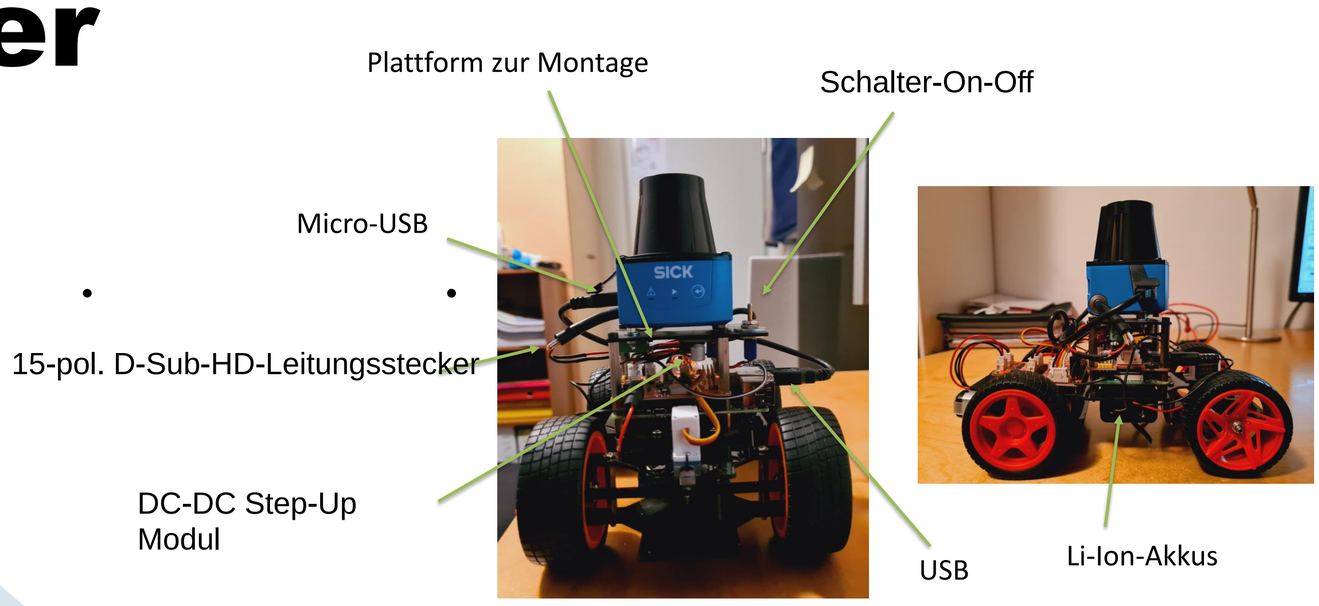
Institut für Naturwissenschaften und Mathematik INM
Hochschule Konstanz
Fakultät Elektrotechnik und Informationstechnik

Betreuer: Prof. Dr. Hartmut Gimpel (Fakultät MA)
Studenten: James Kibii, EIB; Manuel Greiter, EIB

Projektarbeit

Steuerung eines Kleinfahrzeugs mit einem Laserscanner

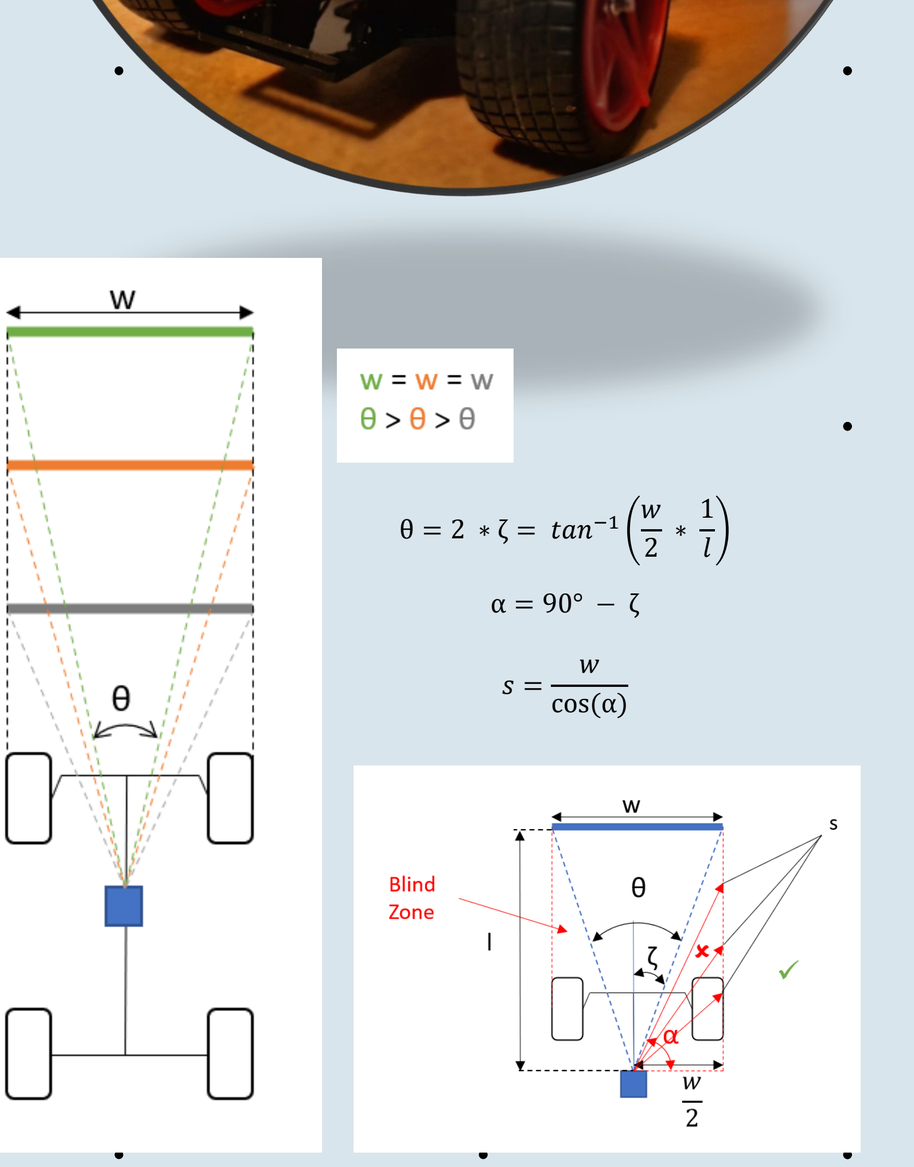
In diesem Projekt wurde ein Maker-Kleinfahrzeug um einen industriellen Laserscanner (2D-LiDAR) ergänzt. Mit dem integrierten Raspberry-Pi 4 Einplatinencomputer konnten die Punktwolken-Daten des Laserscanners ausgelesen werden und das Auto autonom gesteuert werden.



Die Stromversorgung des Laserscanners konnte mithilfe von 2 Li-Ion-Akkus in Reihe geschaltet in Kombination mit einem DC-DC Step-Up Modul sichergestellt werden. Für die sichere Montage der Bauteile und des Laserscanners auf dem Fahrzeug wurde eine Plattform mithilfe eines 3D-Druckers entwickelt.

```python
import usb.core
from lib import parser
from lib import constants as c


class Lidar(object):
    def __init__(self, id_vendor, id_product):
        self.__idVendor = id_vendor
        self.__idProduct = id_product
        self.__dev_TIM = None
        self.__distances = []
        self.__angles_deg = []

    def connect(self):
        # find TIM- Device
        self.__dev_TIM = usb.core.find(idVendor=self.__idVendor,
                                       idProduct=self.__idProduct)

        # In case no TIM Device is found
        if self.__dev_TIM is None:
            raise ValueError("Please ensure that your TIM-Device is "
                             "connected for autonomous driving")

    def __write(self, msg):
        # write(endpoint, data)
        self.__dev_TIM.write(2 | usb.ENDPOINT_OUT, "\x02" + msg + "\x03\0")

    def __read(self):
        # read(endpoint, size_or_buffer, timeout)
        ret = self.__dev_TIM.read(1 | usb.ENDPOINT_IN, 65535, timeout=100)
        return ret

    def scan(self):
        self.__write(c.command["message"])  #message is sRN LMDscandata
        raw_data = self.__read()
        raw_data = "".join(chr(i) for i in raw_data[1:-1])
        raw_data_arr = raw_data.split()
```

Über USB-Endpoints und der Python-Bibliothek PyUSB konnten vom Raspberry Pi aus Kommandos an den Lidar versendet werden und Telegramme mit den Daten des Sensors empfangen werden. Die Telegramme enthalten die Rohdaten, welche mithilfe des mitgelieferten Datenblatts ausgewertet werden konnten. So wurde eine Kommunikation zwischen Raspberry Pi und Laserscanner hergestellt und eine autonome Steuerung des Fahrzeugs mithilfe von Python ermöglicht.

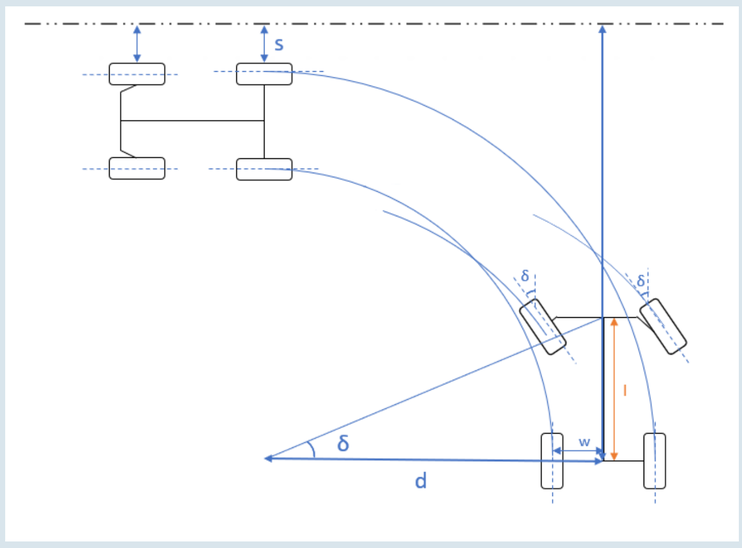


w = w = w
θ > θ > θ

$$\theta = 2 * \zeta = tan^{-1}\left(\frac{w}{2} * \frac{1}{l}\right)$$

$$\alpha = 90° - \zeta$$

$$s = \frac{w}{\cos(\alpha)}$$



Die autonome Pfadplanung wird durch 2 Algorithmen bestimmt. Der Hauptfahralgorithmus, auf den Bildern unten in der Mitte zu sehen, ist verantwortlich für die Überprüfung von Hindernissen, die sich vor dem PiCar befinden können, und für die Einstellung der entsprechenden Geschwindigkeit. Der zweite Algorithmus ist das sogenannte Ackermannische Prinzip., welches auf dem Bild oberhalb dargestellt ist. Das Ackermann-Prinzip kommt zum Einsatz, sobald das PiCar eine Kurve fahren soll. Der richtige Wendepunkt wird an den Vorderrädern eingestellt, was eine Kollision des Autos mit dem vorausfahrenden Objekt verhindert.