

GRUPPE 1: HIMMELWACHT

Fabian Becker, Jedrik Jürgens, Nicolas Koch, Michael Specht & Jonathan Wohlrab



EINFÜHRUNG

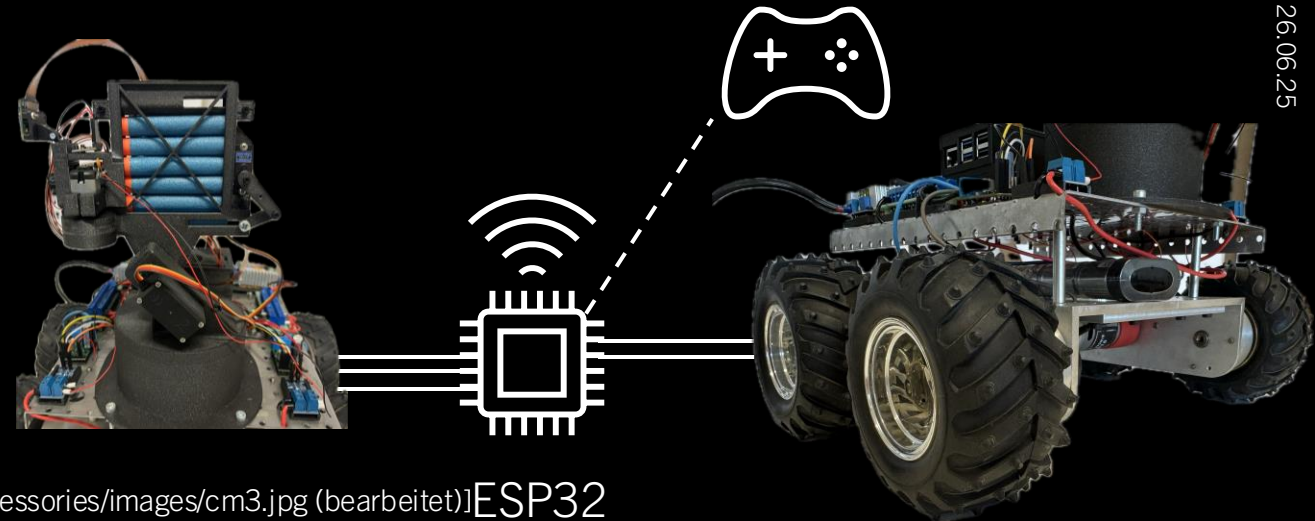
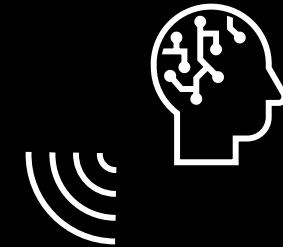
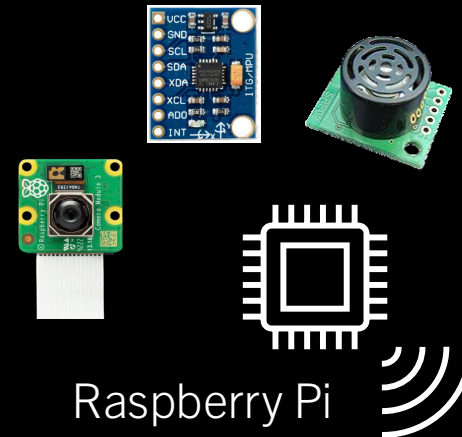


Jonathan Wohlrab

IDEE

- Fahrbares Boden-Luft-Abwehrsystem
- Automatische Erkennung von Zielen
- Abschuss von Zielen mittels Nerf Pfeilen

Papierflieger --> geführten Holzflieger



[Quelle Bild Kamera: <https://www.raspberrypi.com/documentation/accessories/images/cm3.jpg> (bearbeitet)]ESP32

[Quelle Bild Ultraschall: https://exp-tech.de/cdn/shop/products/srf02c_600x600_e765c17a-3d11-4594-a278-14d112147e26.png?v=1689234824 (bearbeitet)]

[Quelle Bild Gyro: <https://www.majju.pk/product/gy-521-mpu6050-module-3-axis-gyroscope-accelerometer-module/>]

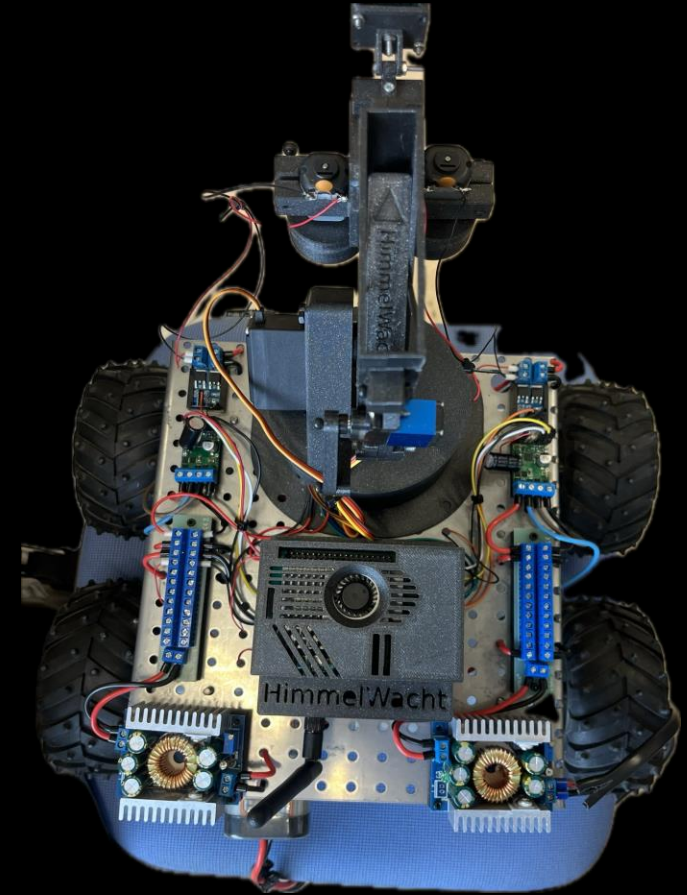
PROJEKTAUFBAU



Nicolas Koch

STROMVERSORGUNG

- Layoutidee von vorherigem Projekt
- Zwei Spannungsbereiche
 - 5V-Verteiler (links) für Sensoren und ESP32
 - 8V-Verteiler (rechts) für Motoren
- Versorgung durch 12V Batterieverbund
- Raspberry Pi wird durch Powerbank versorgt



CAD – KONSTRUKTION

Fabian Becker, Michael Specht

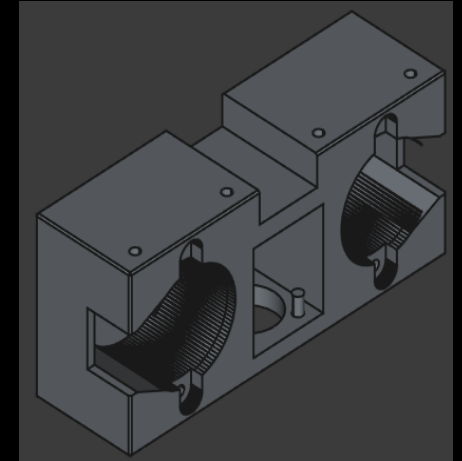
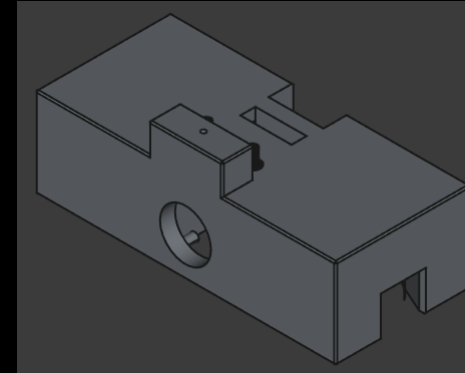
VORLAGE



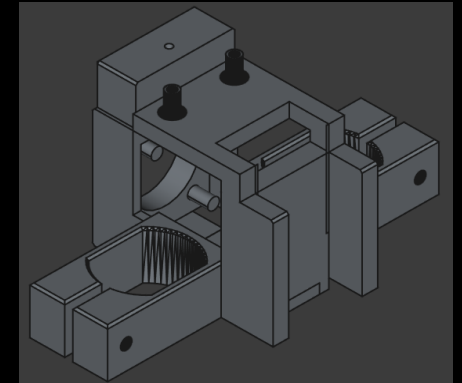
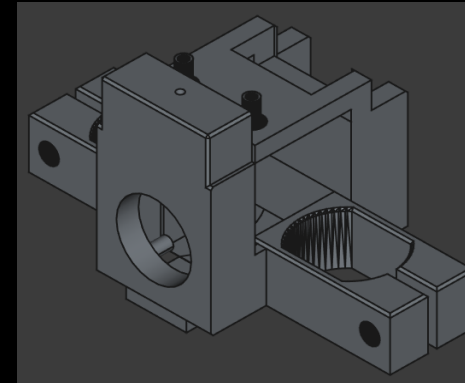
Quelle: <https://www.thingiverse.com/thing:4870102>

GESCHÜTZARM

- Version 1:
 - Neukonstruktion
 - Anforderung: Ultraschall-, Gyrosensor und Servomotoren fixieren
 - Soll bestehendes Magazin/Rohr benutzen
- Version 2:
 - Kombination aus Vorlage und Version 1



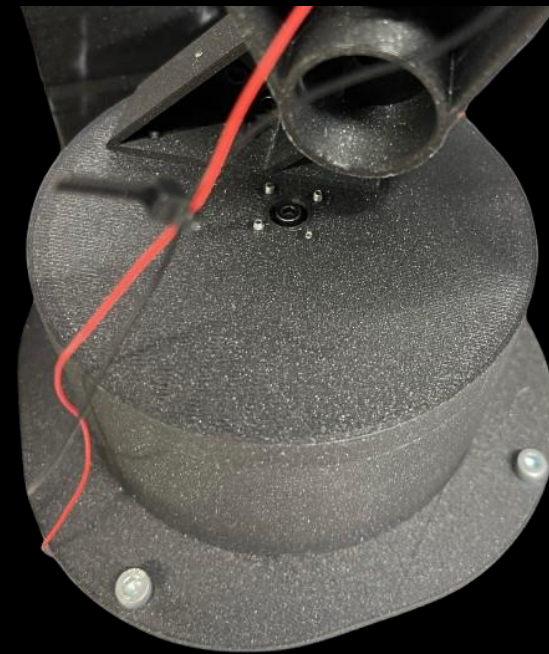
Modell Version 1



Modell Version 2

GESCHÜTZPLATTFORM

- Unterbau aus vorherigem Projekt
 - Größe und Form passend
 - spart Konstruktionsaufwand
- Rekonstruktion der Abdeckung mit passenden Bohrlöchern
 - direkte Übernahme vom Geschützhalter und Magazin (Vorlage) möglich



MIKROCONTROLLER-CASE

- ESP-32 und Raspberry PI übereinander platziert
 - wegen Kabellänge und Platz
- ESP-32 Case auf Basis einer Vorlage selbst konstruiert
 - Kühllöcher für ESP in Form von HimmelWacht als kleines Detail
- Kombination mit Raspberry PI Case (Vorlage)
- auch hier Anwendung unseres modularen Stecksystems



PROGRAMMIERUNG – ESP32

(ESP-IDF)

Fabian Becker, Michael Specht

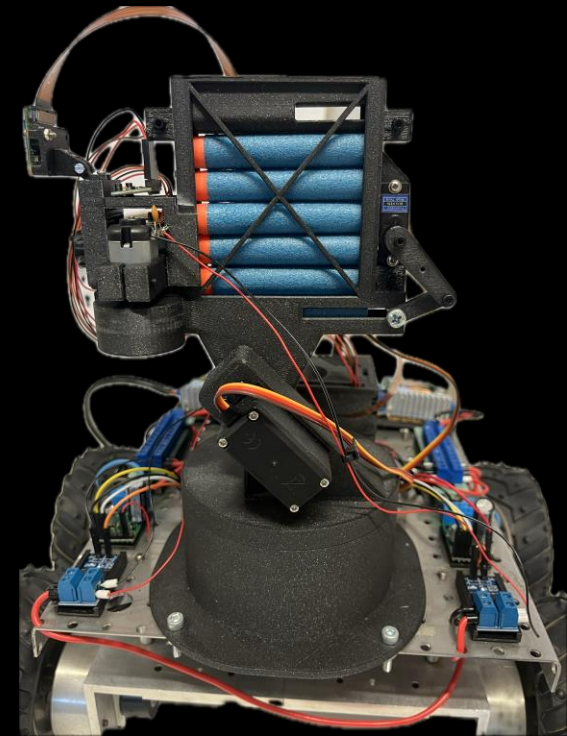
DUALSHOCK 4 - TREIBER

- erste Version:
 - eigener Treiber mittels ESP-IDF Bluetooth Classic – HID API
 - Senderate von über 700Hz überforderte den Bluetooth-Stack
- daher Umstieg auf Bluepad32
 - für kleinere Mikrocontroller optimierter Bluetooth Stack
 - viele Features & Mikrocontroller unterstützt
 - Dokumentation für ESP-IDF sehr dünn
- Implementierung von Vibration und Farbänderung
- Multicore-safe mittels FreeRTOS Queues



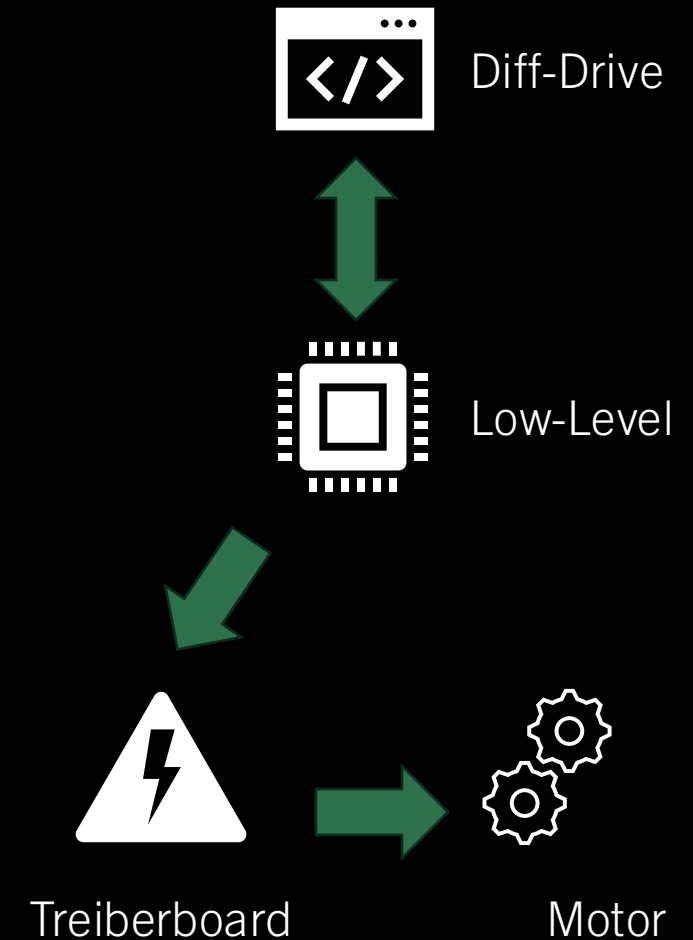
ANSTEUERUNG SERVO MOTOREN

- Ansteuerung von 3 Servomotoren
 - 2 größere Servomotoren für Ausrichtung der Geschützplattform
 - 1 kleinerer Motor schiebt Nerf-Darts in die Flywheel Motoren
- Nutzung eines PWM-Treiberboards mittels I²C
 - ESP hat keine Kapazitäten für direkte Ansteuerung
- Ansteuerung: PWM Duty Cycle Länge -> Drehung an bestimmte Position
 - manuelles Herausfinden der Extrema (-90°, 90°) , Aufstellen von Mapping Grad -> Duty Cycle
 - für kleinen Servo: erfassen der Start & Endposition des Schussarmes durch manuelles testen



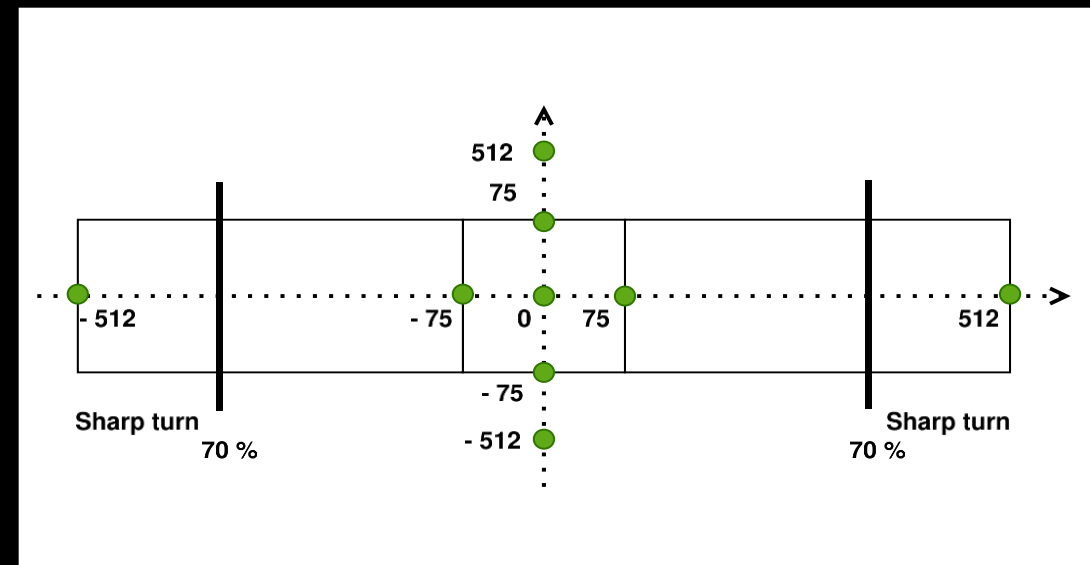
LOW-LEVEL MOTORTREIBER

- Ziel: Abstraktion
 - Polulu-Treiberboard-Modus (Sign-Magnitude + Locked-Antiphase)
 - Performance
 - Hardware zu schützen
 - Tests



DIFFERENTIAL DRIVE

- Anforderung: FreeRTOS Task, der mittels Queue Controllerwerte verarbeitet
- Drive Logik
 - im Stand entgegengesetzte Drehung ermöglichen
 - Controller Stick-Drift beachten
 - Nachrichtentrafic reduzieren



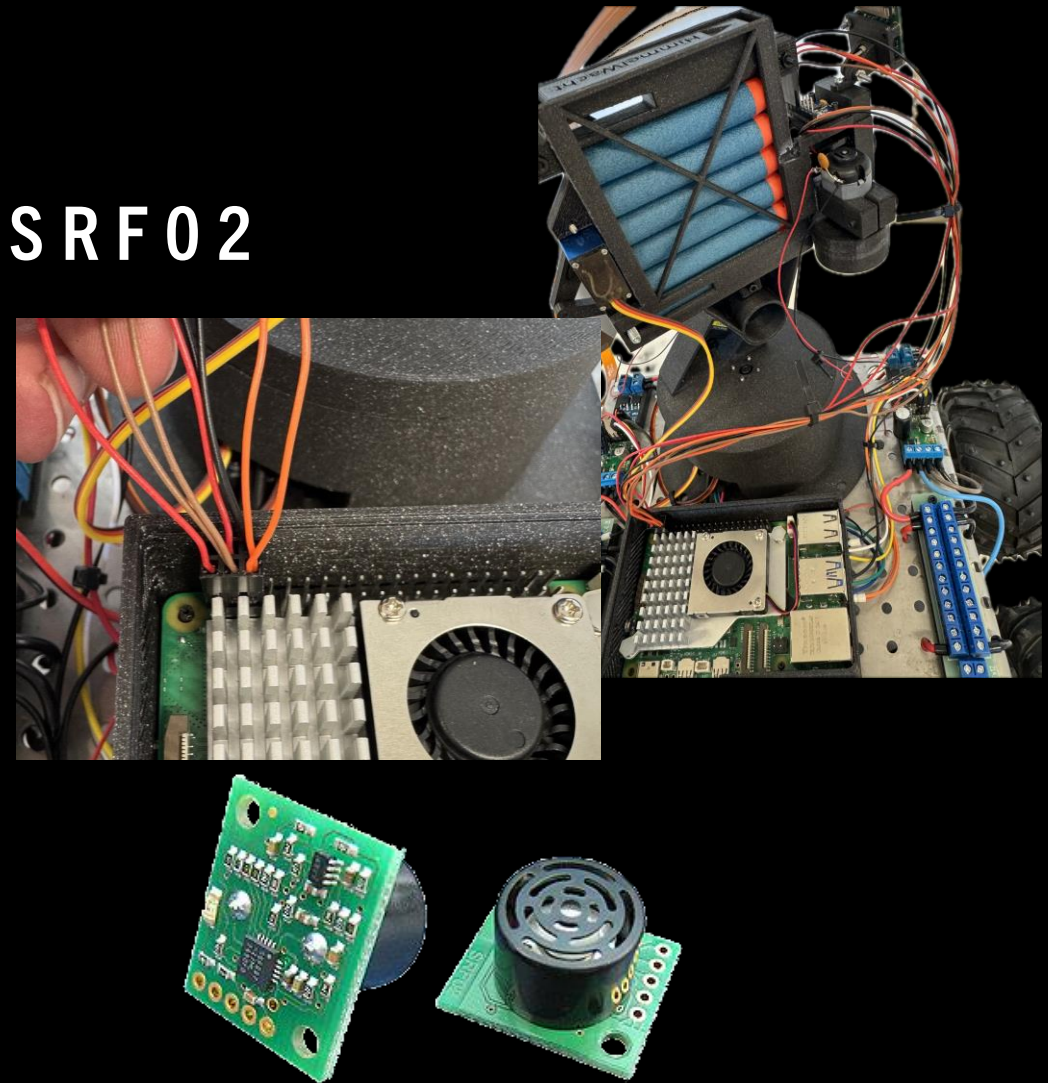
```
bool is_null_position = (abs(x) < DRIVING_NULL_BOUNDARY && abs(y) < DRIVING_NULL_BOUNDARY);  
bool significant_change = (abs(x - diff_drive_prev_x) >= DRIVING_MIN_CHANGE ||  
| abs(y - diff_drive_prev_y) >= DRIVING_MIN_CHANGE);
```

PROGRAMMIERUNG – RASPBERRY PI

Nicolas Koch, Jonathan Wohlrab

ULTRASCHALLSENSOR - SRF02

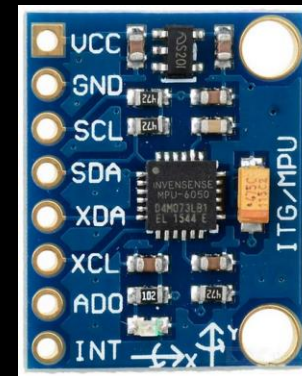
- Kommunikation mit Sensoren über I2C
 - Ultraschallsensor liefert Entfernung zum Ziel
 - Umfangreiche Funktionalität bereits auf dem Chip bereitgestellt
- > Umsetzung deutlich einfacher als erwartet



[Quelle: https://exp-tech.de/cdn/shop/products/srf02c_600x600_e765c17a-3d11-4594-a278-14d112147e26.png?v=1689234824
(bearbeitet)]

GYROSENSOR - MPU6050

- Idee: Kalibrierung der Geschützplattform
- 3-Achsen-Gyroskop – und Beschleunigungssensor
- Liefert Neigungswinkel des Geschützarms
- Glättung der Werte durch Kalman-Filter

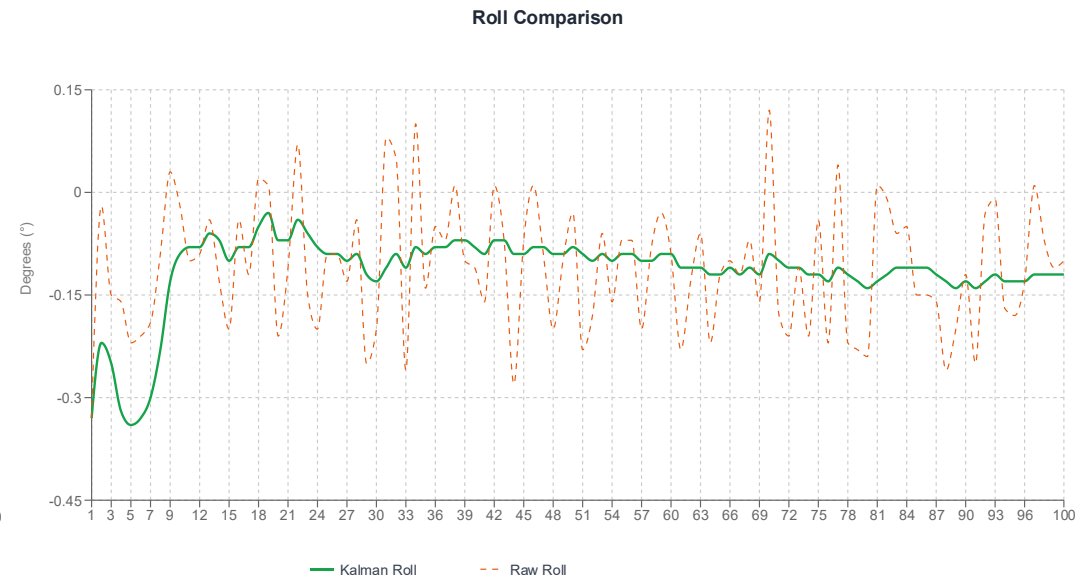


[Quelle: <https://www.majju.pk/product/gy-521-mpu6050-module-3-axis-gyroscope-accelerometer-module/>]

GYROSENSOR - MPU6050



Noise reduction: 59.4%



Noise reduction: 37.4%

KI-ERKENNUNG



Jendrik Jürgens

HARDWAREPLATTFORM

PROBLEME & LÖSUNGEN

- Raspberry Pi 5 & Camera Module 3 verwenden Libcamera-Stack
 - Fehlende Unterstützung in OpenCV - Keine direkte Verwendung der Kamera möglich
 - Lösungsansatz: G-Streamer Pipeline verwenden um Kamera-Video in OpenCV zu pipen
 - Kompilier- und Installationsdauer ~ 2h => Zeitintensiv vor allem bei Fehlern (Busy Waiting)
- KI-Erkennung— Performance Probleme
 - Keine dedizierte GPU => Stream + KI-Auswertung laufen auf CPU, nicht performant
 - Lösungsansatz: KI muss auf Laborrechner laufen, Videostream muss übertragen werden



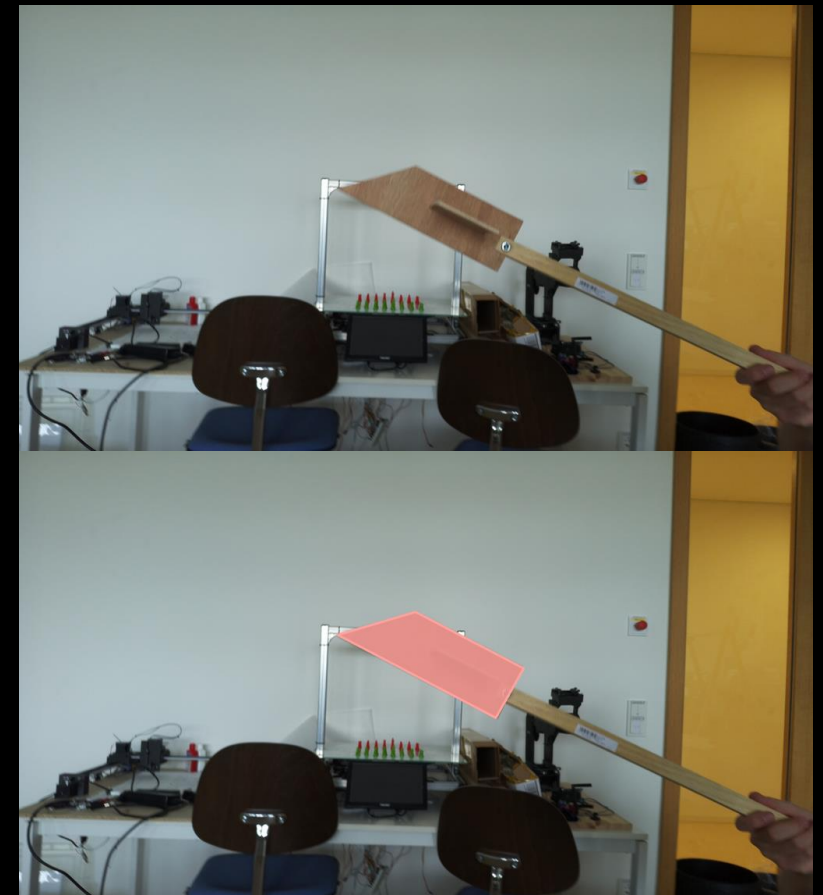
[Quelle:
<https://assets.raspberrypi.com/static/8f6eca535dbb23a21eb41a748050e3a0/f7ad0/16gb.webp> (bearbeitet)]



[Quelle:
<https://www.raspberrypi.com/documentation/accessories/images/cm3.jpg> (bearbeitet)]

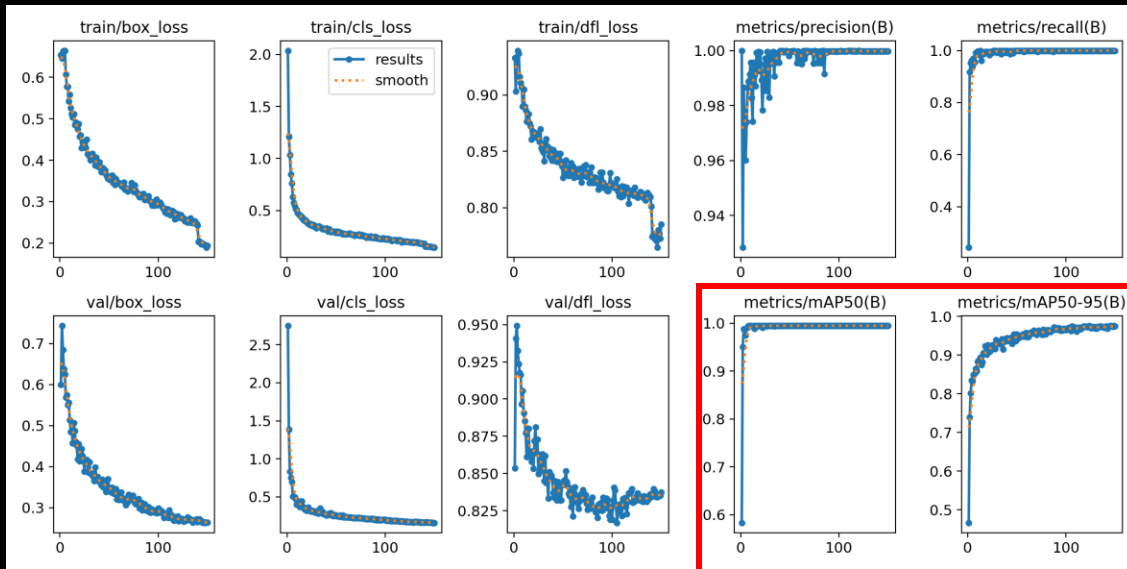
DATENAQUIRIERUNG & LABELING & TRAINING

- Qualität der Daten = Performance der KI
- Trainingsbilder mit libcamera-Stack aufgenommen (vom Pi)
- Labeling mit OpenSource Labelstudio
- Gelabelte Bilder: 1158
- Training: RTX 3070, 150 Epochs, 810/117/231 Train/Test/Val
- Trainingsdauer: ~30 Minuten pro Modell

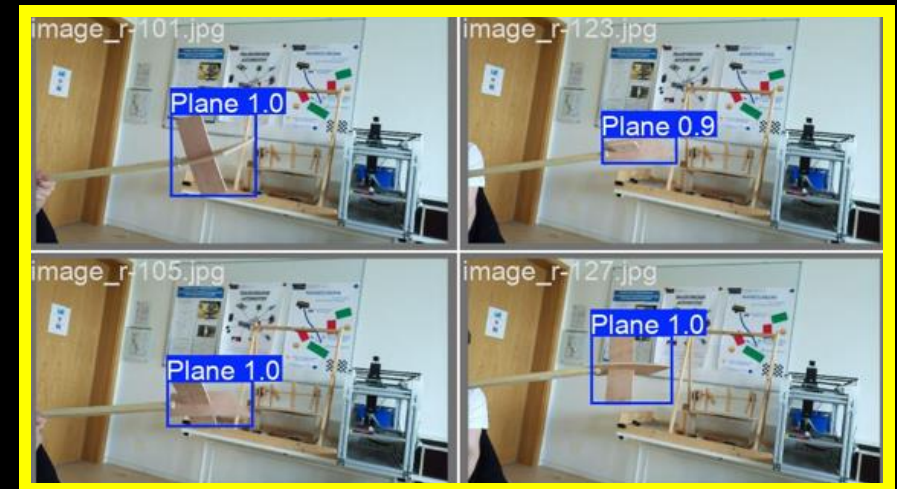
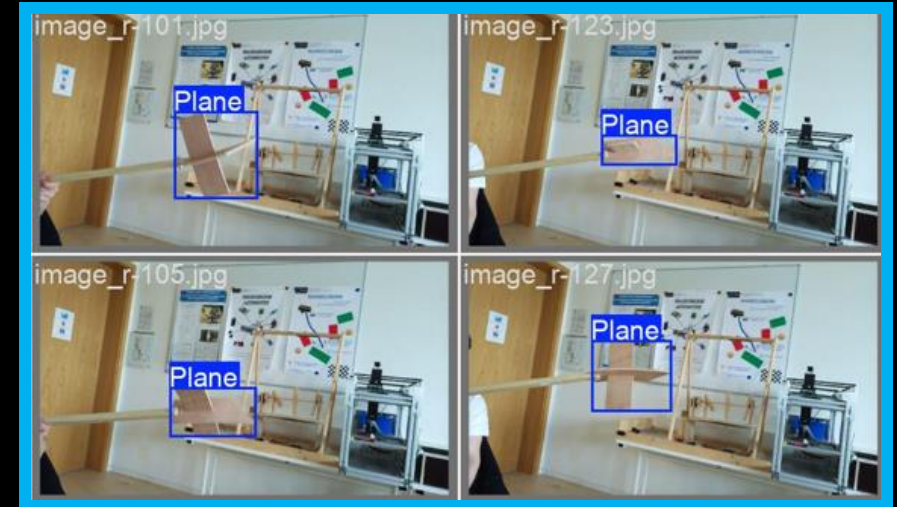


KI-TRAINING & VALIDIERUNG

- YOLOv8n/s, YOLOv11n/s (Industrie-Standard- klein & schnell)
- Transferlearning mit vortrainierten Gewichten



Trainingsverlauf, Intersection over Union (IoU)









Ausschnitt: Validierung Soll / Vorhersage

ERGEBNISVERGLEICH

Jonathan Wohlrab

ABNAHMEKRITERIEN

- Steuerung via Bluetooth-Controller 
- Manuelle Steuerung des Systems 
- Geschütz muss dreh- und neigbar sein 
- Das System muss Nerf Darts abfeuern können 
- Ansatz zur automatischen Erkennung und Positionsberechnung von Zielen vorhanden 
- Lautsprecher soll Töne abspielen können 

QUELLEN

- <https://www.bluetooth.com/specifications/specs/human-interface-device-profile-1-0/>
- <https://bluepad32.readthedocs.io/en/latest/>
- https://www.elektronik-kompodium.de/sites/praxis/bauteil_sg90.htm
- <https://docs.ultralytics.com/guides/yolo-performance-metrics/#introduction>
- <https://github.com/opencv/opencv/issues/21653>