November 2020

„3in1“
Messgerät

by strgaltdel
aka unow
Revision 1.0

- Schwerpunktwaage
- EWD Waage
- Ruderausschlag

Inhaltsverzeichnis

Allgemeine Intention:

Seit einiger Zeit häufen sich im Flächen-RC Bereich DIY Projekte, die elektronische Schwerpunktwaagen, EWD Messgeräte und Ruderausschlagssensoren vorstellen.

Diese Projekte nutzen häufig gleiche (IMU) Sensoren bzw gleiche oder ähnliche Microcontroller. Abgesehen von den Implementationen, bei denen die GUI auf ein Handy umgesetzt wurde, ist die Bedienmöglichkeit relativ einfach gehalten.

Die „3in1“ Lösung greift die Idee auf, diese Einzelprojekte in ein Gerät in etwa kostenneutral zu vereinen. Anstelle mehrerer kleiner z.B. Arduino-Nano-Microcontroller und einfacher 16x2 LCD Shields wird ein etwas leistungsfähigerer Arduino Mega zur Steuerung sowie ein TFT Touch Panel zur Bedienung verwendet.

Die erweiterten Hardwareressourcen werden genutzt um z.B. die üblicherweise „externe“ Kalibrierroutinen der IMUs zu integrieren und die Ergebnisse ins Eeprom des Microcontrollers abzuspeichern.

Es besteht auch die Möglichkeit zwei unterschiedliche „Heck“ Meßwaagen zur EWD Messung von z.B. konventionellen und zusätzlich V-Leitwerken zu verwalten.

Es verbleiben nach Fertigstellung des Codings noch genügend Ressourcen, um weitere Ideen / Funktionalitäten in der Zukunft umzusetzen.

Das Projekt ist für den Open Source DIY Bereich vorgesehen.

Ich bedanke mich unbekannterweise bei den Entwicklern der bereits existierenden Projekte der RC-Network und RC-Groups Foren, die eine hervorragende Inspiration bei der Umsetzung waren.

Der besondere Dank gilt Paul Poschen für die umfangreiche Entwicklung und Bereitstellung der 3 D Plots.

Dieses Manual gliedert sich zunächst in der Beschreibung zur Bedienung der Oberfläche und danach in eine Anleitung den Arduino zu flashen bzw den Code zu Compilieren,
gefolgt von Hinweisen zum Bau des Meßgerätes (Kopfstation und Meßapparate)

Im Anhang befinden sich Verdrahtungsschemata und Teileliste

Unter Anmerkungen sind ergänzende Hinweise zu bestimmten Komponenten zu finden.

Das Projekt (Coding, Plots..) kann unter folgender Adresse heruntergeladen werden

https://github.com/strgaltdel/3in1

Komponenten

Das Gerät besteht aus den Komponenten „Headunit“, den EWD Meßschiebern, der Schwerpunktwaage und den Ruderwegsensoren

Headunit

Die Headunit ist die Steuereinheit.
Hier werden alle Sensoren angeschlossen.
Als Microcontroller dient ein Arduino Mega

Die Bedienung geschieht durch ein 2.8Zoll Touch Display
Es wurde ein Typ ausgewählt, der direkt als „shield“ auf die Pfostenleiste des Arduinos gesteckt werden kann.
Die Grösse ist so ausgesucht, daß sie zum einen genügend Fläche und Auflösung besitzt, zum anderen auf dem Arduino noch genügend frei zugängliche Pins übrig lässt um ein Connectorshield aufzusetzen.
Die mit dem Coding getesteten Displays kommen von Banggood (Geekcreit).

Stromversorgung ist durch eine 9V Blockbatterie vorgesehen, es kann via USB eine Powerbank genutzt werden.

Als Gehäuse dient ein DIY 3-d Druck Gehäuse bzw fertige Acrylglas-Arduino Gehäuse, wobei dann aber Controller und Touch Display recht offen liegen.

Als Konnektoren für die Lagesensoren (EWD/Ausschläge) sind GX 12 Stecker vorgesehen
(benötigt werden 4 Pins)
Die Schwerpunktwaage wir über einen GX 16 Konnektor verbunden
(benötigt werden 6 Pins)

Durch diese Wahl wird ein falsches Anschließen von Waage bzw Lagesensoren ausgeschlossen.

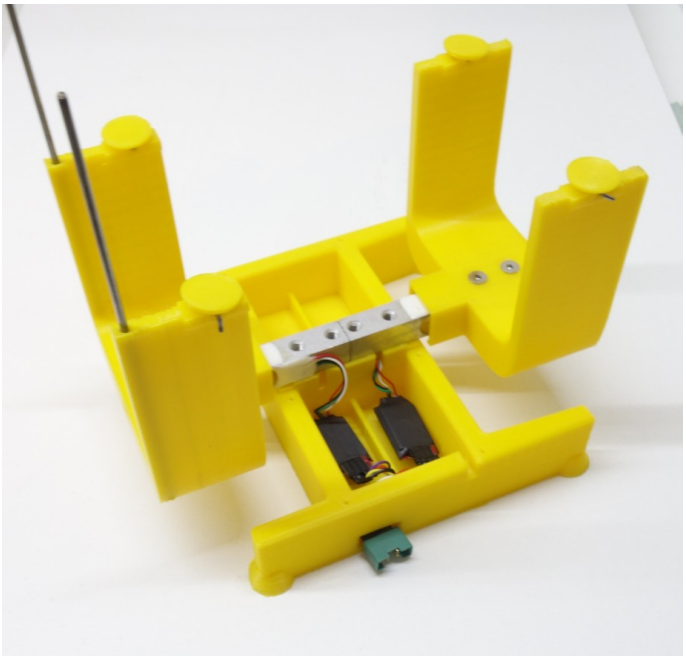
Bsp mit DIY Gehäuse ohne Front

Schwerpunktwaage

Hier werden bereits bewährte 3d DIY Plots aus den einschlägigen Foren verwendet.
Die Waage wird ausgestattet mit den entsprechenden Wägezellen nach Bedarf
(typisch 1,2,3,5 kg Meßbereich)
und den dazugehörigen HX511 Boards.

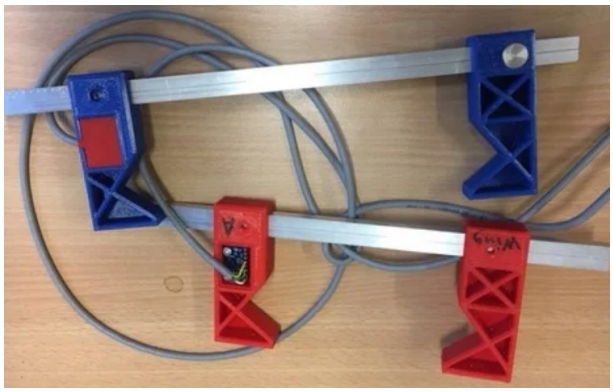
Mehr Elektronik ist in den Waagen nicht notwendig.

Die Verkabelung zur Headunit kann direkt an die HX511 Boards gelötet werden, oder z.B. über einen „Multiplex Stecker“ mit 6 polen trennbar gestaltet werden.
Etliche vorhandene CG Waagen haben Aufnahmen für MPX Steckersystem



EWD-Waage

Hier werden bereits bewährte 3d DIY Plots thingiverse einschlägigen Foren verwendet.
https://www.thingiverse.com/thing:2938217

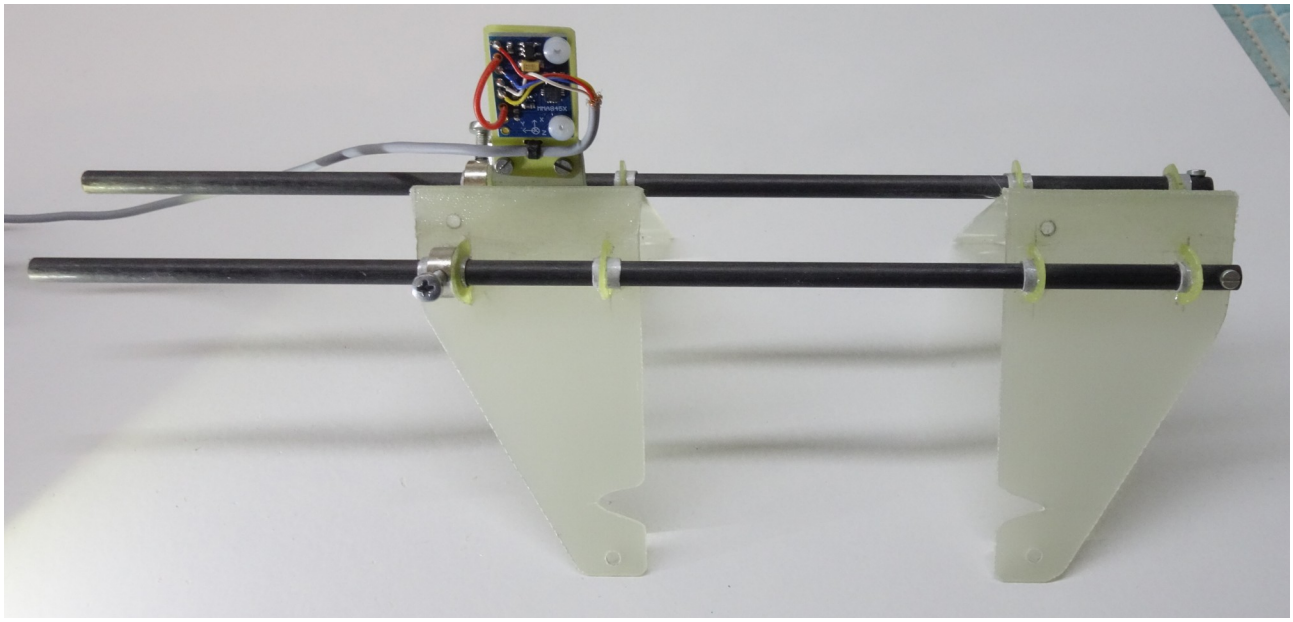


Die Waage wird ausgestattet mit den entsprechenden Lagesensoren vom Typ „GY 521“.

Für die Verkabelung sollte ein möglichst leichtes und flexibles 4-Draht Kabel verwendet werden.
Meine Empfehlung lautet Tonarmkabel.

Für die Messung von V-Leitwerken benutze ich einen Eigenbau.
Das Prinzip funktioniert entsprechend konventioneller EWD Waagen,
nur das über eine Art „Prismenschablone“ das Leitwerk eingespannt wird.

Ein Bild sollte die Funktionsweise erklären:

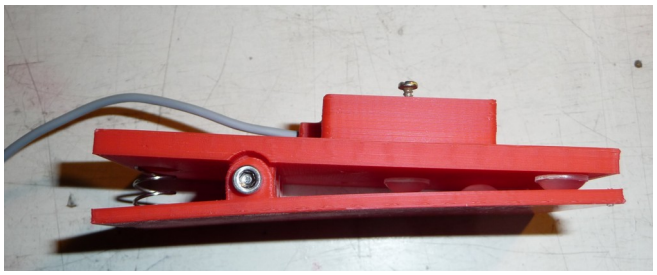


Wichtig ist, dass die vier Aufnahmepunkte / Aussparungen für Nasen und Endleiste exakt eine Ebene aufspannen.
Für die „Nullung“ vor der Messung ist es ebenfalls wichtig, das alle vier Punkte identische Abstände nach unten zum jeweiligen Auflagepunkt besitzen, da es sonst zu einem Offset zum Flächen-Meßschieber kommt.

Leider gibt es dazu keinen 3d Plot

Ruderwegmessung

Hier wird eine Art Klammer verwendet, die ein Kollege entwickelt hat.
DIY 3-D Plot Daten liegen vor



Es wird der gleiche GY-521 Sensortypen wie bei den EWD Waagen genutzt.
Es sollte ebenfalls das gleiche Kabel wie zuvor verwendet werden (leicht & flexibel)

Inbetriebnahme

Vor dem Einschalten müssen die für die jeweilige Messung notwendigen Messgeräte, also entweder

- Waage
- EWD Meßschieber oder
- Ruderweg Sensoren

angeschlossen werden.
Der Microcontroller initialisiert während des Einschaltens die gefundenen Sensoren, so dass ein nachträgliches Anschliessen ggf. zu Problemen führt.

Es muss sichergestellt werden, dass der Arduino mit einer ausreichenden Stromquelle versorgt wird. Der Arduino Mega kann über einen Hohlstecker mittels einer Versorgungsspannung zwischen 7-20V betrieben werden. Alternativ zu einem Steckernetzteil bietet sich z.B. ein 9V Batterieblock an.
Als weitere Möglichkeit kann der Mega auch über den USB Anschluss, z.B. mittels einer PowerBank, gespeist werden.

Aufgrund nicht vorhandener Restwelligkeiten ist eine Versorgung via Batterie/Akku zu empfehlen.

Vor den ersten Messungen müssen die Konfigurationsdaten der Schwerpunktwaage im entsprechenden Menü hinterlegt und die jeweiligen Sensoren über die jeweiligen Menüpunkte kalibriert werden.

Hauptmenu

Funktion

Nach Einschalten der Headunit und Initialisierung der Sensoren erscheinen die Buttons um die jeweiligen Untermenüs der drei Meßmöglichkeiten zu verzweigen

SP Waage

(Schwerpunktwaage)
Durch Drücken des Buttons wird in das Menü zur Messung von Schwerpunkt und Gewicht verzweigt.

Hinweis:
In der Konfiguration der Schwerpunktmessung kann das Einheitensystem zentral von metrisch auf imperial, also mm<>inch, Gramm<>Unze, umgestellt werden.
Diese Einstellung gilt dann übergreifend für alle Messungen !

EWD

(EWD Messung)
Hier wird in das Menü zur Messung der Einstellwinkeldifferenz verzweigt

Ruder

(Ruderausschlag Messung)
Hier wird in das Menü zur Messung von Ruderausschlägen verzweigt

Schwerpunktwaage

Anzeige

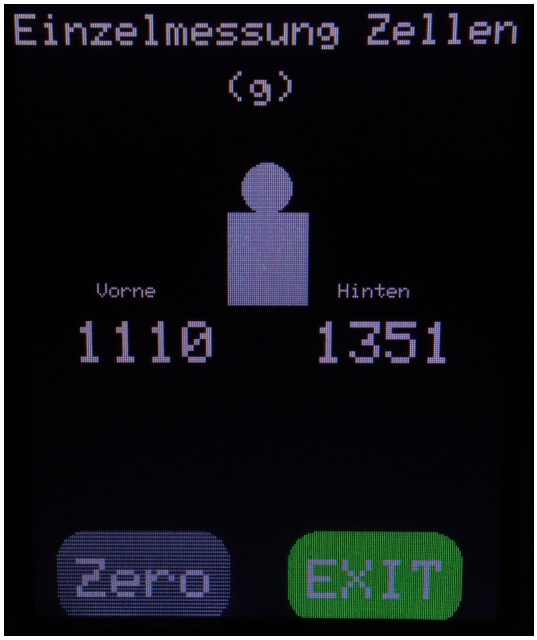
Die Bedienoberfläche zeigt links oben den Schwerpunkt, rechts oben das Gesamtgewicht an.

Button „Zero“

Vor Nutzung muss mittels „Zero“ das Tara Gewicht ermittelt werden, danach kann das Modell aufgelegt werden und nach kurzer zeit stabilisiert sich Anzeige.

Button „Rohdata“

Über den Button „Rohdata“ erhält man Angaben darüber, wie sich das Gewicht jeweils auf die vordere & hintere Messbrücke aufteilt. Diese Information gibt einen Eindruck davon, ob die für die jeweilige Waage gewählten Kraftsensoren wirklich in einem günstigen (linearen) Meßbereich liegen und nicht schlimmstenfalls eine Messung außerhalb des Endwertes des Sensors vorliegt.



Button „Exit“

Hiermit wird das Menue verlassen

Button „config“

Das Menü „Config“ ermöglicht es folgende Einstellungen der Waage vorzunehmen:

Auflage

- der Abstand der Auflagepunkte zwischen den Gabeln der Waage.
Meistens haben die Waagen eine bewegliche Auflagefläche, man misst dann den Abstand der Achsen.

Anschlag

- der Abstand vom Anschlag der Waage zum vorderen Auflagepunkt

Gewicht

- Masse des Referenzgewichtes, mit dem die Kalibrierung der Wiegezellen durchgeführt wird.
Empfehlung: Gewicht sollte in etwa dem halbem Messbereichsendwert einer Zelle entsprechen.

Mass

- Umstellung zwischen metrischen und imperialen Einheitensystem **für die gesamte** Messeinrichtung.

Edit

- über die edit Taste kann der jeweilige Wert geändert werden

Button „calibrate“

Das Menü „Calibrate“ führt in die Kalibrierungsroutine der Waegezellen.
Dazu benötigt man ein Referenzgewicht, dessen Masse zuvor in der Konfiguration angegeben wurde.
Jede Auflagegabel wird zunächst leer, anschliessend mit Referenzgewicht gemessen.
Daraus generiert das Programm die „Kennlinie“ des jeweiligen Meßaufnehmers.
Die Menueführung ist selbsterklärend.
Die Routine kann beliebig oft wiederholt werden.

Folgende Schritte werden nacheinander durchgeführt:

(1) Messung der vorderen Auflage ohne Last

(2) Messung der vorderen Auflage mit Referenzgewichtes

(3) Messung der hinteren Auflage ohne Last

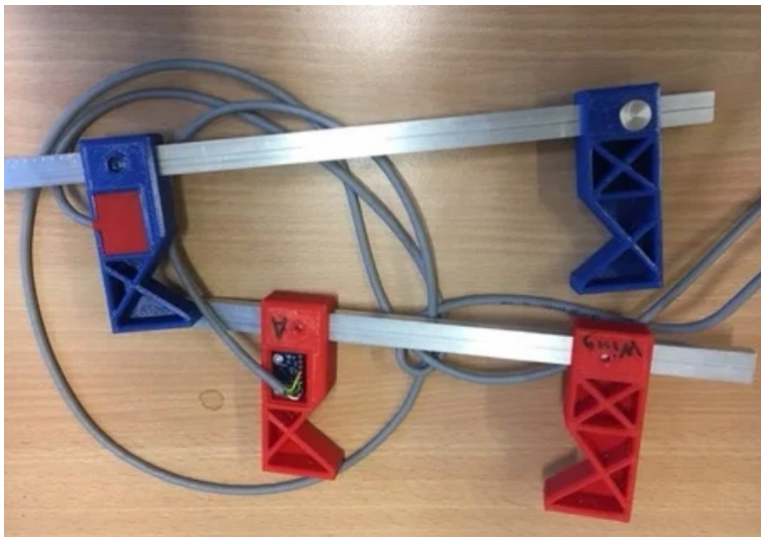
(4) Messung der hinteren Auflage mit Referenzgewichtes

Es versteht sich von selbst, dass die Masse des Referenzgewichtes zuvor unter /config /Gewicht korrekt eingetragen wurde !

EWD

Funktion

In diesem Untermenue kann die EWD eines Modells vermessen werden.
Es müssen zwei Meßsensoren an der Headunit angeschlossen sein.



(Bildquelle: thingiverse)

Zunächst werden die beiden „Meßschieber“ stehend auf eine Ebene Platte gestellt und der Button „Zero“ gedrückt. Damit erreicht man die Referenzierung der „Null Ebene“, also die Messung von 0 Grad bei gleicher Sensorausrichtung.

Danach setzt man den Grösseren Meßschieber an der Fläche an, den kleineren ans Leitwerk.

Die EWD Waage zeigt entsprechend die EWD an.

Die Messung über zwei Schieber hat den Vorteil, dass die Lage des Modell geändert werden kann, ohne die Messung zu beeinflussen.

Die Headunit verwaltet zwei Messchieber für das Leitwerk, um getrennte Schieber für konventionelle und V Leitwerke ansteuern zu können

Anzeige

Die Bedienoberfläche zeigt im oberen Bereich die Einzel-Messergebnisse an.
Zentral wird der Differenzwert in Großschrift dargestellt.

Sollte der Wert ausserhalb typischer Werte (0.5-2.0 Grad) liegen wechselt die Farbe nach rot

Button „Zero“

Vor Nutzung muss wie bereits oben beschrieben die Ausrichtung der Meßschieber „synchronisisert“ werden.
Dazu werden die Schieber senkrecht auf eine Ebene Fläche gestellt und „Zero“ gedrückt.
Danach sollten die Anzeigewerte um 0,0 herum minimal schwanken („Meßrauschen“)

Button „Std <> V“

Durch den Button „Std <> V“ wird der Leitwerkssensor umgeschaltet zwischen Standard und V-Ltw Messung.
Es werden die jeweils zum passenden Sensor hinterlegten Kalibrierungsdaten angewendet um eine möglichst präzise Messung zu ermöglichen

Button „calibrate“

Das Menü „calibrate“ ermöglicht es die verwendeten Sensoren zu kalibrieren.
Wenn die Kalibrierungsroutine gestartet wurde muss der Sensor um alle drei Achsen in jeweils beide Endpositionen (also links/rechts; vorne/hinten; oben/unten) verbracht werden und pro Position mindestens eine Messung durchgeführt werden.
Mehrfachmessungen einer Position sind möglich.

Die drei gedachten „Achsen“ müssen möglichst exakt, wie in einem karthesischen Koordinatensystem, im Winkel von 90 Grad zueinander stehen.

Am praktikabelsten ist es eine Art Würfel zu benutzen, an dem auf einer Ebene der Sensor planparallel montiert wird.
Dann die Messung „auf allen 6 Flächen“ des Würfels durchführen.

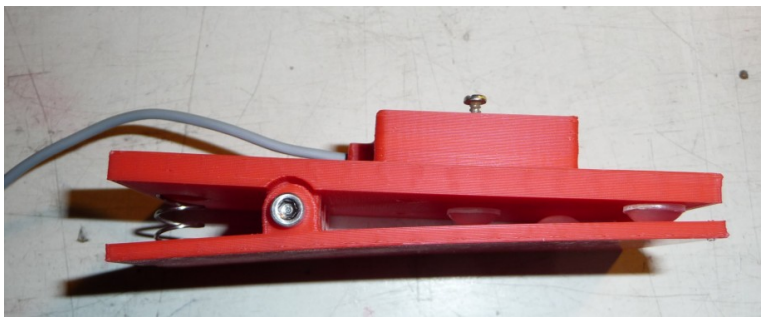
Button „Exit“

Hiermit wird das Menue verlassen

Ruderweg

Funktion

In diesem Untermenue können die Ruderausschläge eines Modells vermessen werden.
Es können zwei Meßsensoren an der Headunit angeschlossen sein.
Die Meßsensoren sind gleichen Typs wie die EWD Sensoren und können daher mit den gleichen Ports wie die EWD „Schieber“ verbunden werden



Idealerweise werden die beiden „Klammern“ zusammen auf eine Ebene Platte „geheftet“ und initialisiert.
Damit erreicht man die Referenzierung der „Null Ebene“, also die identische Winkelmessung bei gleicher Sensorausrichtung.

Danach setzt man Klammern an die jeweils zu messenden Ruder.
Die Ruder sollten i.d.R. so eingestellt sein, dass sie identische Winkel anzeigen.
Danach wird nochmals „Zero“ gedrücktz um diese Position als Neutralstellung zu reklamieren.

Anzeige

Das Instrument zeigt dann im oberen Bereich entsprechend die Ausschläge / Winkelpositionen der Ruder an. Dargestellt werden Winkel in Grad und Ausschlag in mm (bei metrischer Einstellung)

Im mittleren Bildbereich wird die Winkeldifferenz der beiden Sensoren dargestellt.

Button „Zero“

„initialisiert“ die Sensoren, dass diese Momentanposition als Neutralstellung als 0-Grad Messung zugrunde gelegt wird

Button „set Flap“

Hier kann bei Bedarf die Rudertiefe angegeben werden um eine Ausschlagsmessung als Strecke (Weg nach oben / unten) kalkulieren zu können

Button „thrw<>sec“

Wahl der Wegmessung als „Sekante“ bzw „Tangente“

Button „calibrate“

Das Menü „calibrate“ ermöglicht es die verwendeten Sensoren zu kalibrieren.

Wenn die Kalibrierungsroutine gestartet wurde muss der Sensor um alle drei Achsen in jeweils beide Endpositionen (also links/rechts; vorne/hinten; oben/unten) verbracht werden und pro Position mindestens eine Messung durchgeführt werden.
Mehrfachmessungen einer Position sind möglich.

Die drei gedachten „Achsen“ müssen möglichst exakt, wie in einem karthesischen Koordinatensystem, im Winkel von 90 Grad zueinander stehen.

Am praktikabelsten ist es eine Art Würfel zu benutzen, an dem auf einer Ebene der Sensor planparallel montiert wird.
Dann die Messung „auf allen 6 Flächen“ des Würfels durchführen.

Button „Exit“

Hiermit wird das Menue verlassen

Aufbau

Arduino flashen, die einfache Methode

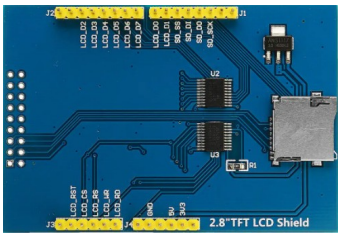
Häufig tauchen bei etwas komplexeren Arduino Projekte Probleme beim Compilieren auf, weil Bibliotheken nicht oder in falscher Version, oder von falscher Quelle heruntergeladen wurden.

Sollte kein Bedarf vorhanden sein den Quellcode individuell anzupassen, ist die schnellste Methode zum Erfolg einfach nur das benötigte Hex (binary) File direkt auf den Arduino hochzuladen.

Ich habe daher zwei Hex-Files im Ordner „bin“ abgelegt.

Das „std“ läuft auf dem „Banggood“ Display mit gelber Stiftleiste,

https://www.banggood.com/de/2_8-Inch-TFT-LCD-Shield-Touch-Display-Screen-Module-Geekcreit-for-Arduino-products-that-work-with-official-Arduino-boards-p-989697.html?rmmds=search&cur_warehouse=UK



das „set“ läuft auf dem banggood Display, das als set inklusive Arduino Uno & 2.4“ Display kommt.

https://www.banggood.com/de/Geekcreit-UNO-R3-Improved-Version-+-2_8TFT-LCD-Touch-Screen-+-2_4TFT-Touch-Screen-Display-Module-Kit-Geekcreit-for-Arduino-products-that-work-with-official-Arduino-boards-p-1428291.html?rmmds=detail-top-buytogether-auto&cur_warehouse=UK

Der Upload erfolgt mit wenigen Klicks durch das Programm „XLOADER“
https://github.com/binaryupdates/xLoader

Code Compilieren und Arduino flashen

Wer den Code anpassen und compilieren möchte muss die üblichen Schritte für die Arduino-IDE Entwicklungsumgebung durchführen:

(1)
Der Source Code bzw. die Quelldateien werden in ein Projektverzeichnis kopiert.
Das Verzeichnis muss den gleichen Namen wie das Hauptprogramm besitzen (hier „3in1“)

(2)
Alle notwendigen Zusatz-Bibliotheken (libraries) müssen der IDE zur Verfügung gestellt werden.
Ein erfolgreiches Kompilieren ist sonst nicht möglich.

Ein Weg ist es über den Bibliotheksmanager zu gehen,
ein anderer der direkte Download, Entpacken und Kopieren in das Bibliotheksverzeichnis.

Die notwendigen Bibliotheken werden in der „3in1“ App durch die Datei „includes_main.h“ in das Coding eingebunden.

Hier Auszugsweise die notwendigen „addon“ Bibliotheken sowie deren Quellenangaben

#include <Wire.h>	// imu communication	https://github.com/jrowberg/i2cdevlib
#include <TouchScreen.h>	// touch support	https://github.com/adafruit/Adafruit_TouchScreen
#include <Adafruit_GFX.h>	// used for rounded buttons	https://github.com/adafruit/Adafruit-GFX-Library
#include <MCUFRIEND_kbv.h>	// flexible lcd support	https://www.arduinolibraries.info/libraries/mcufriend_kbv
#include <Filters.h>	// Filter classes for smoothing IMU data	https://github.com/JonHub/Filters/blob/master/Filters.h
#include <HX711.h>	// load cell support	https://github.com/bogde/HX711
#include <MovingAverage.h>	// simple way to build moving avg/used in cg	https://github.com/sofian/MovingAverage
#include <I2Cdev.h>	// some i2c, mpu6050 required methods	https://github.com/jrowberg/i2cdevlib

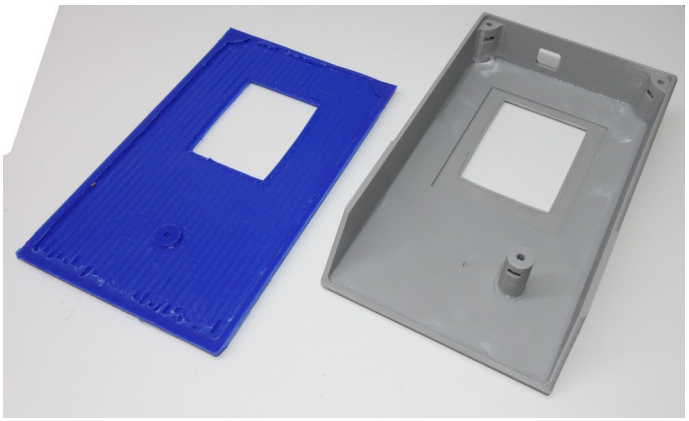
Wurden alle notwendigem Zusatzbibliotheken eingebunden sollte der Compiliervorgang fehlerfrei durchlaufen.
Der Arduino Mega kann geflasht werden.

Anmerkung:
In einer aktuellen Version der Adafruit-GFX lib kam es bei mir zu einem Compilierfehler da die Adafruit_I2CDevice.h Datei nicht existierte.
Ursache ist ein include in den oled libs, es reicht die beiden Dateien „Adafruit_GREYOLED…“ umzubenennen, damit diese nicht mit eingebunden werden.
Die Dateien sind für dieses Projekt nicht notwendig

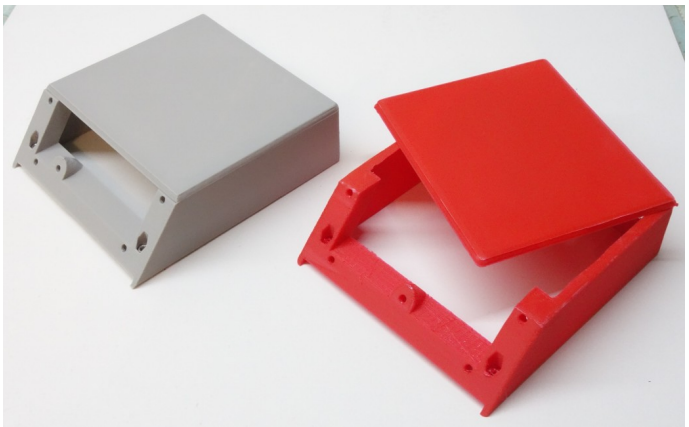
Zusammenbau Headunit Gehäuse

Das für das Projekt durch P. Poschen entwickelte Gehäuse wurde auf effizientes Plotten hin optimiert.
Die Einzelteile sind von der Konstruktion so gestaltet, dass möglichst geringe Druckzeiten notwendig sind.
Daher ist eine Klebung von zwei Deckeln und einem Distanzstück notwendig.
Als Klebstoff kommt Sekundenkleber bzw UHU hart zum Einsatz
Zuvor muss das das „Connectorshield“ angefertigt worden sein, Verdrahtung siehe unten.

(1) Frontblende verkleben
Zunächst wird die Frontblende auf den zugehörigen Rahmen verklebt
Anschliessend der Distanzbolzen, der die Frontblende im unteren Bereich mit dem Sockel verschraubbar macht, in die vorgesehene Position verklebt.



(2) Sockel mit Deckel verkleben
Nun kann der Deckel (das Teil mit der Batteriehalterung für 9V Blockbatterien) mit dem Sockel verklebt werden.



(3) Rückwand (Arduino Halterung) mit Sockel verschrauben
Es werden vier M3 Muttern in die vorgesehenen Positionen des Sockels geschoben (ggf mit Sekundenklebstoff fixiert), anschliessend wird die Rückwand mit dem Sockel verschaubt

(4) Ardunino auf Rückwand schrauben

(5) Display und Connector shield auf den Arduino Mega stecken

(6) Stromversorgung
Hohlstecker mit Ein-/Auschalter und 9V Batteriekabel verlöten
Schalter einbauen

Zwischenergebnis:

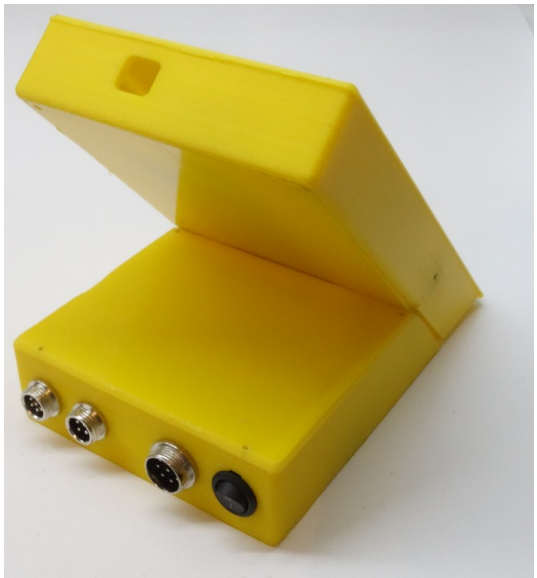
(7) GY 521 Verkabelung (Ruder & EWD Messung)
4pol Kabel mit Dupont Stecker auf die erste Gx12 Buchse löten & Kabel an den zweiten Connector weiter verlöten
GX12 Buchsen einbauen und verschrauben
Alternativ kann auch anstelle der Steckverbindung direkt auf das shield gelötet werden.

(8) HX 711 Verkabelung (Schwerpunktwaage)
6pol Kabel mit der GX16 Buchse verlöten
GX16 Buchsen einbauen und verschrauben

(9) Verbindungen herstellen
Alle Stecker aufstecken & Funktionstest durchführen

Zwischenergebnis (Perspektive von unten auf den Sockel):

(10) Frontblende mit zwei M3 Verschaubungen oben, und einer M3 Verschraubung unten befestigen.
Die M3 Verschraubung unten erfolgt schräg durch den Sockel, in den Aufnahmen der Frontblende befinden sich Platzierungen für M3 Muttern



(11) Bodenelement mit Sockel verschrauben und Gummifüsse aufkleben - Fertig -

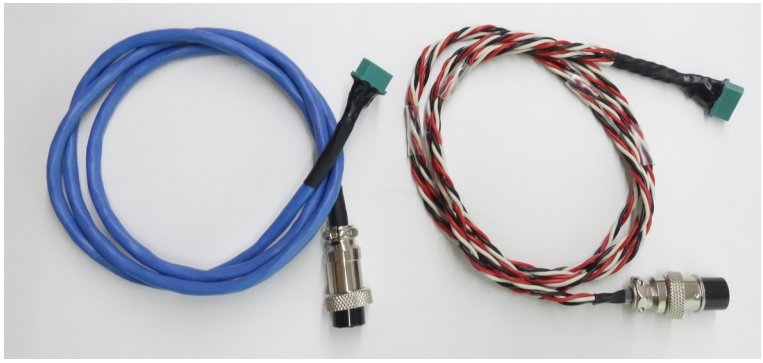
Verkabelung (extern)

(1) Schwerpunktwaage

An der „Headunit im Gehäuse“ kommt ein GX16 Stecker zum Einsatz.
Verwendet man den Mega2560 im Acryl-Gehäuse ist der Phantasie keine Grenze gesetzt.

Es werden 6 Litzen benötigt, eine Länge von ca einem Meter hat sich bewährt.
Wie immer sind flexible Kabel bei der Handhabung angenehmer.
Als Modellflieger hat man meistens 3 adrige Servokabel als Meterware vorrätig (am besten Silikonkabel), diese sind gut geeignet.
Optisch „netter“ sind komplett ummantelte Kabel. Da keine hohen Ströme fließen bieten sich Signalkabel mit geringem Querschnitt an. Ein gutes Beispiel dafür sind Patchkabel aus dem Netzwerkbereich.
Aber Achtung, es gibt hier von starr bis einigermassen flexibel erhebliche Unterschiede in der Biegesteifigkeit.

Die einfachste Art die Schwerpunktwaage mit der Headunit zu verbinden ist es, das Kabel direkt an die HX711 Sensorplatine zu löten.
Alternativ kann man am Durchbruch der Schwerpunktwaage auch z.B. eine MPX Steckverbindung einsetzen.



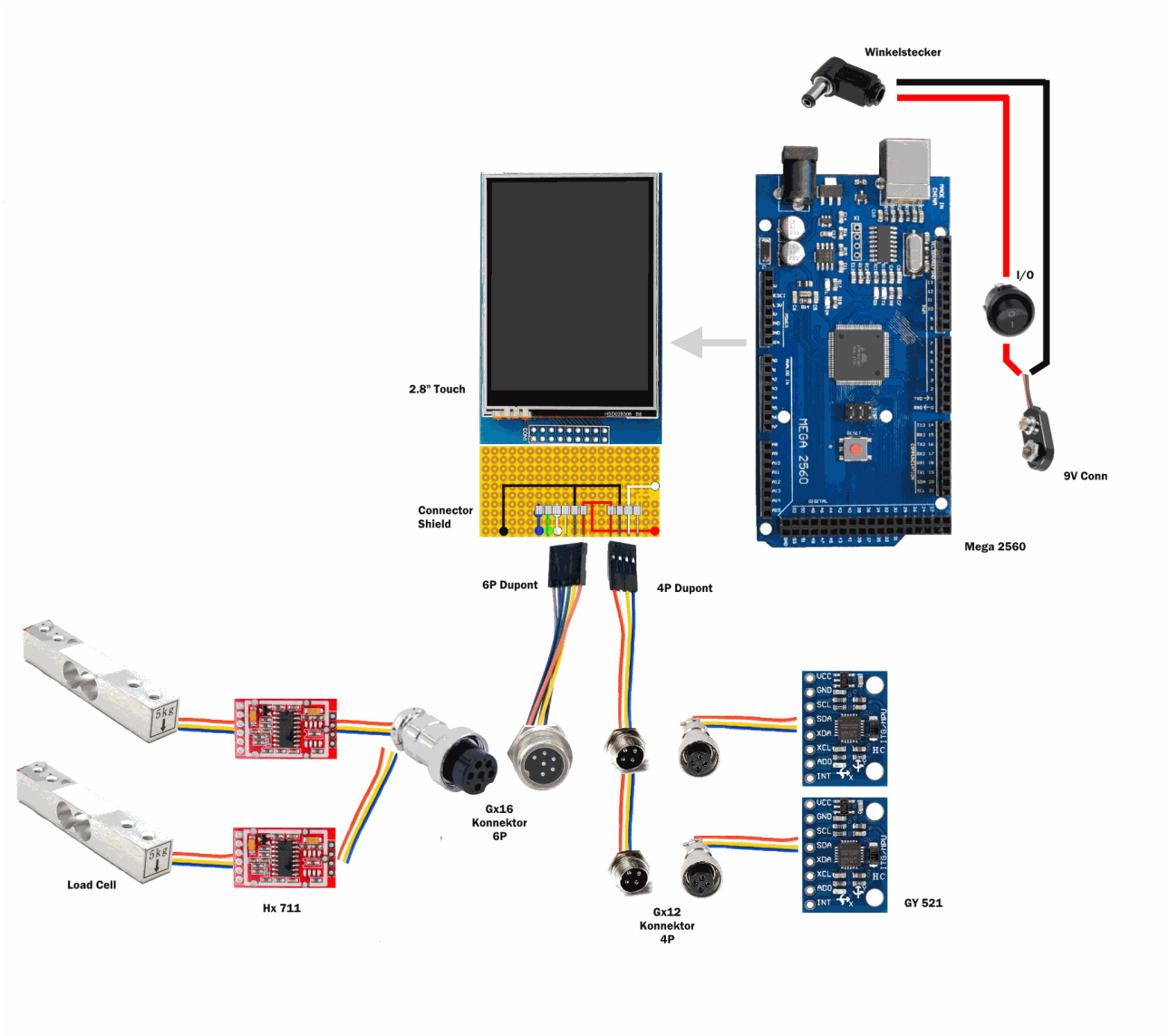
(2) EWD und Rudersensoren

An der „Headunit im Gehäuse“ kommt ein GX12 Stecker zum Einsatz.
Es werden jeweils 4 Litzen pro Sensor verwendet.
Die Länge sollte den jeweiligen Modellgrössen entsprechend angepasst sein, rnd 80 cm sollten für EWD Messungen ausreichen, 1...1,6 Meter sind typisch für Rudermessungen z.B. für Seglermodelle bis 5m Spannweite

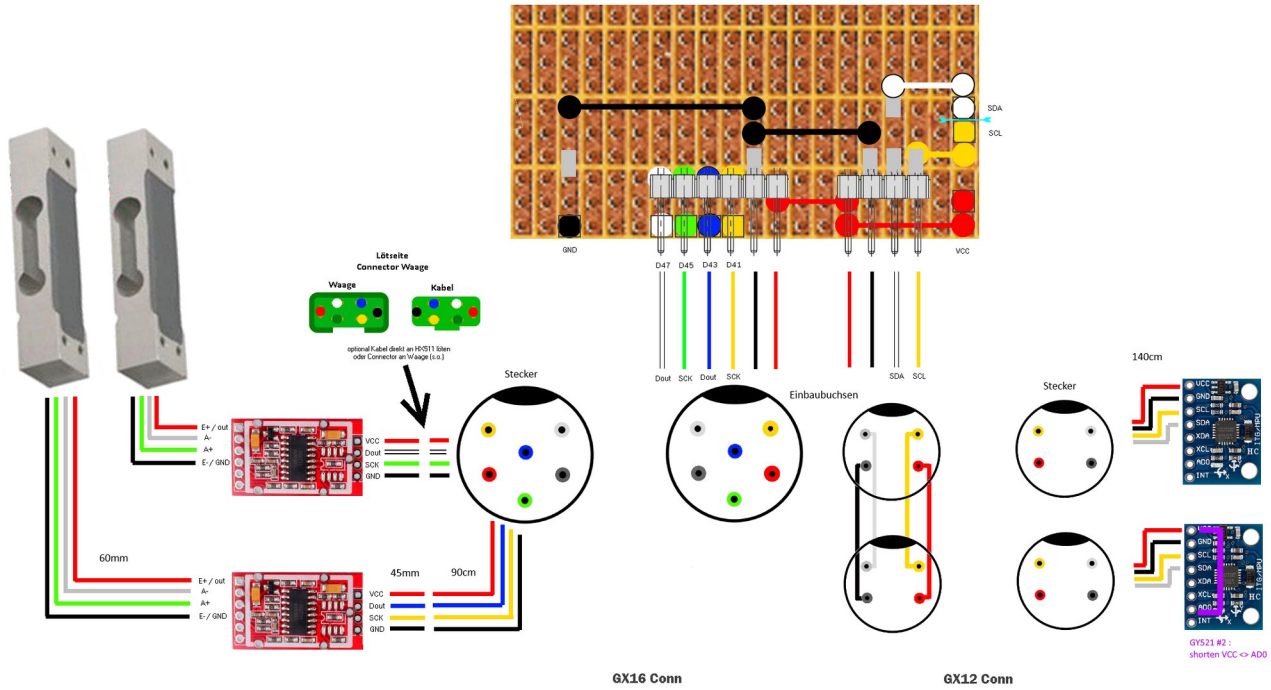
Das Kabel sollte besonders leicht & flexibel sein, um bei Ruderbewegungen bzw. EWD Messungen keine Verfälschungen der Meßwerte zu verursachen.
Besonders empfehlenswert sind z.B. Tonabnehmerkabel.

Anhang

Komponentenübersicht



Verdrahtungsschema Connector Shield



!! jeweils zweiten GY521 Sensor (IMU) wg Adressierung zwischen VCC und AD0 brücken !!

Teileliste „3in1“ (Vorschlag)

Head Unit	Arduino Mega 2560	CPU Board	1 Stck	Roboter Shop	https://www.roboter-bausatz.de/2/mega-25
	2,8“ Touch	Display	1 Stck	banggood	https://www.banggood.com/de/Geekcreit-2_
	9V Clip	P-Supply	1 Stck	ebay	https://www.ebay.de/itm/9V-Batterieclip-Bat
	Wippschalter sw	P-Supply	1 Stck	ebay	https://www.ebay.de/itm/10Pcs-16mm-klein
	Hohlwinkelstecker	2.1 x 5.5mm	1 Stck	ebay	https://www.ebay.de/itm/DC-Hohlstecker-2-
	Lochrasterplatine	ConnectorBoard	1 Stck	Roboter Shop	https://www.roboter-bausatz.de/1070/lochra
	Pfostenleiste 4 Pin	Gewinkelt, 2.54mm, Anschluss GY521 Conns	1 Stck	Roboter Shop	https://www.roboter-bausatz.de/629/10x-40
	Pfostenleiste 6 Pin	Gewinkelt, 2.54mm, Anschluss HX711 Conns	1 Stck	Roboter Shop	s.o.
	Connectorkabel 4Pol	Innenverkabelung Dupont Stecker	1 Stck	Roboter Shop	https://www.roboter-bausatz.de/510/4-pin-c
	Connectorkabel 6Pol	Innenverkabelung Dupont Stecker	1 Stck	Roboter Shop	https://www.roboter-bausatz.de/2673/6-pin-
	GX12 4Pol	Anschlüsse extern	1 Stck	ebay	https://www.ebay.de/itm/Stecker-GX12-12m
	GX16 8Pol	Anschlüsse extern	1 Stck	ebay	https://www.ebay.de/itm/GX16-Aviation-Kab
	Gehäuse	Pexiglas oder 3d Plot	1 Stck	ebay	https://www.ebay.de/itm/Gehause-fur-Ardui
	Gummifüsse	10mm	4 Stck	Klebeshop24	https://www.klebeshop24.de/Elastikpuffer-G
Ruderweg Messung	GY521	Sensoren	2 Stck	Roboter Shop	https://www.roboter-bausatz.de/9/3-achsen
	Kabel	Verbindung (Tonarmlitze)	3,5 m	ebay	https://www.ebay.de/itm/4-x-0-05-mm-Tona
	Halterung	Eigenbau / DIY 3d Plot	2 Stck		
EWD Messung	GY521	Sensoren	2 Stck	Roboter Shop	https://www.roboter-bausatz.de/9/3-achsen
	Kabel	Verbindung	3 m	ebay	https://www.ebay.de/itm/4-x-0-05-mm-Tona
	Halterung	Eigenbau Alu Schiene & 3d Plot	1 Stck		
CG Waage	HX711+Waegezelle	Sensoren, Meßbereich je nach Bedarf	2 Stck	ebay	https://www.roboter-bausatz.de/204/hx711-
	Connectorkabel 4Pol	Innenverkabelung, z.B. DuPont	2 Stck	Roboter Shop	
	Kabel	Verbindung, 6 adrig, z.B. LIYCY 6x0,14	1 m		
	GX16, 8pol	optional, wenn Kabel abnehmbar	1 Stck	ebay	https://www.ebay.de/itm/GX16-Aviation-Kab
	Gehäuse	DIY 3d Plot	1 Stck		

Anmerkungen

Allgemein

Ein DIY Projekt motiviert schnell dazu die eigenen Ideen an vorhandene Vorschläge anzupassen oder Verbesserungen vorzunehmen.
Man sollte diese Veränderungen imho in Abhängigkeit der eigenen Kompetenz umsetzen, sonst veriirt man sich schnell in Themengebiete, die einem nicht liegen und das „Projekt“ verliert schnell seinen Reiz.

Daher ein paar Anmerkungen zu ein paar Themengebiete, die einem das Leben ggf. erleichtern können.

Touchdisplay

Es gibt eine Vielzahl unterschiedlichster Touch Displays auf dem Markt.
Die Konfiguration der Ansteuerung und Treibervielfalt ist dementsprechend.
Ich kann nur empfehlen die beiden von mir getesteten Panels zu verwenden, wenn man noch nicht viel Übung im Umgang mit Arduinos besitzt.
Weitere Displays supporte ich zunächst nicht.

Wer andere Panels verwenden möchte sollten zumindestens darauf achten, dass sie über „Adafruit_GFX.h“ und „TouchScreen.h“ ansteuerbar sind.

Die Beispielprogramme der Hersteller zeigen meist, wie die Panels zu konfigurieren sind.
Diese Konfig wird in der Datei „lcd_config“ übernommen
Üblicherweise müssen diese Parameter angepasst werden:

```
36  // Touchscreen pinning
37  // please evaluate by looking into sample code from supplier
38
39  // 2.8 lcd; set together with uno & 2.4" display
40  // touch panel
41  #define YP A2  // must be an analog pin, use "An" notation!
42  #define XM A1  // must be an analog pin, use "An" notation!
43  #define YM 6   // can be a digital pin
44  #define XP 7   // can be a digital pin
45  //SPI Communication
46  #define LCD_CS A3
47  #define LCD_CD A2
48  #define LCD_WR A1
49  #define LCD_RD A0
50
```

Sollte ein Touch Display neu kalibriert werden müssen (Unterlagen & Software des Anbieters zu Rate ziehen !“), so sind die neuen Kalibrierdaten zu hinterlegen:

```
67  // calibration mins and max for raw data when touching edges of screen
68  #define TS_MAXX 910
69  #define TS_MINX 141
70
71  #define TS_MAXY 906
72  #define TS_MINY 130
73
74
```

Das Touch/Tipp-Verhalten kann bei Bedarf an in der Datei „touch.h“ angepasst werden

```
// Pressure Defs
#define MINPRESSURE 3
#define MAXPRESSURE 2000

#define TOUCHPRESSURE 20
```

GY 521 / MPU6050

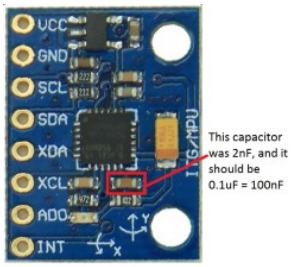
Der Gy521 ist ein IMZ Sensor mit hoher Auflösung un sehr gutem Preis/Leistungsverhältnis

In der Vergangenheit waren “günstige China Boards“ leider häufig falsch bestückt, was ein deutliches Rauschen zur Folge hat.
Ob man damit leben kann, ist eine individuelle Entscheidung.

Erkennbar wird der Effekt an den unruhigen Werten bei Winkelmessungen.
Alternativ berechnet man die Varianz/Streuung einer Meßreihe.
Der Rohdaten Meßbereich liegt zwischen +/-16384, typische Varianzwerte bei ca 55.
Ich persönliche tausche bei Varianzen > 70 den u.a. Kondensator aus:

Um das Rauschen zu minimieren wird einen SMD Kondensator ausgelötet und gegen einen mit 100nF Kapazität getauscht.
Dazu sind entsprechende Lötkenntnisse Voraussetzung

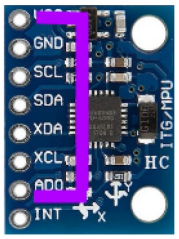
https://forum.arduino.cc/index.php?topic=394691.0



Es kommen für EWD und meistens für die Rudermessung immer zwei Sensoren gleichzeitig zum Einsatz.

Der zweite Sensor (Leitwerkssensor , zweites Ruder) muss über den Arduino über eine zweite Adresse angesprochen werden können

Daher müssen alle zweiten Sensoren zwingend mit einer Brücke zwischen VCC und AD0 ausgestattet werden



GY521 #2 :
shorten VCC <> AD0

3D plots

Die ideale Lösung ist der 3 Plot für die Headunit (STL Dateien im Ordner \plots\headunit , der Schwerpunkt bzw EWD Waage sowie den Klammern für die Rudermessung.

Nicht jeder besitzt einen 3d Drucker, oder hat Freunde die dies übernehmen können.

Alternativen:

- Headunit:
hier kann als einfache Lösung ein Plexiglas/Acryl Gehäuse für den Arduino Mega verwendet werden, auf dem die shield einfach „offen“ aufgesteckt werden
 (s.a. eckstein-shop, ebay …)

- EWD Waage
falls vorhanden, können vorhandene Waagen / Messchieber mit dem GY 521 Sensor ausgestattet werden

- Rudermessung
hier sind durchaus Möglichkeiten durch Eigenbau ohne 3D drucker gegeben,
s.a. https://www.rc-network.de/threads/ruderweg-messung-mit-arduino.639364/page-33#post-7298236

- Schwerpunktwaage
„Ausserhalb“ der 3d Plot varianten sind mir nur Varianten für den „schweren“ Einsatz, erstellt aus Aluprofilen bekannt.