

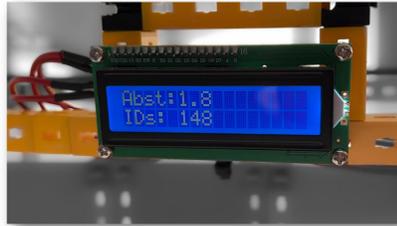
Steuerung eines Fischertechnik-Tors mit einem Radar-Sensor

Julius Czarnecki (EIB), Samuel Henkel (EIB)

Betreuer: Prof. Dr. Hartmut Gimpel

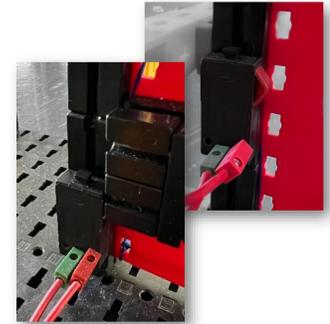
Aufgabenstellung

- Selbstständiger Entwurf und Aufbau eines Fischertechnik Tors
- Mithilfe eines industriellen Radar Sensors soll sich das Tor automatisch öffnen, sobald sich ein Kleinfahrzeug nähert
- Dafür sollen Radar Messdaten mit einem Raspberry Pi 4 in Python ausgelesen und ausgewertet werden



Displaysteuerung

- Kommunikation zwischen Raspberry Pi und 16x02 LCD Display mittels I²C Kommunikationsprotokoll
- Entsprechende Python Module *Icddriver* und *i2c_lib* stellen Bus und Methoden zum Beschreiben des Displays
- Wiedergabe der Distanz der gemessenen Objekte unter 2m mit jeweiliger Objekt-ID



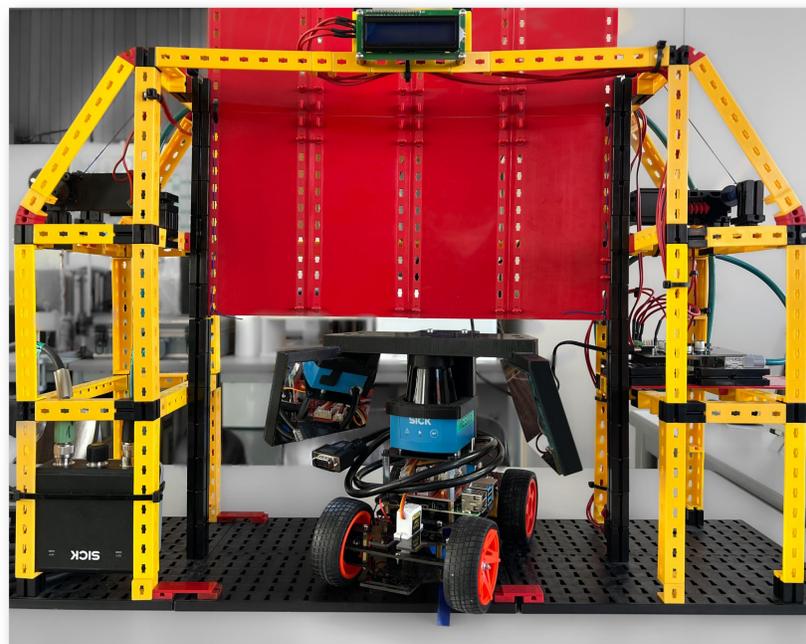
Auslesen der Taster

- 2 Fischertechnik-Taster an der Laufschiene des Tors, die auslösen wenn Tor vollständig geöffnet bzw. geschlossen
- Liegen jeweils an 3,3V und einem GPIO des Raspberrys an und so beschaltet, dass normalerweise geöffnet
- Auslesen in Python durch Definieren der jeweiligen GPIOs als Input und Aktivierung der internen Pull-Down Widerstände des Raspberrys, sodass sauberer Pegel anliegt



Motoransteuerung

- 2 Gleichstrommotoren von Fischertechnik mit maximal 1A Eingangsstrom
- Bei Ansteuerung wird die Seilbinde aufgezogen, wodurch sich das Tor öffnet
- Nutzung eines Getriebes, für ein ca. 10-faches Drehmoment
- Steuerung erfolgt über die PWM-Generierung des MC33932 Motortreiber-Boards
- Frequenz und Duty Cycle des PWM-Signals können variiert werden
- Implementierung in Python via *Motor_Driver* und *PiSoftPwm*



SICK RMS 1000 Radarsensor

- Arbeitsbereich des Radars: 0,4m bis 100m
- Öffnungswinkel: 60° horizontal und 4° vertikal
- Frequenzband: 61 GHz bis 61.5GHz
- Kommunikation zwischen Radarsensor und Raspberry Pi über Ethernet
- Nach Messanfrage des Raspberry Pi schickt der Sensor Abstandsangaben in mm von sichtbaren Objekten, sowie die zugehörigen Objekt-IDs
- Für Auswertung der Daten werden nur Objekte innerhalb eines Umkreises von 2m beachtet
- Realisierung der Kommunikation via TCP IP4v über das Socket-Modul in Python
- SICK verwendet das Übertragungsformat CoLa A
- Grafische Darstellung einer Testmessung durch die Herstellersoftware SOPAS:

Raspberry Pi 4 und seed Motor Driver

- Herzstück des Aufbaus, verantwortlich für Auslesen der Radar Messdaten und der Taster, Ansteuerung der Motoren und des Display
- An Motor Driver Board sind 12V Versorgungsspannung für die Motoren und den Raspberry Pi angeschlossen
- Auswertung der Radar Messdaten für die darauf basierende Entscheidung zur Öffnung des Tores
- Zur Feststellung wird Distanz der Objekte über Zeitraum getrackt und überprüft ob es sich 5mal nähert bevor es sich 5mal entfernt
- Notwendig, da selbst statische Objekte in ihrer Distanz einige mm variieren, jedoch nicht zu einer Öffnung des Tores führen sollen
- Öffnung des Tores bei gleichmäßiger Näherung eines Objektes auf 1m

