver gewählt. Die Gründe dafür sind hauptsächlich, dass die Manheim TIGERs diesen benutzen und beim Wettbewerb in Bangkok deutlich weniger Funkstörungen hatten. Nach weiterer Forschung hat sich auch ergeben, dass der SX1280 zur Steuerung von Dronen viel benutzt wird. Als Leistungsverstärker wird der SKY6112 benutzt, welcher für eine Ausgangsleistung von 200mW (maximale erlaubte Leistung in der EU bei 2.4Ghz) sorgt. Dieser bietet auch die Möglichkeit, eine von zwei Antennen auszuwählen.

### Ethernet

Eine Möglichkeit der Kommunikation mit dem Server ist Ethernet. Ethernet hat den Vorteil, dass auch Vision und Gamecontroller Daten verarbeitet werden können. Als Ethernet PHY (Physical layer) wird der LAN8741A-EN benutzt. Die Wahl dieses Ethernet PHYs wurde uns aufgrund der Chipknappheit abgenommen. Es wird ein RJ45 Stecker mit integrierten Magnetics genutzt.

### USB

Die zweite Möglichkeit zur Kommunikation mit dem Server ist USB 2 FS. In der ersten Version von Caesar wurde auch USB als Schnittstelle zum Server verwendet und das hat ganz gut funktioniert.

### ARGB

Insgesamt sind auf dem Board 16 Adressierbare RGB LEDs verbaut. Diese können genutzt werden, um den Status von jedem Roboter anzuzeigen.

## uController

Als Microcontroller wurde der ATSAM4E8C verwendet. Auch wurde die Auswahl hauptsächlich durch die Verfügbarkeit entschieden. Die mindestanforderungen an den uController waren: USB, SPI, Ethernet. Der ATSAM4E8C war einer der wenigen Controller, der Ethernet hatte.

## Fehler

Der USB port kann nicht zur kommunikation genutzt werden, da der Microcontroller dafür einen externen Crystal braucht.

## Bilder

caesarv2-top.png

caesarv2-bottom.png

# Bossix (Mainboard) v2

## ACHTUNG!!!

Nicht am Peaktech netzteil benutzen!!!

- Notiz für Spätere Entwicklungen: Dioden zur Limitierung der Eingangsspannung einbauen

Das Mainboard Steuert den gesamten Roboter.

mainboard\_v2\_block\_diagram.drawio.svg

## Komponenten

### Stromversorgung

Die Stromversorgung wird durch einen 6S LiPo Akku bereitgestellt. Ein Taster wird mit einer toggle schaltung zum an- und ausschalten verwendet. In diese Schaltung sind auch ein Softstart und eine 10A Sicherung integriert. Die 18V bis 25.2V vom Akku werden von zwei Buck Convertern in 12V und 5V umgewandelt. Die 5V werden ebenfalls nochmal mit einem Linearregulator in 3.3V umgewandelt.

### Haupt uController

Der haupt Microcontroller ist für die Kommunikation mit dem Server, das Ausschalten des Roboters, die IMU, den Buzzer, die Lichtschranke, den Dribbler und die UI zuständig. Dafür wird ein RP2040 verwendet. Über USB kann der Controller konfiguriert und debugged werden.

### 2.4GHz Funk

Für die Kommunikation mit dem Server wird der SX1280 Transceiver zusammen mit dem SKY66112 Leistungsverstärker verwendet. Mit dem Verstärker ist eine ausgangsleistung von bis zu 200mW möglich. Die antenne wird mit einem u.fl Stecker verbunden, was viel freiheit in der Platzierung dieser bietet.

# Inertial Measurement Unit (IMU)

Die IMU misst die Beschleunigung sowie die Rotation des Roboters. Diese Daten können dann genutzt werden, um die Position des Roboters zu bestimmen. Die IMU ist in der Mitte des Roboters verbaut. Die IMU ist über SPI mit dem haupt uController verbunden.

## Neben uController

Der zweite Microcontroller ist ebenfalls ein RP2040. Dieser ist jedoch für die Steuerung der Motoren und die Bedienung des Kickers zuständig.

## Pins

gpio #	Maincontroller	Motorcontroller
0	RF CRX	Kicker Trigger 1
1	RF CPS	Kicker Trigger 2
2	RF CTX	Kicker DAC0
3	RF Reset	Kicker DAC1
4	RF MISO	Kicker DAC2
5	RF CS	Kicker DAC3
6	RF SCK	Kicker DAC4
7	RF MOSI	Kicker DAC5
8	RF BUSY	Kicker DAC6
9	UI LED	Kicker DAC7
10	UI SDA	Kicker DAC8
11	UI SCL	Kicker DAC9
12	Not Shutdown	Kicker Not Fault
13	Power Switch	Kicker Not Done
14	RF DIO1 (V2.1 only)	Kicker Clear
15	Ball sense	Kicker Charge
16	TX	TX
17	RX	RX

gpio #	Maincontroller	Motorcontroller
18	CTS	CTS
19	RTS	RTS
20	Dribbler	-
21	Buzzer	-
22	IMU Int2	ESC CS1
23	IMU Int1	ESC CS3
24	IMU MISO	ESC CS0
25	IMU CS	ESC CS2
26	IMU SCK	ESC SCK
27	IMU MOSI	ESC MOSI
28	Battery Current	ESC MISO
29	Battery Voltage	Kicker Voltage

# Patches

## V2.0

Bei der dual 3.3V Zener sind pin 1 und 3 vertauscht.

Beim nächsten mal drauf achten, dass der sx1280 die DO pins für interrupt lines nutzt. Also mindestens einen an den uController anschließen.

Am sky66122 ist fälschlicherweise der CHL pin low, weshalb dauerhaft der low power transmit modus aktiv ist. Eigentlich sollte CHL high sein, um dauerhaft high power trans mit zu machen, was in einer output power von 200mW resultieren würde.

Es ist beim nächsten mal darauf zu achten für buck converter spulen mit shield zu nutzen.

Durchtrennungen auf der Oberen Lage	Durchtrennungen auf der Unteren Lage	Extra Verbindungen
mainboardv2-top-cuts.png	mainboardv2-bottom-cuts.png	mainboardv2-top-patches.png

## Alle Versionen

Der OpAmp zur Strommessung ist falschrum beschaltet. Durch das Positive feedback wird dauerhaft die Maximalspannung (3.3V) ausgegeben.

~~D101 wird weggelassen. Dadurch wird der Soft Start ausgeschaltet. Das ist nötig, da der Power MOSFET den Slow Start nicht aushält.~~

C101 wird durch 1uF und R101, R102 und R103 werden durch 1MR ersetzt. Das erzeugt einen deutlich langsameren Soft-Start, wodurch das Mosfet nicht so extrem belastet wird.

Die Kicker schaltung schaltet zu schnell. Deshalb schaltet sich der Kicker charge controller ab und geht in einen fehler zustand. Um das zu beheben wird R521 durch einen 1KOhm und R511 durch einen 100kOhm Widerstand ersetzt. Damit ist zumindest ein laden bis 200V möglich.

# Bilder

mainboard-2023-front.png

mainboard-2023-back.png

# Asterix

## ESC

Die Aufgabe eines Electronic Speed Controller (ESC) ist es einen Motor anzusteuern. Dabei kann zur Steuerung z. Bsp. ein Drehmoment, eine Geschwindigkeit oder eine Position vorgegeben werden. Der ESC steuert den Motor so an, dass die soll Größe erreicht wird.

Generell wird zwischen zwei arten unterschieden:

- Closed loop
- Open loop

Bei closed loop controllern werden vom System gemessene Daten verwendet um den Motor anzusteuern. Gemessene Daten können beispielsweise der Strom oder die Position des Motors sein.

Bei open loop controllern werden keine Daten des Systems gemessen und die Steuerung nutzt nur ihren internen Zustand um den Motor zu steuern. Meistens führt dies dazu, dass der Motor nicht so gut gesteuert werden kann wie in einer closed loop Steuerung.

## BLDC Motor

Brushless Direct Current (BLDC) Motoren sind in der Robotik viel verwendete Motoren. Vorteile sind geringer Verschleiß so wie gute regelbarkeit der Geschwindigkeit.

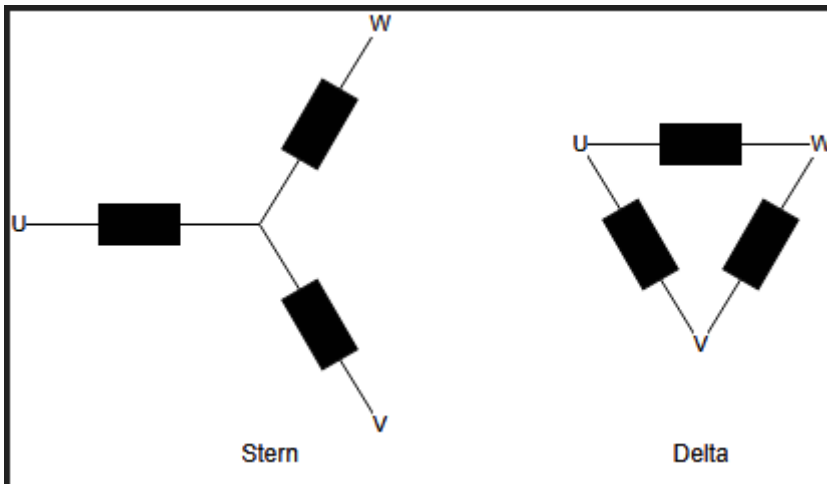
“ Konträr zum Namen können BLDC Motoren nicht direkt mit Gleichstrom (Direct Current) betrieben werden. Der Name kommt von den DC Motoren, welche so genannte Bürsten (Brushes) besitzen, da BLDC Motoren immer mit einem ESC verwendet werden, welcher Gleichstrom für den Motor passend umwandelt, und die Verkabelung bis zum ESC somit sehr ähnlich zu der eines DC Motors ist. Daher können DC Motoren oft mit wenig Aufwand durch BLDC Motoren ersetzt werden.

# Aufbau

Ein BLDC Motor besteht Elektrisch aus drei Spulen. Diese werden genutzt um alternierende Magneten an- bzw. ab-zustoßen.

fig3-a-bldc-monitor-en.gif

Elektrisch werden die Spulen entweder als Stern oder als Delta verbunden:



Für die Steuerung ist die Art der Verbindung egal, da die beiden Darstellungen in einander umgewandelt werden können. Die Anschlüsse **U**, **V** und **W** werden auch Phasen genannt.

Der Motor kann, wie oben gezeigt, durch einfaches an und aus schalten der Spulen gedreht werden. Dies wird in der Praxis jedoch nicht gemacht, da die Steuerung damit sehr ungenau ist. Stattdessen wird für jede Phase eine Spannung erzeugt. Dadurch kann der Strom durch die Spulen fein eingestellt werden und eine genauere Steuerung realisiert werden. Die Spannung wird meistens mittels Puls Width Modulation (PWM) erzeugt. Die Frequenz dieser sollte dabei über 25kHz liegen um unangenehme geräusche durch bspw. Spulenfielen zu vermeiden. Höhere Frequenzen führen oft zu einem effizienterem Betrieb des Motors, reduzieren aber auch das Drehmoment.

Mehr:

- [wikipedia bldc](#)

# FOC

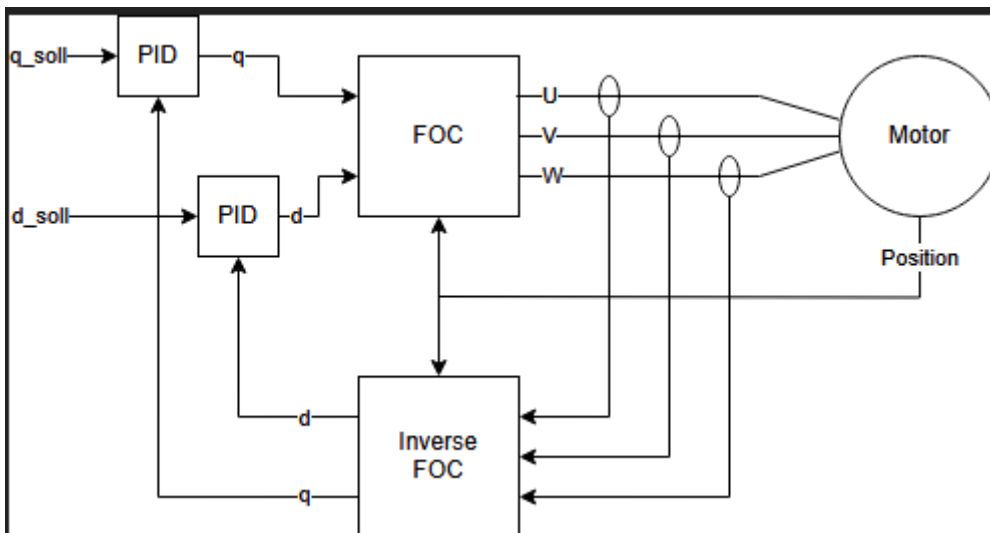
Field Oriented Control (FOC) ist eine Methode um den Strom durch die Spulen in zwei Komponenten zu teilen:

- q: Trägt zum Drehmoment bei
- d: Trägt nicht zum Drehmoment bei (Feldschwächung)

Zur Umwandlung werden die sogenannten Clarke und Park Transformationen verwendet. Durch die Trennung der Komponenten können das Drehmoment und die Feldschwächung separat geregelt werden. Die d-Komponente wird dabei i.d.R. auf 0 geregelt, es kann jedoch Vorteile haben den d-Anteil nicht auf 0 zu regeln um eine Feldschwächung zu erzeugen.

Damit die d- und q-Komponenten getrennt werden können muss die Rotation des Motors bekannt sein. Hierfür können ein Encoder oder Hall Effekt Sensoren genutzt werden. Ersteres ist sehr genau und führt daher zu einem guten ergebnis, besonders bei langsamen bewegungen, während zweitere oft schon im Motor verbaut sind und dadurch keinen extra platz brauchen. FOC ist also eine closed loop technik.

FOC wird oft mit einer PID Regelung zu dem folgendem System kombiniert:



“ Die Idee von FOC ist nicht auf BLDC Motoren beschränkt. Schrittmotoren können damit genau so angesteuert werden, wobei die Clarke Transformation entfällt, da Schrittmotoren nur zwei Spulen haben. Bei DC Motoren wird die Transformation durch die Bürsten in Hardware erledigt. Allerdings steuert die angelegte Spannung nur die q-Komponente.

“ Es gibt weitere Transformationen um d und q in U, V und W umzuwandeln. Ein Beispiel dafür ist die Space-Vector-Transformation.

Mehr:

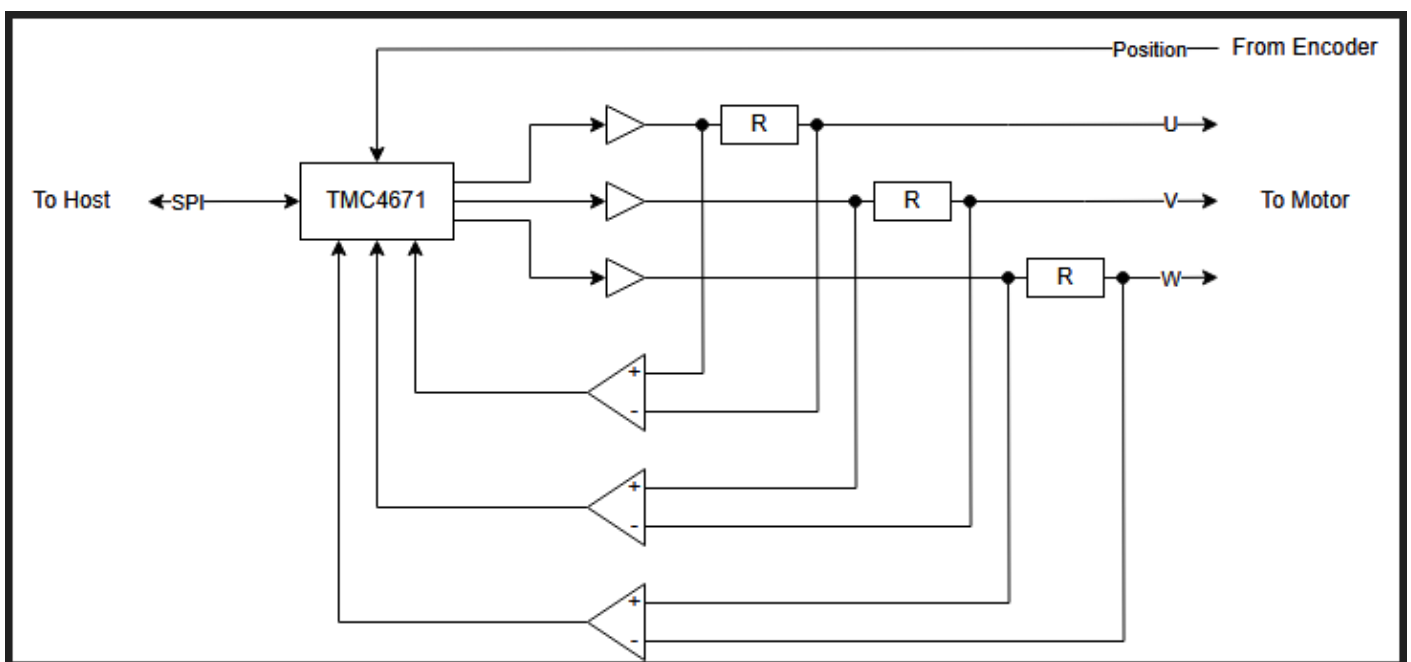
- [wikipedia FOC](#)

- [wikipedia Clark Transformation](#)
- [wikipedia Park Transformation](#)

# Asterix

Asterix ist unsere Motor Steuerung für die erste Roboter Generation. Diese basiert auf dem TMC4671 von Trinamic. Der Integrated Circuit (IC) implementiert FOC sowie PID Regler für Drehmoment, Feldschwächung, Geschwindigkeit und Position.

Auf dem PCB ist folgendes Implementiert:



Die Verstärkung der Spannung über die Strommesswiderstände wurde versucht selbst zu Implementieren. Die Implementierung lässt allerdings zu viel von dem PWM-Signal durch, wodurch die Messung unbrauchbar ist. Dadurch ist es nicht möglich die PID-Regler für d und q zu verwenden. Da die Architektur des TMC4671 den Geschwindigkeits-PID-Regler fest mit dem q-PID-Regler verdrahtet ist es allerdings auch nicht möglich die Geschwindigkeits-Regelung zu benutzen. Als workaround werden p und q direkt vom Host gesetzt. Die Regelung wird dabei auf den Host ausgelagert und benutzt einen PID-Regler um die Geschwindigkeit direkt mit dem q-Wert zu regeln.

aktuelle Version	Version in Arbeit	verantwortliche Person
[v2.4]	-	Max Känner

## Versionen

Version	Kommentar	Bearbeiter	Bearbeitungsstand	Fertigstellung
[v1.0]	-	Max Känner	veraltet	-
[v2.0]	-	Max Känner	veraltet	-
[v2.1]	-	Max Känner	veraltet	-
[v2.2]	-	Max Känner	veraltet	-
[v2.3]	-	Max Känner	aktuell	-
[v2.4]	-	Max Känner	aktuell	-

## Features

Version	# Regler / PCB	Regler Typ	Strom Messung
[v1.0]	1	TCM4671	□
[v2.0]	2	TCM4671	□
[v2.1]	2	TCM4671	□
[v2.2]	2	TCM4671	□
[v2.3]	2	TCM4671	□
[v2.4]	2	TCM4671	□

## Verzeichnis

<https://nextcloud.luhbots-hannover.de/f/14842>

# PCBS

# PCBS

Diese Seite ist eine Sammlung der unterschiedlichen PCBs, welche die Elektronik erstellt hat.

## Roboter

Name	Funktion	Gen1	Gen2
<a href="#"><u>Bossix</u></a>	Mainboard	X	
<a href="#"><u>Asterix</u></a>	Electronic Speed Controller (ESC)	X	
[?]	Encoder	X	
<a href="#"><u>Maya</u></a>	Kicker Charge Controller	X	
<a href="#"><u>Obelix</u></a>	Kick Controller	X	
<a href="#"><u>Verleihnix</u></a>	Lichtschanke	X	
[?]	Dribbler ESC	X	
<a href="#"><u>Batterix</u></a>	Battery Management System (BMS)	-	X

## Periferie

Name	Funktion	Gen1	Gen2
<a href="#"><u>Caesar</u></a>	Basestation	X	
[Diagnostix]	Roboter Interface	X	
[LUlgi]	Roboter UI	X	

# Sonstige

Name	Funktion
[Idefix]	Teststation?
[?]	Workshop PCB?
[?]	Workshop PCB
[Kabeltester]	Kabeltester

# PCB BOM

Asterix

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Asterix\\_V2.3.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Asterix_V2.3.html)

Bossix

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Bossix\\_V2.2.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Bossix_V2.2.html)

Caesar

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Caesar\\_V1.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Caesar_V1.html)

Diagnostix Portable

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Diagnostix-Portable\\_V1.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Diagnostix-Portable_V1.html)

Diagnostix Stationary

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Diagnostix-Stationary\\_V1.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Diagnostix-Stationary_V1.html)

Encoder

[https://luhbots-hannover.de/wp-content/uploads/2022/03/Encoder\\_V1.1.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/Encoder_V1.1.html)

ESP Breakout

[https://luhbots-hannover.de/wp-content/uploads/2022/03/ESP-Breakout\\_V1.html](https://luhbots-hannover.de/wp-content/uploads/2022/03/ESP-Breakout_V1.html)

ESPower V1.2

<https://luhbots-hannover.de/wp-content/uploads/2022/06/ESPower-V1.2.html>

Idefix

<https://luhbots-hannover.de/wp-content/uploads/2022/03/ibom.html>