

W132

Inhalt

W132	1
1) Vorwort	3
2) Umbau des W132	4
2.1. Stückliste	4
2.2. Öffnen des Gehäuses und Zerlegen des Sensors	5
2.3. Umbau der Blende.....	5
2.4. Spannstift ablängen.....	6
2.5. Spannstift einsetzen	6
2.6. Magnethalterung anbringen	7
2.7. Magnet fixieren und Einheit einkleben	7
2.8. Auflage für den Hallsensor vorbereiten	7
2.9. Kabel anlöten.....	8
2.10. Platine einsetzen	9
2.11. Kabel durchführen und AS5600-Einheit einsetzen.....	9
2.12. Windfahnen-Einheit einsetzen	9
2.13. Sensormodul wieder verschließen	10
2.14. Platine verbinden	10
2.15. NMEA 2000.....	10
2.16. Skizzen der Verkabelung	11
3) Software	12
3.1. Vorüberlegung.....	12
3.2. Vorbereitung des ESP32	12
3.3. Herunterladen des Flash Tools.....	12
3.4. Herunterladen der aktuellen Firmware.....	12
3.5. Aufspielen der aktuellen Firmware	13
4) Nutzung der Arduino-IDE	14
4.1. Den Boardmanager für den ESP32 hinzufügen	14
4.2. Die ESP32-Boards über den Boardmanager hinzufügen	14
4.3. Korrektes Board wählen	15
4.4. Die benötigten Bibliotheken herunterladen	15
4.5. Das GitHub-Repo herunterladen.....	15
4.6. Die benötigten Bibliotheken hinzufügen.....	16
4.7. Vorbereitungen zum Flashen der Firmware.....	16

5) Einrichtung	17
5.1. TCP	17
5.2. Standalone	17
6) Grafische Oberfläche	18
6.1. Anzeige der Winddaten	18
6.2. Einstellungsseite	18
6.2.1. Offset	19
6.2.2) Korrekturfaktor	19
6.2.3) Einheit der Windgeschwindigkeit	19
6.2.4) Blinken der IP	19
6.2.5) Netzwerk- oder Standalone-Modus	20
6.2.6) NMEA 2000	20
6.2.7) Sprache/Language	20

1) Vorwort

Bei diesem Projekt handelt es sich primär um eine Machbarkeitsstudie. Zielstellung war es, bei diesem Umbau mit so wenig neuen Teilen wie möglich auszukommen. Somit braucht es, neben der neuen Platine und der elektronischen Bauelemente dafür, die an dieser Stelle jedoch nicht weiter betrachtet werden sollen, lediglich ein weiteres Teil: Einen 5mm-Spannstift. Die restlichen Teile werden aus dem ursprünglichen Windsensor übernommen.

Zweifelsohne lassen sich elegantere und ausgefeiltere Umbaumöglichkeiten finden, etwa indem der Magnethalter ein dediziertes, 3D-Gedrucktes Bauteil bekommt. Dies ist jedoch nicht der Anspruch dieser Anleitung, auch wenn ich diese Teile gegebenenfalls noch an geeigneter Stelle nachreichen werde.

Das verhältnismäßig große Gehäuse des W132 macht es möglich, einen ESP32 als Chip zu verwenden, was gleich zwei Vorteile bringt:

- 1) Das Löten einer dedizierten Platine entfällt
- 2) NMEA 2000-Daten können direkt im Sensor erstellt und (zusätzlich zu NMEA 1803) ausgegeben werden

Dieser Funktionsumfang, zusammen mit einem Umbau der kein Spezialwerkzeug und zwar einen LötKolben, aber keine Löterfahrung, erfordert, macht dieses Projekt in meinen Augen so interessant.

2) Umbau des W132

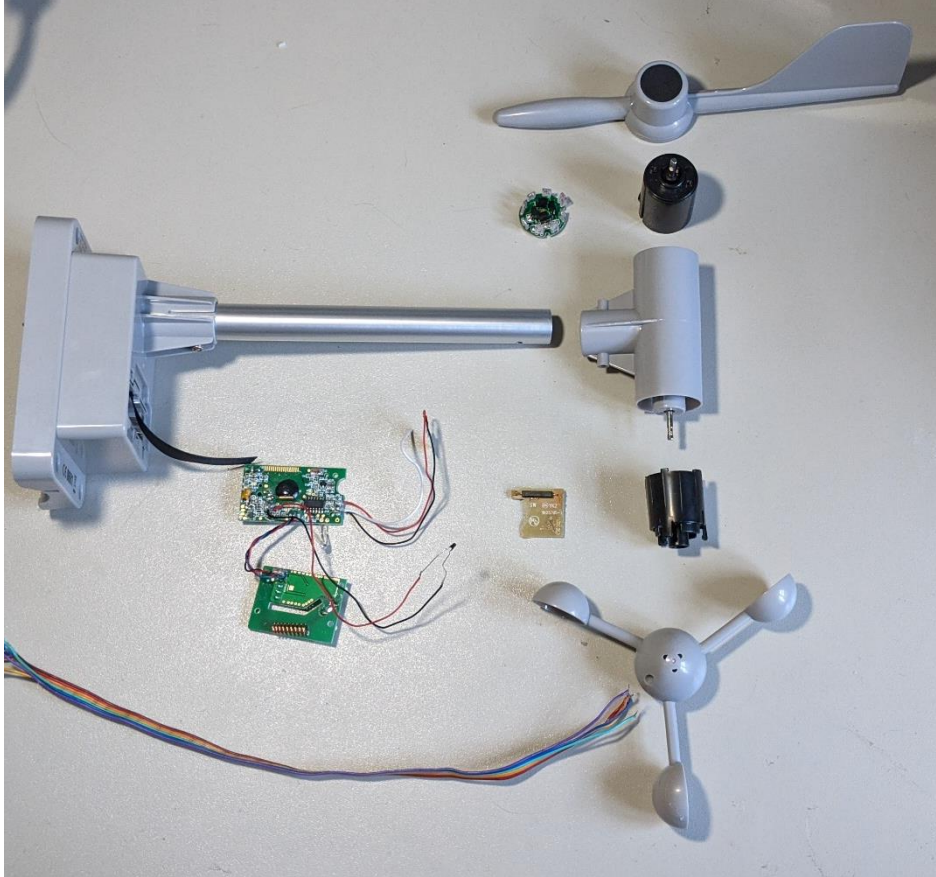
Der erste Teil der Dokumentation wird sich mit dem Umbau des W132 befassen. Wie in der Einleitung erwähnt hat der Umbau den Zweck, möglichst wenig zusätzliche Teile zu benötigen, und dabei gleichzeitig möglichst einfach nachzubauen und robust zu sein.

2.1. Stückliste

ID	Menge	Bezeichnung	Material	Bemerkung
Mechanikteile				
1	1	Spannstift 5mm	Federstahl	auf einer Seite angespitzt
Elektronik				
2	1	ESP32		Das Pinout der Entwicklerbords unterscheidet sich
3	1	AS5600-Platine		Sensor auf Breakout-Platine
4	1	DC/DC-Wandler		von benötigter Eingangsspannung auf 5v
NMEA 2000				
5	1	SN65HVD230 CAN BUS Modul		
6	4	Jumperkabel		
Software				
7	1	Firmware		https://github.com/jukolein/W132/tree/main/Firmware
8	1	ESP32-Flasher		https://github.com/doayee/esptool-esp32-gui

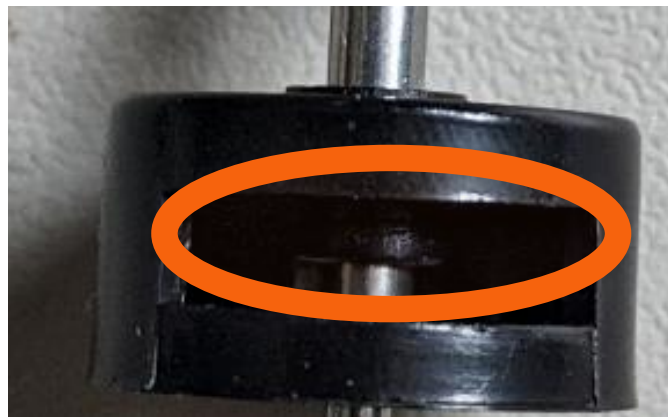
2.2. Öffnen des Gehäuses und Zerlegen des Sensors

Hierfür sind die Verschraubungen auf der Rückseite der Basiseinheit sowie an dem Alurohr, der Windfahne und dem Windrad zu lösen. Das 7-adrige Kabel an beiden Enden ablöten, es wird später wieder verwendet.



2.3. Umbau der Blende

Um möglichst viele der Teile wiederverwenden zu können wird die Blende zur neuen Aufhängung umgebaut. Dafür wird diese zuerst vom Metallstift abgezogen und anschließend der nach innen zeigende Teil der Führung mit einer Zange abgedreht.



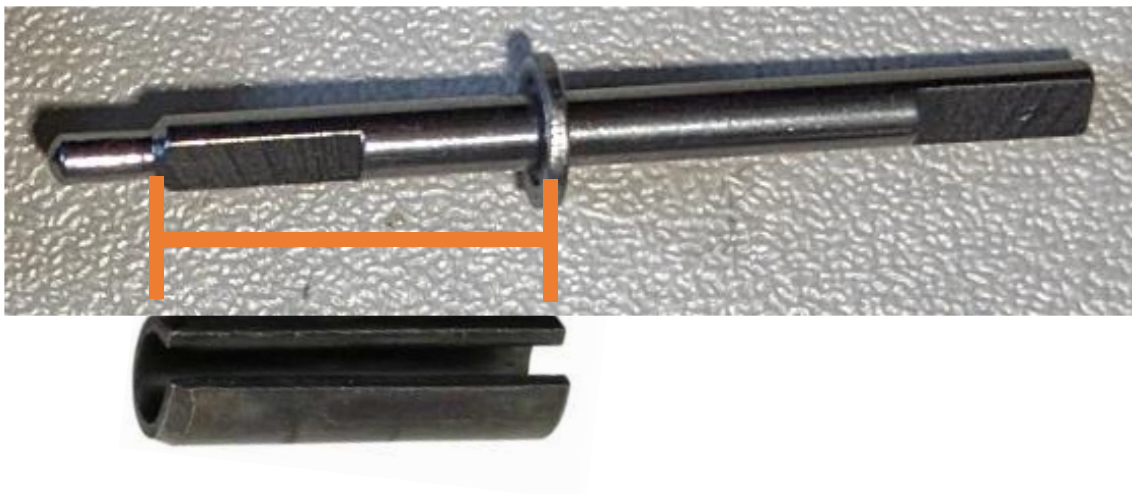
Anschließend wird die Öffnung mit einem 5mm-Bohrer aufgebohrt.



2.4. Spannstift ablängen

Aufgrund der mechanischen Belastung habe ich mich an dieser Stelle für ein Metallteil entschieden. Ein Spannstift schien mir hierbei am geeignetsten, da dieser bereits auf einer Seite angespitzt ist, was die Reibung vermindert, sehr gut zum Durchmesser des Metallstiftes passt und zudem problemlos in jedem Baumarkt erhältlich ist.

Der Spannstift wird zuerst auf die benötigte Länge zurechtgesägt, wobei von der angespitzten Seite aus gemessen wird.



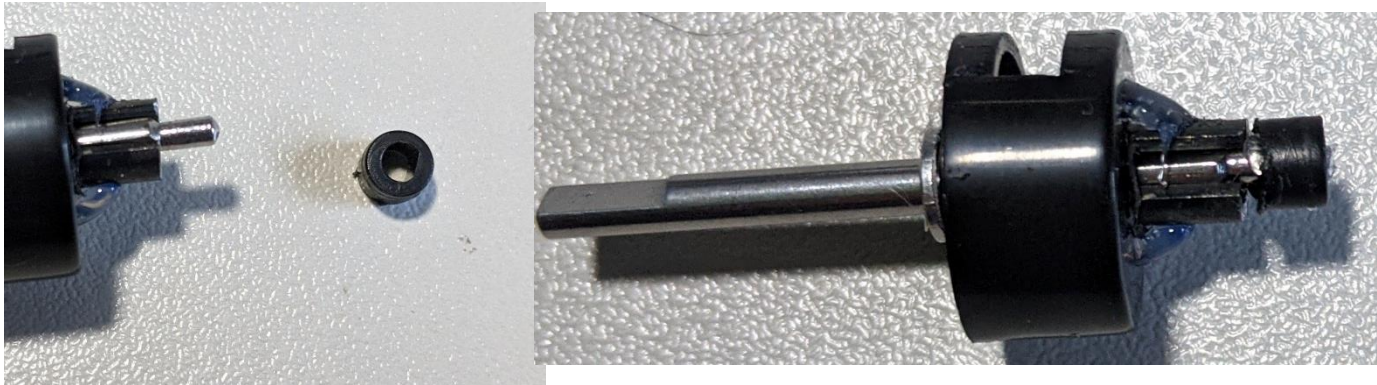
2.5. Spannstift einsetzen

Der Spannstift wird mit der angespitzten Seite nach innen durch die Bohrung gesteckt bis er bündig mit der Unterseite der Blende abschließt und anschließend mit Kleber fixiert.



2.6. Magnethalterung anbringen

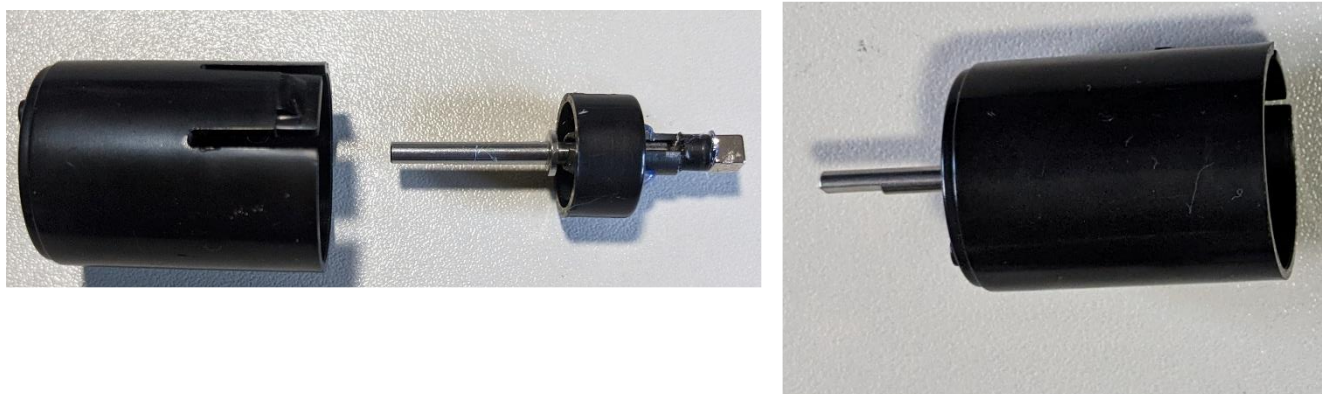
Um dem Magneten eine größere Auflagefläche zu bieten, wird nun, nachdem der Metallstift mit dem verjüngten Ende zuerst durch den Spannstift gesteckt wurde, das zuvor in Schritt 2) abgetrennte Führungsstück auf die Spitze des Metallstifts geklebt. Es ist darauf zu achten, dass sich der Metallstift danach noch frei und leichtgängig drehen lässt.



Alternativ hierzu bietet sich das „Einschmelzen“ eines Plastikquaders an, wobei das Ende des Metallstifts erhitzt und anschließend in den quaderförmigen Plastikblock gedrückt wird. Dies führt zu einer stabileren Verbindung, setzt jedoch ein weiteres, wenn auch einfach zu beschaffendes, Teil voraus.

2.7. Magnet fixieren und Einheit einkleben

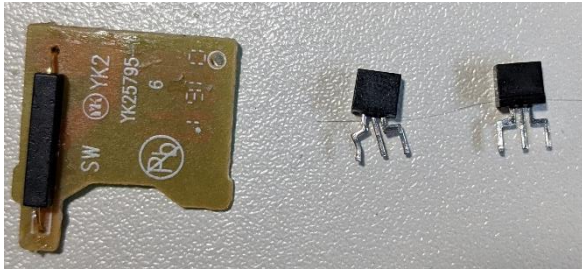
Nachdem der Quadermagnet auf der vergrößerten Auflagefläche fixiert wurde, kann nun die gesamte Einheit wie abgebildet eingeklebt werden. Auch hier ist darauf zu achten, dass sich der Metallstift möglichst leichtgängig und ohne viel Spiel bewegen lässt.



2.8. Auflage für den Hallsensor vorbereiten

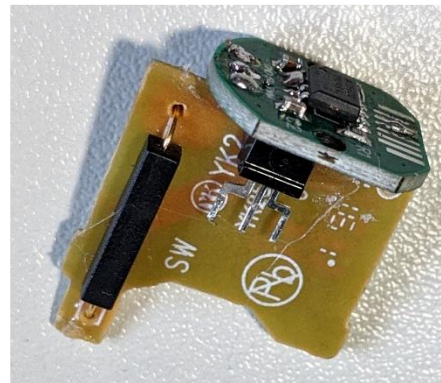
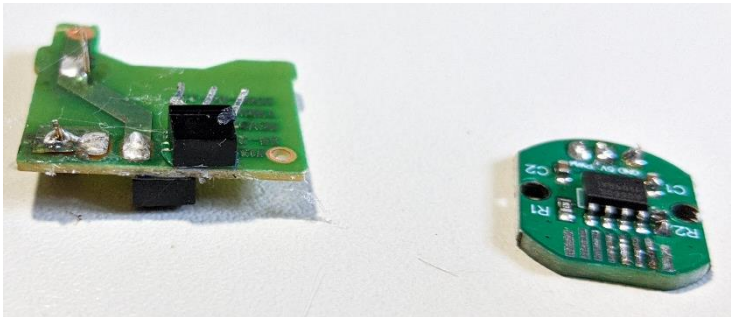
Wie in der Einleitung erwähnt handelt es sich hierbei hauptsächlich um einen „proof of concept“, dass ein Umbau des Windsensors mit nur einem einzigen zusätzlichen Teil möglich ist. Daher werden auch im Folgenden Teile des Windsensors wiederverwendet, obgleich sich mithilfe eines 3D-Druckers auch passgenauere Teile fertigen ließen.

Für die Auflage des AS5600 werden die Platine mit dem Reed-Switch sowie zwei der schwarzen Photodioden benötigt. Letztere werden einfach aus der ursprünglichen Platine ausgelötet.



Diese werden dann leicht versetzt oben und unten an die Platine geklebt, um eine stabile Auflagefläche für den AS5600 zu schaffen.

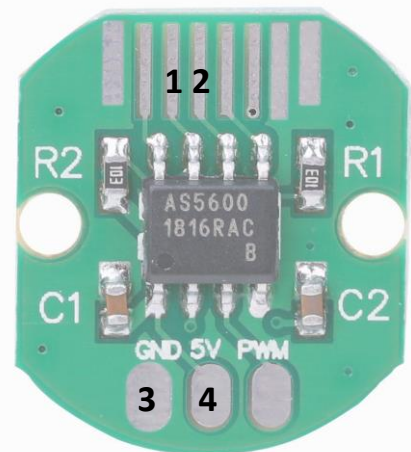
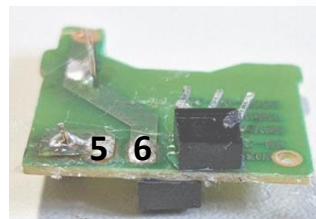
Der AS5600 wird anschließend so auf die Platine und die Photodioden geklebt, dass sich der schwarze Sensor mittig befindet.



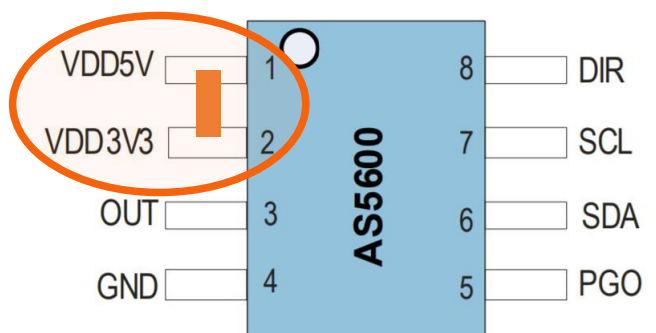
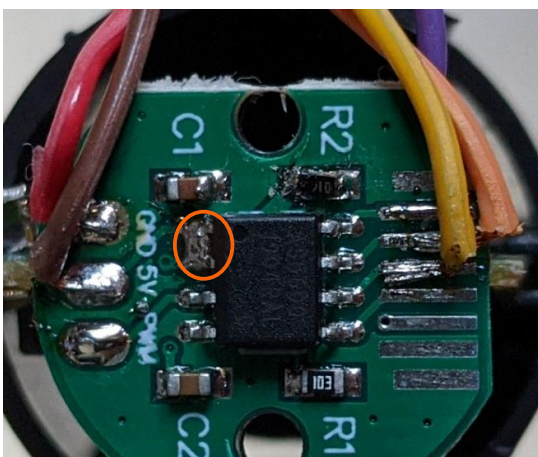
2.9. Kabel anlöten

Anschließend wird das 7-Adrige Kabel mit dem Sensor wie folgt verbunden:

- Orange --> 1
- Gelb --> 2
- Rot --> 3
- Braun --> 4
- Blau und Grün mit den beiden Kontakten des Reedswitches (5 und 6)

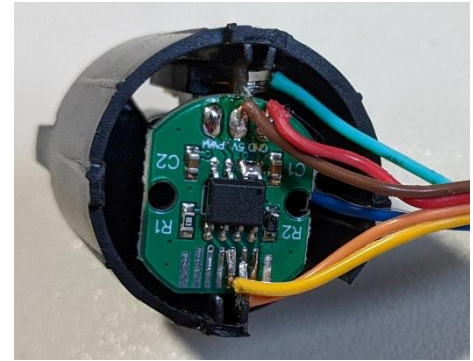
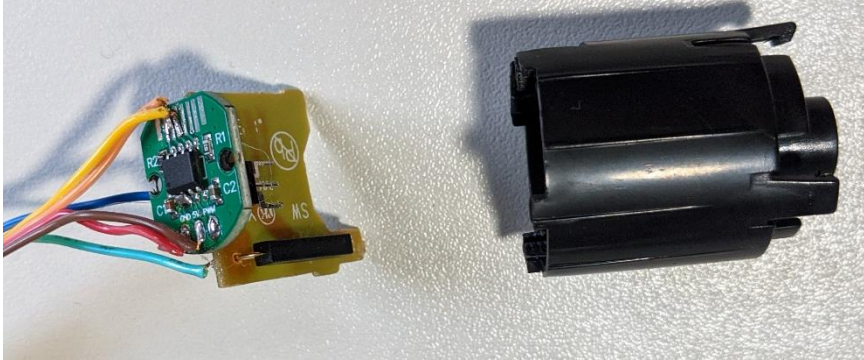


Anschließend wird mit Lötzinn eine Brücke zwischen VDD5V und VDD3V3 hergestellt, da der Chip mit 3V gespeist wird.



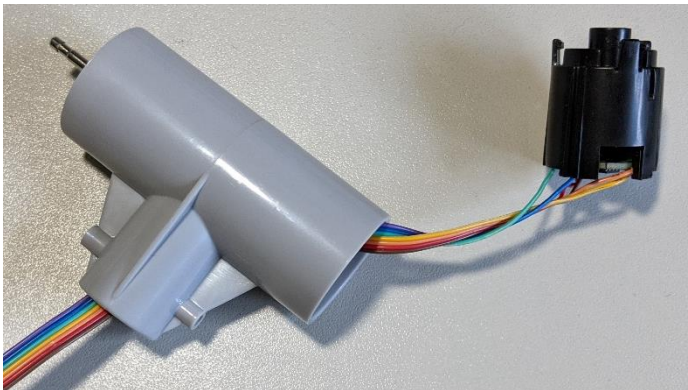
2.10. Platine einsetzen

Nachdem die Adern angelötet wurden kann die Einheit mit den zwei Platinen wieder in die alte Fassung eingesetzt werden.



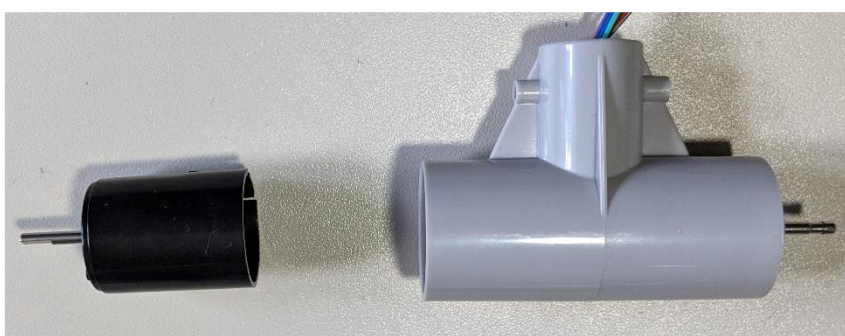
2.11. Kabel durchführen und AS5600-Einheit einsetzen

Das Kabel wird nun durch das T-Stück des Gehäuses geführt und das Modul mit der AS5600-Platine wieder eingesetzt. Dabei ist auf die Führung durch eine Nut zu achten. Auch sollte das Kabel absolut flach liegen, da es sonst zu Quetschungen der Adern kommen kann.



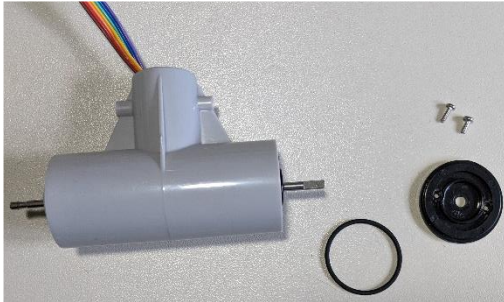
2.12. Windfahnen-Einheit einsetzen

Nun kann die in Schritt 8) fertiggestellte Einheit auch wieder eingesetzt werden. Diese muss hörbar in das graue Gehäuse einrasten.

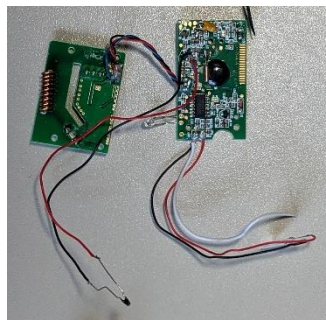


2.13. Sensormodul wieder verschließen

Abschließend wird der Dichtungsring wieder eingesetzt und die Deckplatte verschraubt.



Anschließend kann der Sensor wieder gänzlich zusammengebaut werden. Übrig bleiben sollten dabei nun lediglich die Platine mit den LEDs und Photodioden sowie das Funkmodul und die ursprüngliche Logikplatine:



2.14. Platine verbinden

Nachdem das 7-Adrige Kabel durch das Alurohr zurück in die Basis geführt wurde werden die Adern mit dem ESP32 wie folgt verbunden:

- Braun --> 3v3
- Rot --> Ground
- Gelb --> D21
- Orange --> D22
- Blau --> Ground
- Grün --> D18

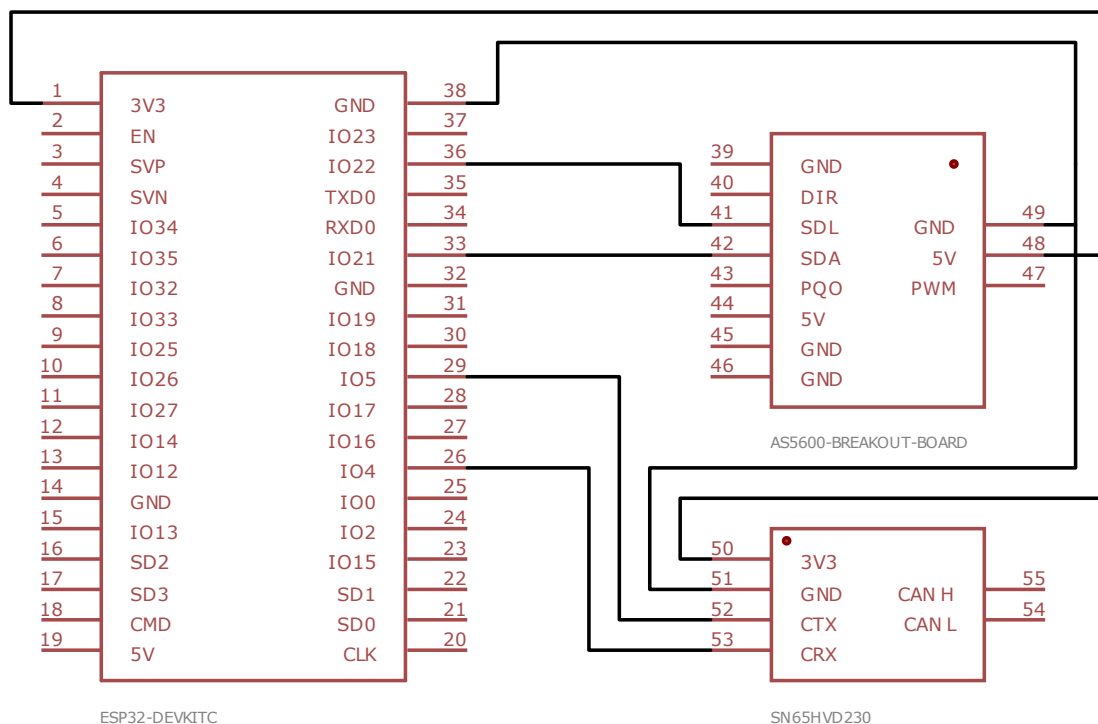
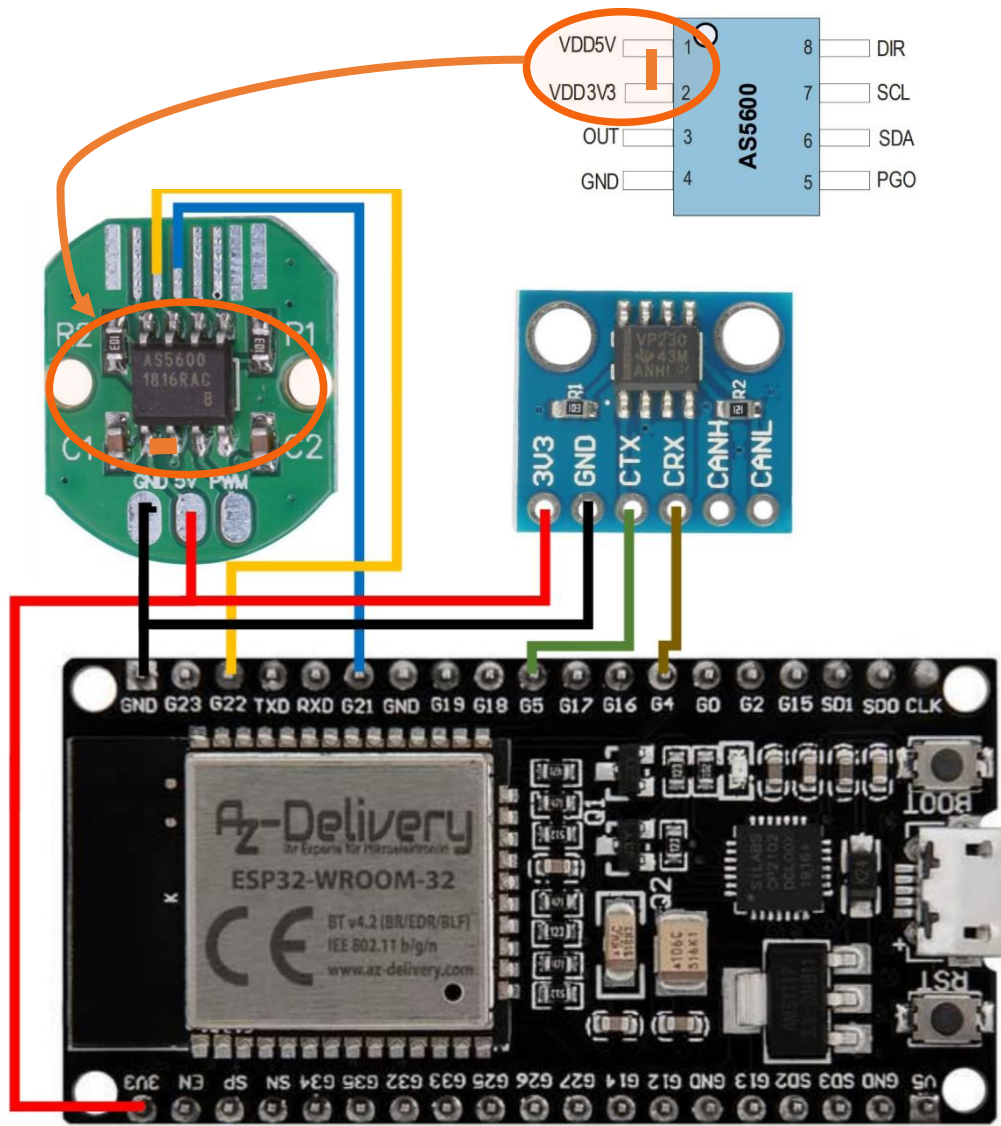
Zwischen den verschiedenen Entwicklerplatinen gibt es Unterschiede bezüglich des Layouts der Pins, daher sollte immer genau auf die Beschriftung auf dem Bord geachtet werden. Das exemplarische Layout für einen AZ-Delivery -Board ist auf der nachfolgenden Seite gezeigt. Bei der Verwendung anderer Boards muss das gegebenenfalls abweichende Layout der Pins berücksichtigt werden.

2.15. NMEA 2000

Ist zudem die Ausgabe von NMEA 2000 gewünscht, muss noch ein CAN BUS Modul angeschlossen werden. Hierbei hat sich das SN65HVD230 bewährt. Die Verbindungen sind:

- 3v3 --> 3v3
- Ground--> Ground
- CAN TX--> D5
- CAN RX--> D4

2.16. Skizzen der Verkabelung



3) Software

3.1. Vorüberlegung

Da es sich bei dem Chip um einen ESP32 handelt, der quasi ausschließlich auf Entwicklerplatinen verkauft wird, ist die Verbindung mit dem Rechner zwar einfacher als mit dem ESP8266, jedoch wird das Aufspielen der Software ohne IDE durch die Partitionen des Speichers erschwert.

Grundsätzlich gibt es für das Aufspielen der Firmware zwei Möglichkeiten:

- 1) Den Programmcode über die Arduino-IDE aufspielen
- 2) Die kompilierten Binärdatei direkt aufspielen

Welcher Weg gewählt wird hängt hauptsächlich davon ab, ob noch Änderungen am Programmcode vorgenommen werden sollen. Ist dies der Fall, so ist die Verwendung der Arduino-IDE unumgänglich. Die Vorbereitung der IDE wird im folgenden Kapitel näher betrachtet. Ist die IDE erst fertig eingerichtet, so kann das Aufspielen ganz einfach über den entsprechenden „Hochladen“-Knopf erfolgen. Hier ist jedoch auf die richtigen Einstellungen des Boards zu achten, welche ebenfalls im nächsten Kapitel näher betrachtet werden.

Sind keine Änderungen am Code gewünscht, so kann man auch die vorbereiteten Dateien direkt aufspielen. Dies geht schneller und es muss auf weniger Einstellungen geachtet werden.

Beide Möglichkeiten sind für alle drei gängigen Betriebssysteme (Windows, Mac, Linux) verfügbar.

3.2. Vorbereitung des ESP32

Da sich der ESP32 bereits auf einer Entwicklerplatine befindet, muss in diesem Schritt lediglich mittels USB eine Verbindung zum Rechner hergestellt (und, abhängig davon welches Board gekauft wurde, eventuell noch die Chiptreiber installiert) werden. Sobald das Bord vom Rechner erkannt wurde, kann es weiter gehen.

3.3. Herunterladen des Flash Tools

Zwar bietet ESPRESSIF, der Hersteller des ESP32, auch eine eigene Software an (<https://www.espressif.com/en/support/download/other-tools>), für unsere Zwecke gibt es jedoch noch ein intuitiveres Programm: <https://github.com/doayee/esptool-esp32-gui>. Zudem ist dieses, anders als das originale Flash Tool, auch unter Mac lauffähig.

Das Programm kann direkt aus dem originalen GitHub-Repo heruntergeladen werden, befindet sich aber auch noch mal in meinem Repo, zusammen mit den BIN-Dateien.

Die Verwendung des DoayeeESP32DFU soll nachfolgend näher beschrieben werden.

3.4. Herunterladen der aktuellen Firmware

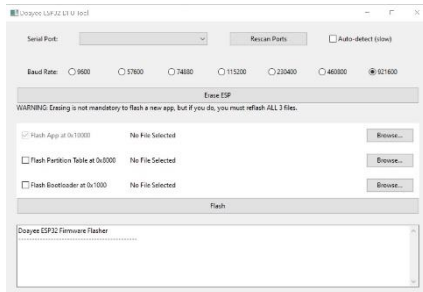
Die aktuelle Firmware kann, ebenfalls über GitHub, über folgenden Link bezogen werden: <https://github.com/jukolein/W132/tree/main/Firmware>.

Die herunterzuladende Datei muss auf „.esp32.bin“ enden.

3.5. Aufspielen der aktuellen Firmware

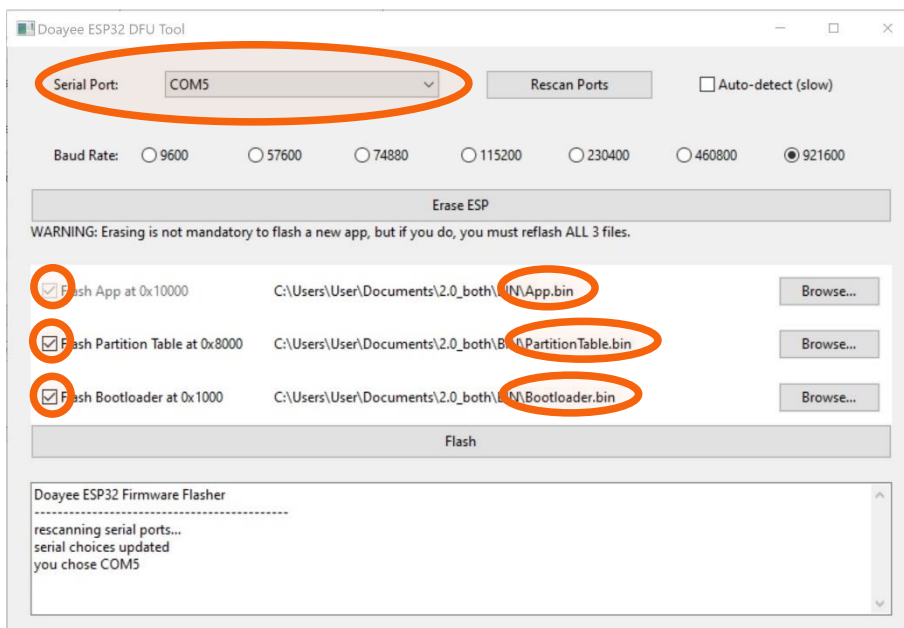
Dieser Schritt ist, wie bereits angesprochen, beim ESP32 etwas komplexer, wird jedoch durch die grafische Benutzeroberfläche des Programms stark vereinfacht.

Nach dem Starten von DoayeeESP32DFU.exe sollte folgendes Fenster erscheinen:

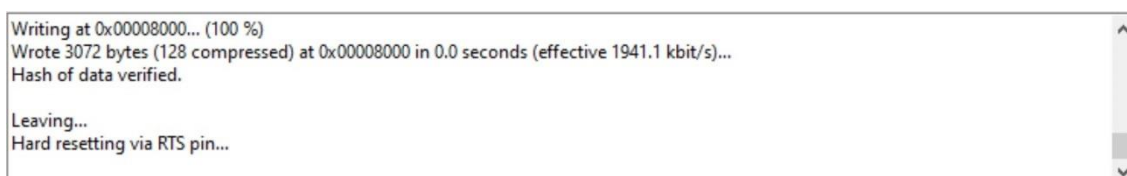


Hierbei müssen nun der korrekte Serielle Port und alle drei Dateien ausgewählt und die Haken davor gesetzt werden.

Die Benennung der BIN-Dateien entspricht dabei den jeweiligen Feldern.



Anschließend auf „Flash“ drücken. Wenn der Vorgang erfolgreich war, sollte die Konsole in etwa so aussehen:



Um sicher zu gehen, kann nun die Arduino IDE geöffnet werden. Darin nun den richtigen Port einstellen und den Seriellen Monitor aufrufen, beim ESP32 auf „RST“ klicken und überprüfen, ob das Programm korrekt gestartet wird.

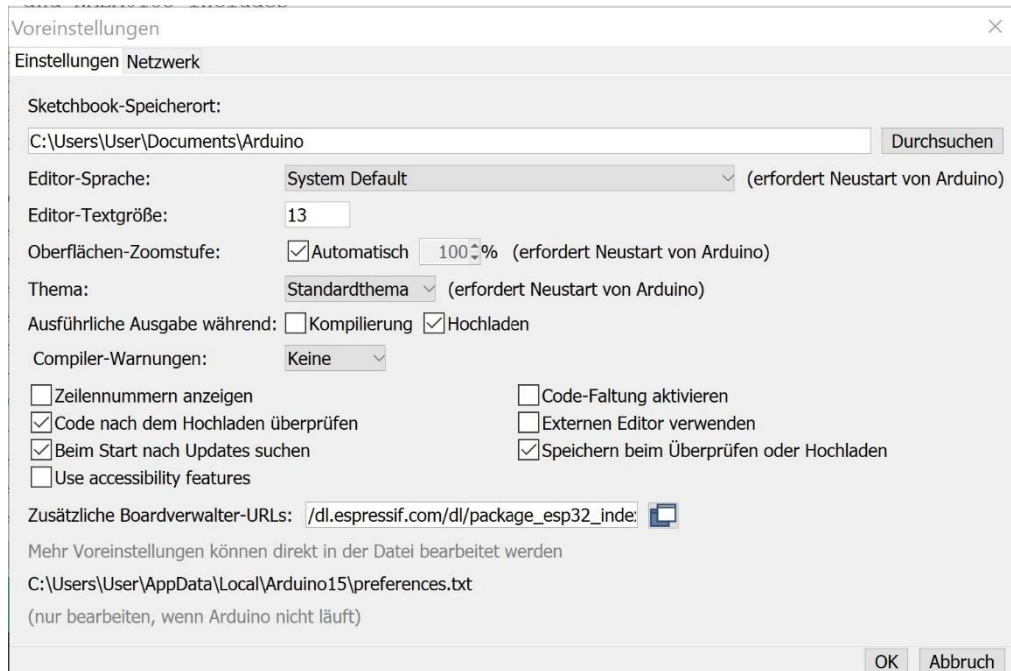
Nähere Informationen zur Verwendung der Arduino IDE folgen im nächsten Kapitel.

4) Nutzung der Arduino-IDE

4.1. Den Boardmanager für den ESP32 hinzufügen

Hierfür muss unter „Datei --> Voreinstellungen --> zusätzliche Boardverwalter-URLs“ folgende URL eingegeben werden: https://dl.espressif.com/dl/package_esp32_index.json

Anschließend mit „OK“ bestätigen.



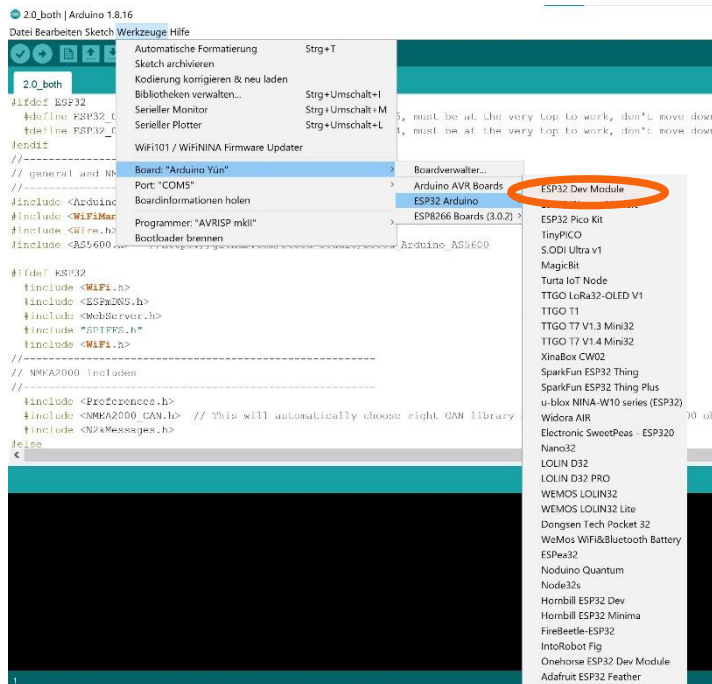
4.2. Die ESP32-Boards über den Boardmanager hinzufügen

Öffnet man nun unter „Werkzeuge --> Board --> Boardverwalter“ den Boardverwalter und gibt oben in die Suchleiste „esp32“ ein, erscheint folgender Eintrag:



4.3. Korrektes Board wählen

Unter „Werkzeuge --> Board --> ESP32 Boards“ nun „ESP32 Dev Module“ auswählen.



4.4. Die benötigten Bibliotheken herunterladen

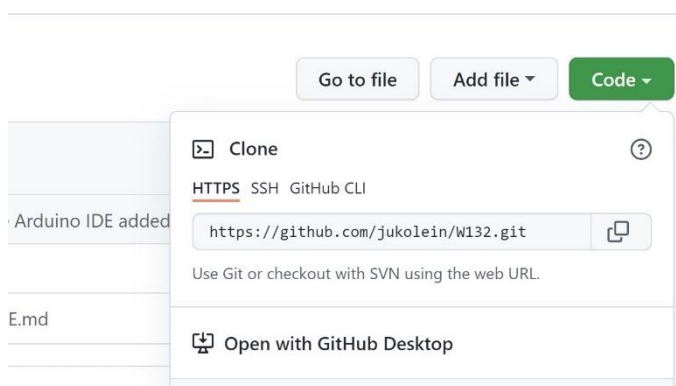
Es werden vier zusätzliche Bibliotheken verwendet: „WiFiManager“, „AS5600“, „NMEA2000“ und „NMEA2000_esp32“

Um diese hinzuzufügen, gibt es zwei Möglichkeiten:

- 1) Die Bibliotheken werden direkt über die entsprechenden GitHub-Repos bezogen. Die Links dafür lauten wie folgt:
 - WiFiManager: <https://github.com/tzapu/WiFiManager>
 - AS5600: https://github.com/SeeedStudio/Seeed_Arduino_AS5600
 - NMEA2000: <https://github.com/ttlappalainen/NMEA2000>
 - NMEA2000_esp32: https://github.com/ttlappalainen/NMEA2000_esp32
- 2) Es werden die im GitHub-Repo des Windsensors enthaltenen Bibliotheken verwendet. Dieser Weg wird im Folgenden gezeigt.

4.5. Das GitHub-Repo herunterladen

Die Dateien können bequem über den Browser heruntergeladen werden. Hierfür wird erst auf „Code“ geklickt und anschließend das Repo durch einen Klick auf „Download ZIP“ heruntergeladen. Die ZIP-Datei muss danach auf dem Rechner entpackt werden.

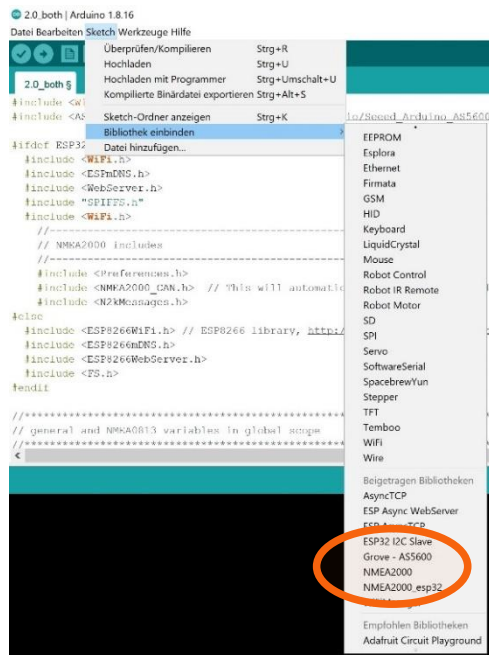


4.6. Die benötigten Bibliotheken hinzufügen

Über „Sketch --> Bibliothek einbinden --> „ZIP-Bibliothek hinzufügen“ können nun die benötigten Bibliotheken eingebunden werden.

Diese befinden sich in „Firmware --> Libraries“. Die darin enthaltenen .ZIP-Dateien müssen nicht entpackt werden, sondern werden als solche direkt in die Arduino-IDE importiert.

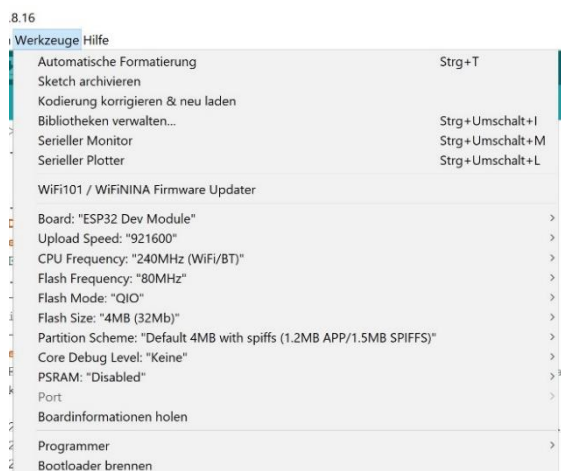
Anschließend sollten diese unter „Sketch --> Bibliotheken einbinden“ im Abschnitt „Beigetragene Bibliotheken“ aufgelistet sein.



Damit ist die Arduino IDE nun fertig vorbereitet und es können gegebenenfalls Änderungen am Code vorgenommen werden. Durch Lade der unveränderten Firmware in die Arduino-IDE und Klickens auf „Überprüfen“ (oben links) lässt sich feststellen, ob die Einrichtung erfolgreich war.

4.7. Vorbereitungen zum Flashen der Firmware

Um sicherzustellen, dass die Firmware erfolgreich auf den ESP32 überspielt werden kann und auf diesem auch lauffähig ist, müssen folgende Einstellungen unter „Werkzeuge“ vorgenommen werden:



Sind die Einstellungen entsprechend vorgenommen worden und wurde der ESP32 vom Rechner erkannt (siehe Abschnitt 4.2), kann die Firmware durch einen Klick auf „Hochladen“ aufgespielt werden.

5) Einrichtung

Beim ersten Start nach dem Aufspielen der Firmware erstellt der ESP32 einen Access Point mit dem Namen „Windsensor“.

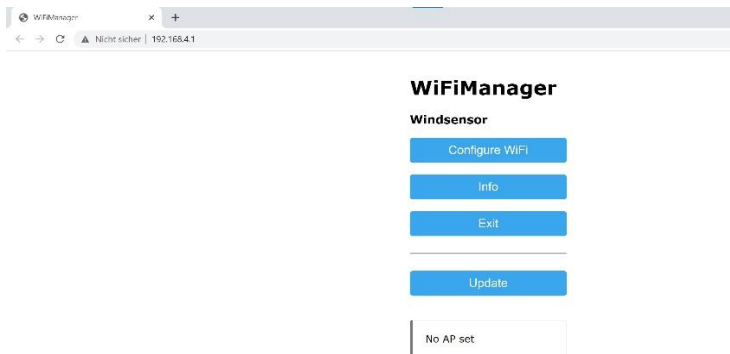


Nach erfolgreicher Verbindung mit diesem Netzwerk sollte man automatisch auf „192.168.4.1“ weitergeleitet werden. Ist dies nicht der Fall, so muss die IP manuell aufgerufen werden. Wie es danach weiter geht, hängt davon ab, welcher Nutzungsmodus (Integration in ein bestehendes Netzwerk oder Standalone) geplant ist. In beiden Fällen werden die **NMEA0813-Daten zusätzlich immer seriell ausgegeben**.

5.1. TCP

Hierbei stellt der ESP32 einen TCP-Webserver bereit, von dem sich die Clients (etwa Signalk oder AVnav) die NMEA-Daten holen können.

Hierfür ist es in aller Regel erforderlich, den ESP8266 mit einem bereits bestehenden Netzwerk zu verbinden. Dies kann leicht über die grafische Oberfläche des WiFiManagers geschehen.



Durch einen Klick auf „Configure WiFi“ gelangt man zu einer Auflistung aller verfügbaren WLAN-Netzwerke, durch einen Klick auf das gewünschte Netzwerk wird dessen SSID automatisch in das dafür vorgesehene Feld eingetragen, so dass nur noch das Passwort ergänzt werden muss. Durch Klicken auf „Save“ werden die Änderungen übernommen, der ESP32 startet sich neu und versucht, sich mit dem angegebenen WLAN zu verbinden. Gelingt dies nicht, so startet sich der Access Point „Windsensor“ erneut und wartet auf Eingaben.

Anschließend muss der Client (etwa Signalk oder AVnav) darauf konfiguriert werden, die NMEA0813-Daten unter der IP des ES32 auf Port 8080 zu empfangen.

5.2. Standalone

Der Windsensor ist auch in der Lage ohne vorhandene Netzwerke zu arbeiten und baut in diesem Fall sein eigenes WLAN-Netz auf.

Um dies zu erreichen, klickt man auf der Konfigurationsseite direkt auf „Exit“. Nach kurzer Zeit sollte nun das neue WLAN-Netzwerk „Windsensor_AP“ sichtbar sein, mit dem man sich mit dem Passwort „123456789“ verbinden kann. Der Windsensor ist von allen mit diesem Netzwerk verbundenen Geräten aus unter „192.168.4.1“ erreichbar.

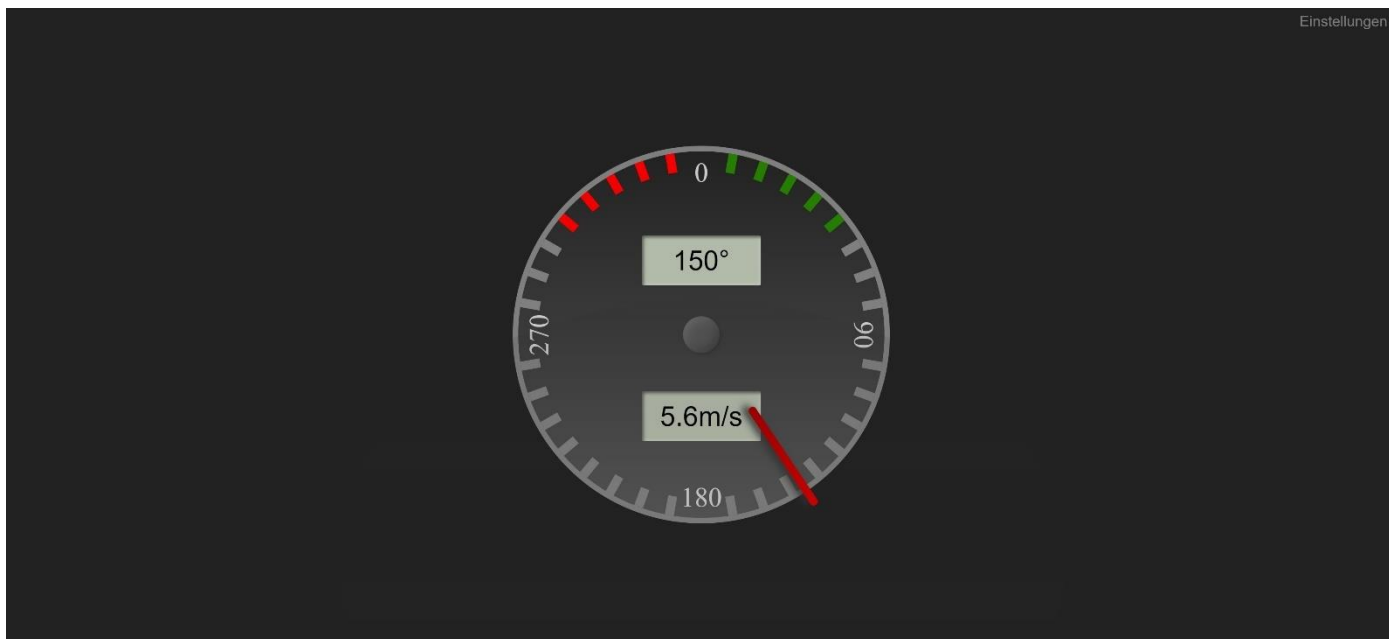
Um den Sensor dauerhaft im Standalone-Modus zu halten, muss dies noch in der Einstellungsseite aktiviert werden.

6) Grafische Oberfläche

Die grafische Oberfläche des Windsensors ist sowohl unter seiner IP-Adresse als auch unter „windsensor.local“ erreichbar. Jedoch funktioniert letzteres nur unter IOS zuverlässig und ohne weiteren Einrichtungsbedarf, unter Windows nur mittels spezieller „Bonjour“-Software, unter Linux nur nach Installation von mDNS-Paketen und unter Android gar nicht.

6.1. Anzeige der Winddaten

Der Windsensor verfügt neben einem TCP-Server auf Port 8080, über den die NMEA0813-Telegramme ausgesendet werden, auch noch einen http-Webserver auf Port 80. Dieser dient sowohl zur grafischen Anzeige der Winddaten als auch zur Verwaltung des Sensors.



Angezeigt wird der scheinbare Wind in Grad sowie die aktuelle Windgeschwindigkeit in m/s.

Vielen Dank an ziegenhagel.com <https://ziegenhagel.com/> für das Programmieren der Oberfläche.

Die Verarbeitung und Anzeige der Daten findet vollständig auf dem Client statt, so dass der ESP32 nicht belastet wird. Dieser stellt lediglich unter „/data“ den aktuellen NMEA0813-Satz zur Verfügung und aktualisiert ihn jedes Mal, wenn die Seite neu aufgerufen wird.



In der oberen rechten Ecke befindet sich ein Link, der auf die Einstellungsseite des Windsensors führt.

6.2. Einstellungsseite

Unter „/config“ besitzt der Windsensor eine Seite um Einstellungen an ihm vorzunehmen.

W132

Einstellungen

Vorzeichenbehafteter Offset der Windrichtung in Grad:

linearer Korrekturfaktor der Windgeschwindigkeit:

Aktuelle Einheit der Windgeschwindigkeit:

Aktuell wird die IP beim Start nicht geblinkt

Beim nächsten Start verbindet sich der Windsensor mit einem bestehendem Netzwerk

NMEA2000-Daten werden erstellt und ausgegeben

Sprache / Language

[Zur Windanzeige](#)

Alle vorgenommenen Einstellungen bleiben auch nach Neustart sowie Trennung vom Strom erhalten.

6.2.1. Offset

Unter „Vorzeichenbehafteter Offset der Windrichtung in Grad“ können ganze Zahlen mit oder ohne Vorzeichen eingegeben werden. „5“ wird dazu führen, dass immer fünf Grad auf den durch den AS5600 ermittelten Winkel aufaddiert werden, „-5“ führt zu einem Abzug von fünf Grad. Um die Änderungen anzuwenden, muss auf den Knopf „Anwenden“ geklickt werden.

6.2.2) Korrekturfaktor

Unter „linearer Korrekturfaktor der Windgeschwindigkeit“ kann eine beliebige Fließkommazahl eingegeben werden. Diese wird als Faktor zur ermittelten Windgeschwindigkeit hinzumultipliziert. Um die Änderungen anzuwenden, muss auf den Knopf „Anwenden“ geklickt werden.

6.2.3) Einheit der Windgeschwindigkeit

An dieser Stelle kann eingestellt werden, in welcher Einheit (Kilometer pro Stunde, Meter pro Sekunde oder Knoten) die Windgeschwindigkeit dargestellt werden soll.

Dies hat jedoch keine Auswirkungen auf die ausgesendeten NMEA-Datagramme, diese sind nicht veränderbar in Knoten.

6.2.4) Blinken der IP

Zwar besitzt der Windsensor einen mDNS-Server, der ihn, unabhängig von seiner IP unter „windsensor.local“ erreichbar macht, jedoch funktioniert das nur mit iOS-Geräten zuverlässig.

Um die Einrichtung des Windsensors trotzdem so einfach wie möglich zu gestalten, blinkt dieser beim Start seine ihm zugewiesene IP-Adresse ziffernweise mit der integrierten LED. Um auch Nullen darstellen zu können, wird immer einmal mehr geblinkt als der Wert der Ziffer angibt, Somit wird eine Null durch ein Blinken dargestellt, eine Eins zur zwei Blinken, eine Drei durch vier blinken etc.

Punkte zwischen den Blöcken werden durch dreimaliges kurzes Blinken schnell hintereinander signalisiert.

Das Blinken der IP verlangsamt den Startvorgang des ESP erheblich, da die gesamte IP zunächst geblinkt werden muss, bevor es weiter gehen kann. Daher ist es empfehlenswert, das Blinken der IP über die Einstellungsseite zu deaktivieren, sobald es nicht mehr benötigt wird. Dafür muss der „Ändern“-Knopf gedrückt werden. Dies sollte dazu führen, dass sich der Text von „Die IP wird beim Start geblinkt“ in „Die IP wird beim Start nicht geblinkt“ ändert.

6.2.5) Netzwerk- oder Standalone-Modus

An dieser Stelle kann eingestellt werden, ob der Windsensor beim Start versucht sich mit einem bestehenden Netzwerk zu verbinden oder ob er einen Access Point aufspannen soll.

6.2.6) NMEA 2000

Da das Erstellen der NMEA 2000-Daten sehr rechenintensiv ist, es aber auch durchaus möglich (wenn nicht gar wahrscheinlicher) ist, dass der Sensor „nur“ für NMEA 0183 verwendet wird, gibt es an dieser Stelle die Möglichkeit, die Verarbeitung von NMEA 2000 zu deaktivieren.

6.2.7) Sprache/Language

Die grafische Oberfläche des Windsensors ist sowohl auf Deutsch als auch auf Englisch verfügbar. Hier kann die Sprache umgestellt werden.