

Abgabe Gruppe: Linus Blanke & Christoph Lederbogen

Dieses Dokument ist die Dokumentation der Tätigkeit im 6. Semester und umfasst das [Pflichtenheft](#), die [Implementierung](#) der Pflichten sowie eine Beschreibung der Erweiterung der [Balancer Library für den Laser Sensor: SensorTOF](#).

Pflichtenheft

Die Gruppe wird sich als Hauptaufgabe auf die Ansteuerung des Drehdrucktasters konzentrieren (im 5. Semester geschehen). Zum Themengebiet im 6. Semester gehören die Ansteuerung der Sensoren, Visualisierung der Ergebnisse und Interaktion über den Drehdrucktaster. Dabei sollen die Aufgaben im Programmablauf zeitlich terminiert ablaufen.

- P001: Display

Das Display ist für die Visualisierung vorhanden.

- Es soll möglich sein, die Sensorergebnisse in einer passenden Visualisierung auf dem Display darzustellen.
- Eine Drehung des Displays soll festgestellt werden und dementsprechend soll das Display im Landscapemodus geflippt werden (hier wurde der Anwendungsfall betrachtet, bei welchem es zwei Ausgangspositionen des Balancers gibt). Die Notwenige Sensorik dafür soll angesteuert werden

- P002: Drehdrucktaster

Der Drehdrucktaster ist für die Navigation zuständig

- Es soll möglich sein, mit dem Bildschirm in geeigneter Weise über den Drehdrucktaster zu interagieren. Dies kann das Wechseln der Anzeige zur Visualisierung verschiedener Sensoren sein, aber auch die Eingabe von Projektparametern

- P003: Sensorik

Durch die Sensorik werden die verschiedenen Funktionen des Balancers erst möglich

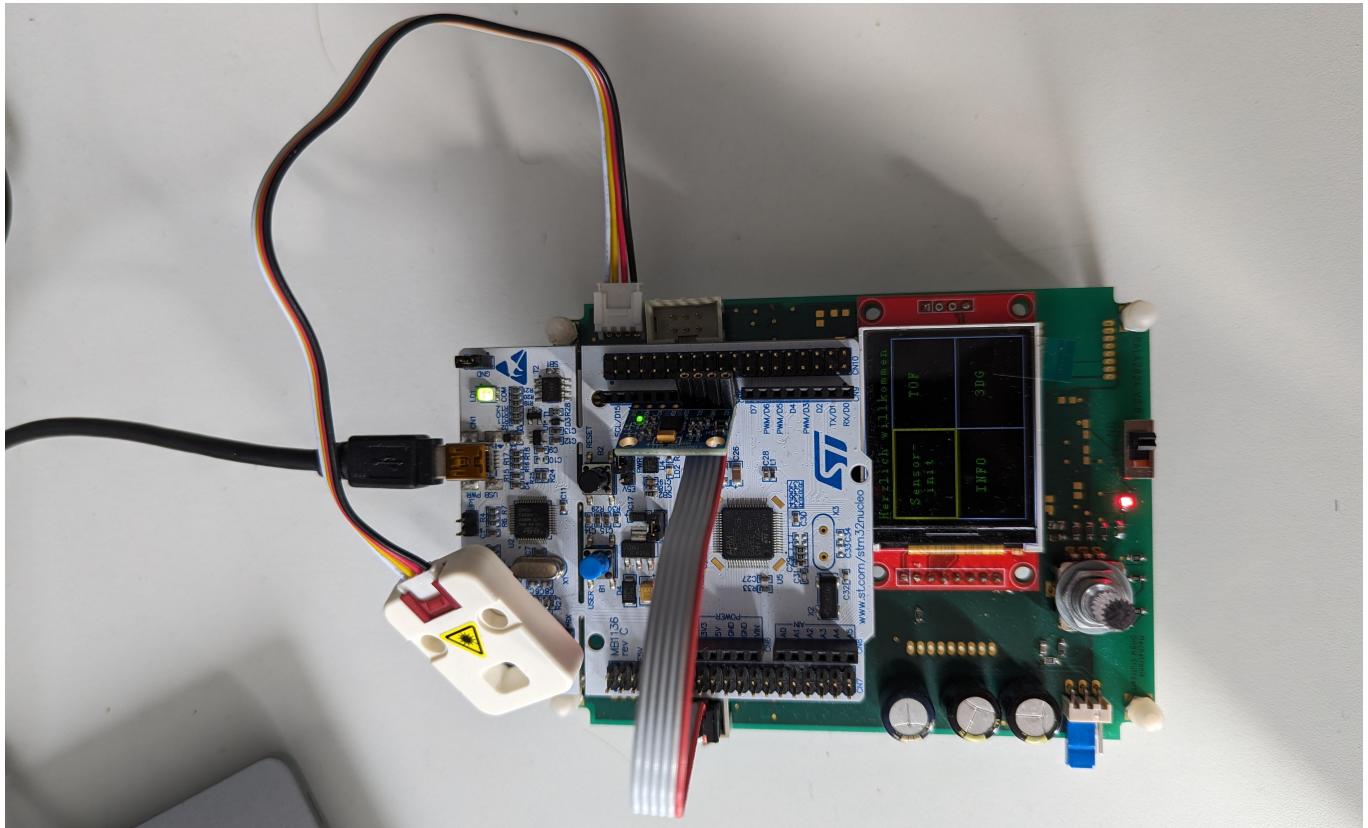
- Der Lidar soll über I2C angesteuert und ausgelesen, sowie das Ergebnis visualisiert werden.
- Der MPU6050 soll über I2C angesteuert und ausgelesen, sowie das Ergebnis visualisiert werden. Zudem soll dieser Sensor für die Bildschirmanzeigerichtung verwendet werden.
- P004: Dokumentation
Das implementierte Endergebnis soll verständlich dokumentiert werden.

Implementierung

Bei der Implementierung wurde auf die Bibliotheken MCAL, CMSIS sowie Balancer zurückgegriffen.

Die Implementierung ist im Ordner ./Bala24 unter dem STM-Projektnamen Drehgeber angelegt.

Ausgelegt und getestet wurde das System mit dem TOF VL53L0X Sensor am I2C1 und dem MPU6050 am I2C2. Das folgende Bild zeigt das Setup:



Bedienung

Das System interagiert über den Dreh-Druck-Taster mit dem Anwender. Im Hauptmenü kann man durch Drehen eines der vier Untermenüs auswählen und durch einen Tastendruck bestätigen. In den Untermenüs kann man durch einen Tastendruck wieder zurück zum Hauptmenü gelangen.

Main

Die Datei `./Bala24/src/main.c` sowie der zugehörigen Header-Datei `./Bala24/inc/main.h` beinhaltet das Beispiel-Programm und ist der Anfangspunkt für die Ausführung.

`int main(void)` : Diese Funktion beinhaltet die zeitliche Steuerung der Prozesse. Alle 10 ms werden die Daten des Drehdrucktasters eingeholt, der Seitenwechsel von Menü und Untermenüs durchgeführt und die Rotationswerte des 3DG Sensors berechnet. In diesem Timer werden auch die Systemausführungen durchgeführt, darunter die Sensorinitialisierung. Alle 100 ms wird der dynamische Inhalt des Bildschirms (Sensorwerte etc.) visualisiert. Eine blaue LED wird alle 250 ms getoggelt, dadurch kann der Anwender die zeitliche Auslastung des Systems sehen. Neue Sensorwerte des 3DG-Sensors werden alle 2 ms eingeholt.

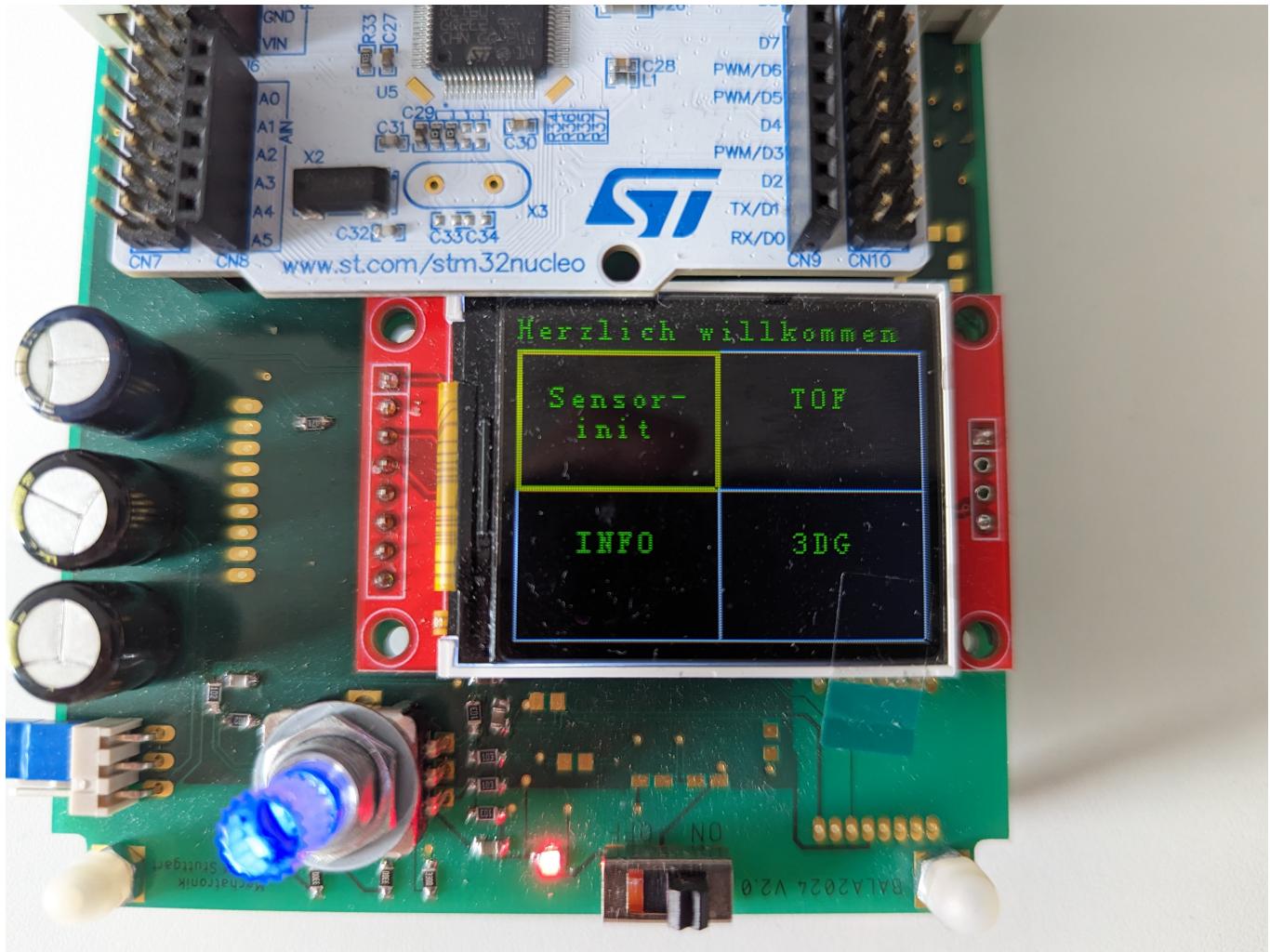
Des Weiteren beinhaltet dieses Programm Funktionen zur Initialisierung des Systems (`void initBala(void)`), des Submenüs (`void initSubMenu(SCREEN_PAGES_t page)`) sowie des Scans und der Initialisierung mittels I2C (`void i2cScanAndInit(I2C_TypeDef *i2c), uint8_t I2C_SCAN(uint8_t scanAddr, I2C_TypeDef *i2c)`).

Visualisierung

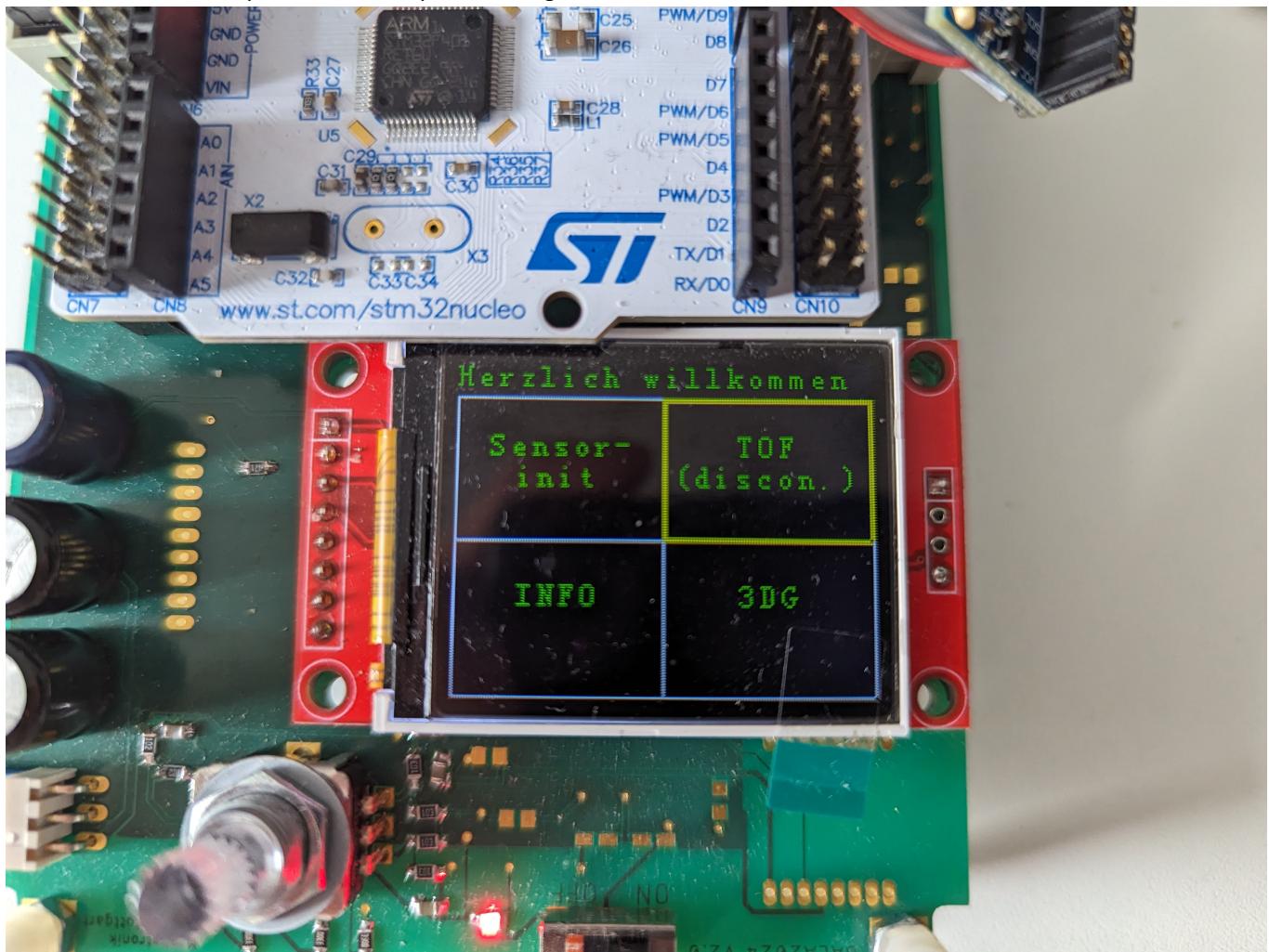
Die Ausgabe sämtlicher Visualisierung auf dem TFT-Display ist in der Datei `./Bala24/src/visualisation.c` sowie der zugehörigen Header-Datei `./Bala24/inc/visualisation.h` implementiert. Das Display wurde über die Balancer Library angesteuert. Mittels "Defines" wurden einheitliche Zeilen definiert und auch der Schwellenwert für das Flippen des Displays festgelegt. Im Folgenden wird das Menü und die Untermenüs vorgestellt:

Hauptmenü

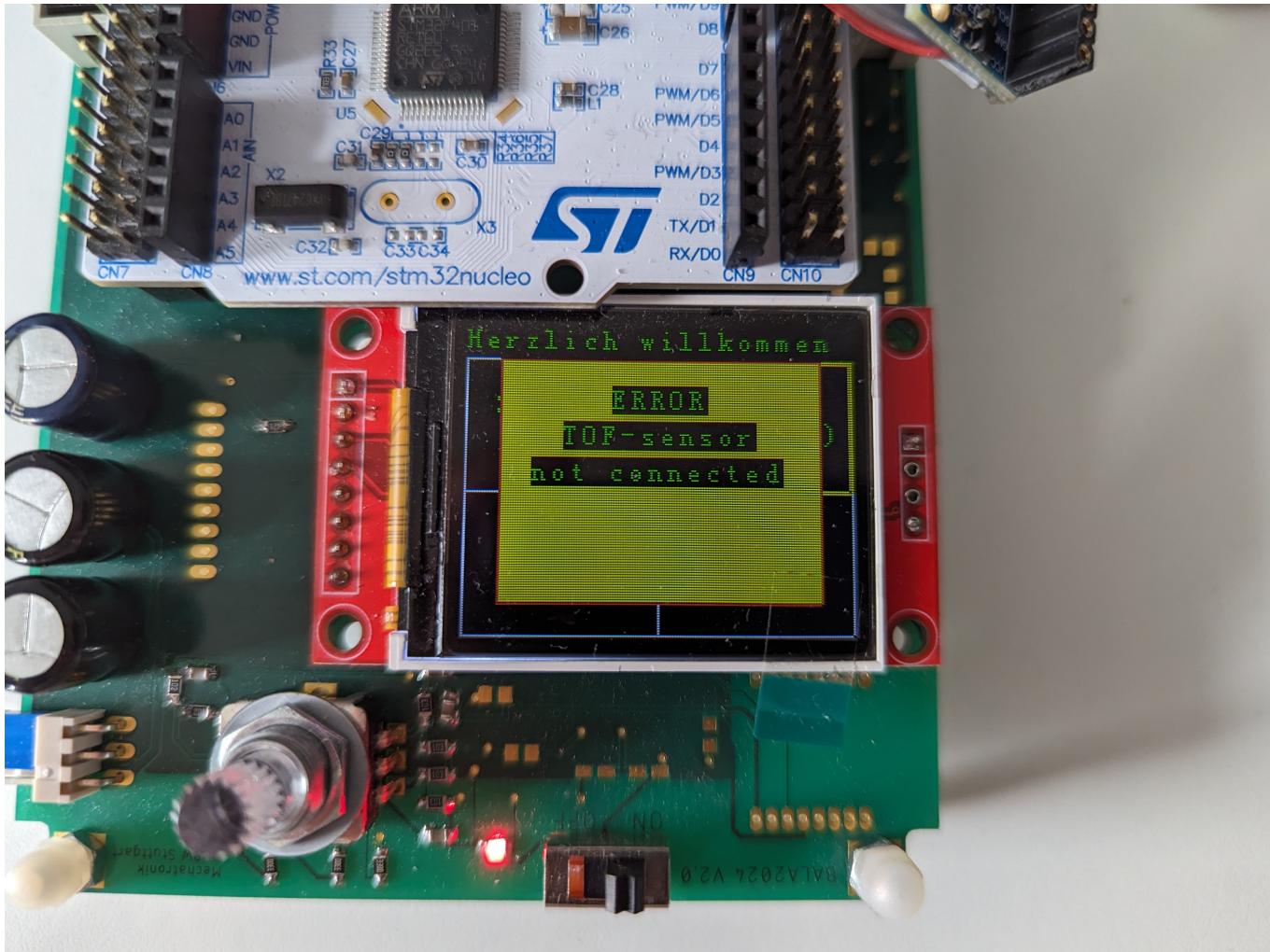
das Hauptmenü ist in vier Teile unterteilt, durch welche man durch Drehen des Dreh-Druck-Tasters durchwechseln kann. Auch sieht man hier, ob ein Sensor angeschlossen und initialisiert ist.



Wenn ein Sensor (hier der TOF) nicht angeschlossen ist:



Wenn man dann versucht, auf das Submenü des nicht initialisierten Sensors zuzugreifen kommt folgender Fehler:



Submenü: Sensor Initialisierung

Zuerst werden alle I2C Adressen durchsucht, wird ein Sensor gefunden, wird dieser initialisiert. Im Fall, dass es 2 I2C Schnittstellen gibt, werden beide nacheinander durchsucht und initialisiert.



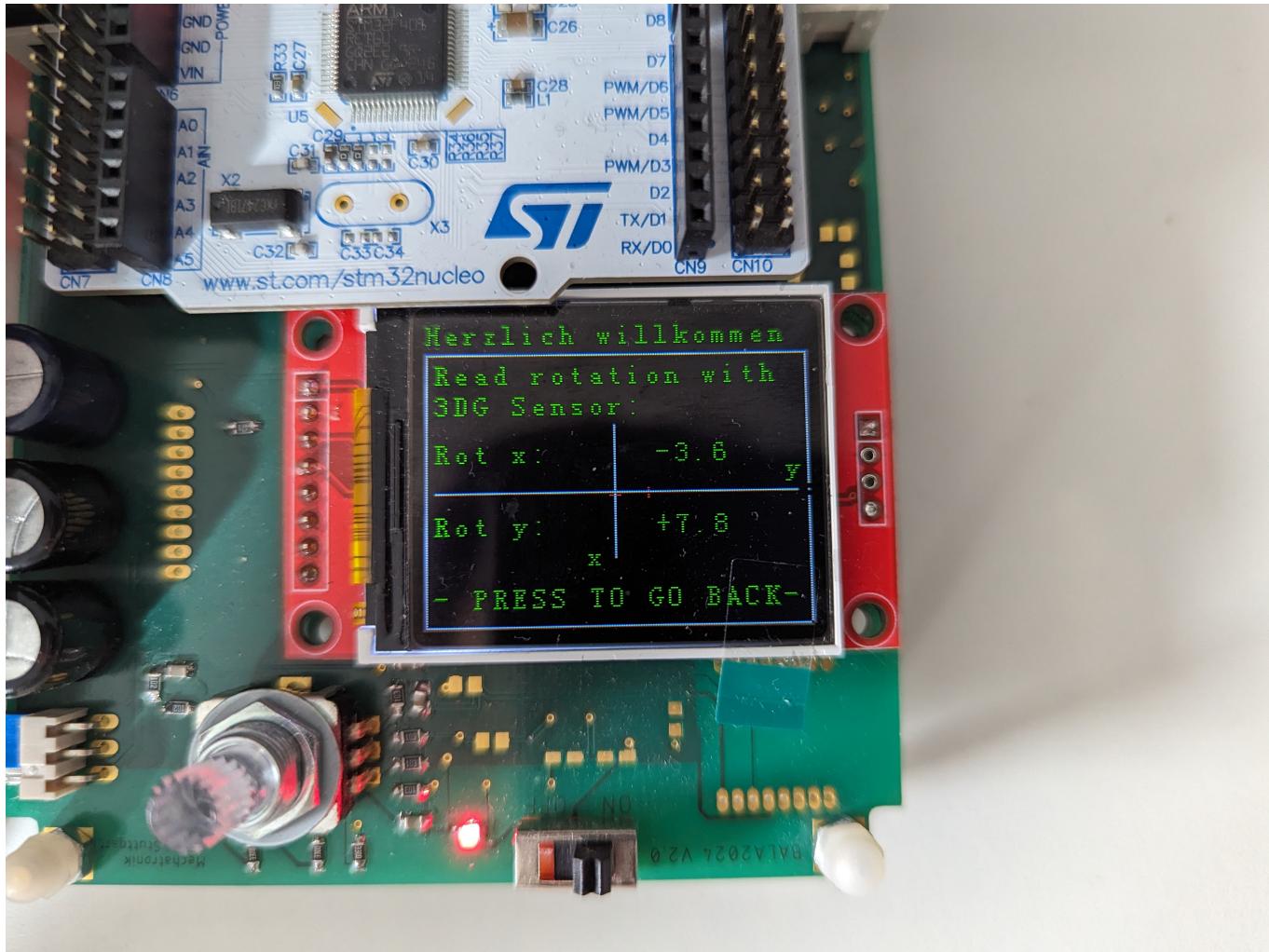
Submenü: TOF-Sensor

Dieses Untermenü zeigt den gemessenen Sensorwert des TOF-Sensors an. Dabei werden die Sensorwerte im kontinuierlichen Modus abgerufen.



Submenü: 3DG-Sensor

Hier werden die Rotationen in X und Y dargestellt. Zur besseren Visualisierung sollte sich der Sensor Achsengleich zur Visualisierung befinden. Durch rote Indikatoren auf dem weißen Balken wird neben dem Zahlenwert auch der Winkel grafisch visualisiert. Diese Visualisierung invertiert sich entsprechend bei geflippten Bildschirm, damit es noch immer zum Sensor passt.



Submenü: System-Info

Diese Seite gibt eine kleine Navigationsübersicht.



Sensor

Als Sensoren wurden der TOF-Sensor (VL53L0X) angesteuert sowie der MPU650. Beide werden über die `Balancer` Library angesteuert. Die vorliegende Gruppe hat die entsprechende Library für den TOF-Sensor im Balancer Projekt implementiert.

Balancer Library Erweiterung: SensorTOF

Die Implementierung des TOF-Sensors (Time of Light Sensor) ist in der Datei

`./Balancer/src/SensorTOF.c` sowie der zugehörigen Header-Datei

`./Balancer/inc/SensorTOF.h` implementiert. Sie basiert auf der Vorgehensweise gängiger

Bibliotheken und der offiziellen API von STM zum Sensor [VL53L0X \(link\)](#). Die Register konnten

nicht aus dem Handbuch entnommen werden, da dies keine detaillierte Beschreibung beinhaltet und wurde daher direkt aus dem Programmcode entnommen.

Als externe Funktionen gibt es eine Initialisierungsfunktion (`bool TOF_init(I2C_TypeDef *i2c, TOF_ADDR_t addr)`) sowie Funktionen zur Entfernungsmessung. Dabei kann man entweder eine einzelne Messung durchführen (mit der Funktion `bool TOF_ReadSingleDistance(uint16_t *range)`) oder im kontinuierlichen messen: Hierbei wird der kontinuierliche Modus gestartet (`bool TOF_startContinuous(uint32_t period_ms)`), dann können so lange Messwerte ausgelesen werden (`bool TOF_ReadContinuousDistance(uint16_t *range)`), bis der Modus wieder gestoppt wird (`bool TOF_stopContinuous()`).

Validierung

Im Nachfolgenden Bild ist der Test zu sehen. Durch das Kegelförmige Messen des Sensors, kann der Sensor nicht direkt am Lineal angelegt werden, dadurch kann eine kleine Ungenauigkeit in die Messung hinzukommen. Bei dem 300 mm langen Lineal konnten in diesem Aufbau 297 mm gemessen werden.

