



EMOS
E06016

Inhaltsverzeichnis

1.	Vorwort	4
2.	Umbau des EMOS E06016	5
2.1.	Stückliste	5
2.2.	Zerlegen des Gehäuses	6
2.3.	Umbau der Sensoreinheit	7
2.3.1)	Durchtrennen des Kabels	7
2.3.2)	Einsetzen der Lüsterklemmen	7
2.3.3)	Einbau der Widerstände	8
2.3.4)	Anbringen der übrigen Jumperkabel und Schließen der Sensoreinheit	9
2.4.	Entfernen der Platinen und Anbringen des Magneten	9
2.5.	Verbinden und Befestigen der Richtungssensor-Lüsterklemmen	9
2.6.	Verschrauben des oberen Teils und Anschließen der Jumperkabel für NMEA0183	10
2.7.	Schaltplan	11
2.8.	Anbringen des AS5600	12
2.9.	Fixieren der Einheit	12
2.10.	Zusammenführen der beiden Hälften	13
3.	Software	14
3.1.	Vorüberlegung	14
3.2.	Vorbereitung des ESP32	14
3.3.	Herunterladen des Flash Tools	14
3.4.	Herunterladen der aktuellen Firmware	14
3.5.	Aufspielen der aktuellen Firmware	15
4.	Nutzung der Arduino-IDE	16
4.1.	Den Boardmanager für den ESP32 hinzufügen	16
4.2.	Die ESP32-Boards über den Boardmanager hinzufügen	16
4.3.	Korrektes Board wählen	16
4.4.	Die benötigten Bibliotheken herunterladen	17
4.5.	Das GitHub-Repo herunterladen	17
4.6.	Die benötigten Bibliotheken hinzufügen	18
4.7.	Vorbereitungen zum Flashen der Firmware	18
5.	Einrichtung	19
5.1.	TCP	19
5.2.	Standalone	19
6.	Grafische Oberfläche	20
6.1.	Anzeige der Winddaten	20
6.2.	Einstellungsseite	21
6.2.1.	Windsensor-Typ	21

6.2.2.	Offset.....	21
6.2.3.	Korrekturfaktor	21
6.2.4.	Mittelwertbilung der Windrichtung/Windgeschwindigkeit	22
6.2.5.	Einheit der Windgeschwindigkeit	22
6.2.6.	Blinken der IP.....	22
6.2.7.	Netzwerk- oder Standalone-Modus.....	22
6.2.8.	NMEA 2000.....	22
6.2.9.	Sprache/Language	22
6.2.10.	Messung von Temperatur und Luftfeuchtigkeit	22
6.2.11.	Mittelwertbildung der Temperatur/Luftfeuchtigkeit	23
6.2.12.	Korrekturfaktor für die Temperatur/Luftfeuchtigkeit	23
6.2.13.	Offset für die Temperatur/Luftfeuchtigkeit	23

1. Vorwort

Mit dem Ende der Verfügbarkeit des Ventus-Windsensors bestand Bedarf an einer Alternative. Der EMOS E06016 bot sich aufgrund der guten Verfügbarkeit und des geringen Preises an. Außerdem stellte sich heraus, dass aufgrund der Gehäuseform ein Umbau mit einer anderen AS5600-Platine in Betracht kam, die einen Umbau nicht nur ganz ohne 3D-Drucker, sondern auch ohne Lötkolben ermöglichen würde.

Der erste Versuch bestand darin, das Funksignal zu decodieren, um Informationen wie Windgeschwindigkeit, Temperatur und Luftfeuchtigkeit direkt von der dedizierten Hardware zu erhalten. Es stellte sich jedoch heraus, dass dies für mich nicht in einem vertretbaren Zeitaufwand möglich war, weshalb ein anderer Weg für die Messung von Luftfeuchtigkeit und Temperatur gefunden werden musste.

Dieses Problem ließ sich dank des Analog-Digital-Converters und der zwei Kerne des ESP32 jedoch technisch lösen.

Bei diesem Projekt handelt es sich, zum aktuellen Zeitpunkt, primär um eine Machbarkeitsstudie. Auch bietet es sich an, die Verbindungen zu verlöten und nicht, wie in der Anleitung beschrieben, mit Jumperkabeln zu stecken., sofern die Möglichkeit dazu besteht.

Die Formeln zur Berechnung der Temperatur bieten eine erstaunlich hohe Genauigkeit, bei der Luftfeuchtigkeit ist die Präzision aktuell noch nicht ganz so hoch. Aus Gründen der Einfachheit wurde auf eine Lookup-Tabelle verzichtet und stattdessen ein Korrekturpolynom verwendet, welches jedoch noch nicht in jeder Umweltsituation korrekte Werte liefert. Tendenzen und Trends zeichnen sich jedoch stets korrekt ab.

2. Umbau des EMOS E06016

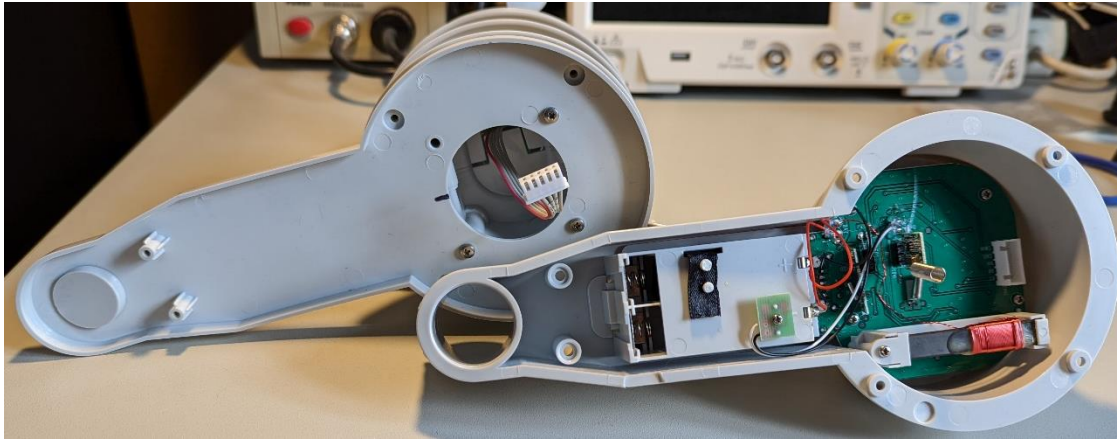
Der erste Teil der Dokumentation wird sich mit dem Umbau des W132 befassen. Wie in der Einleitung erwähnt hat der Umbau den Zweck, sowohl möglichst wenig zusätzliche Teile als auch möglichst wenig Werkzeug zu benötigen, und dabei gleichzeitig möglichst einfach nachbaubar und robust zu sein.

2.1. Stückliste

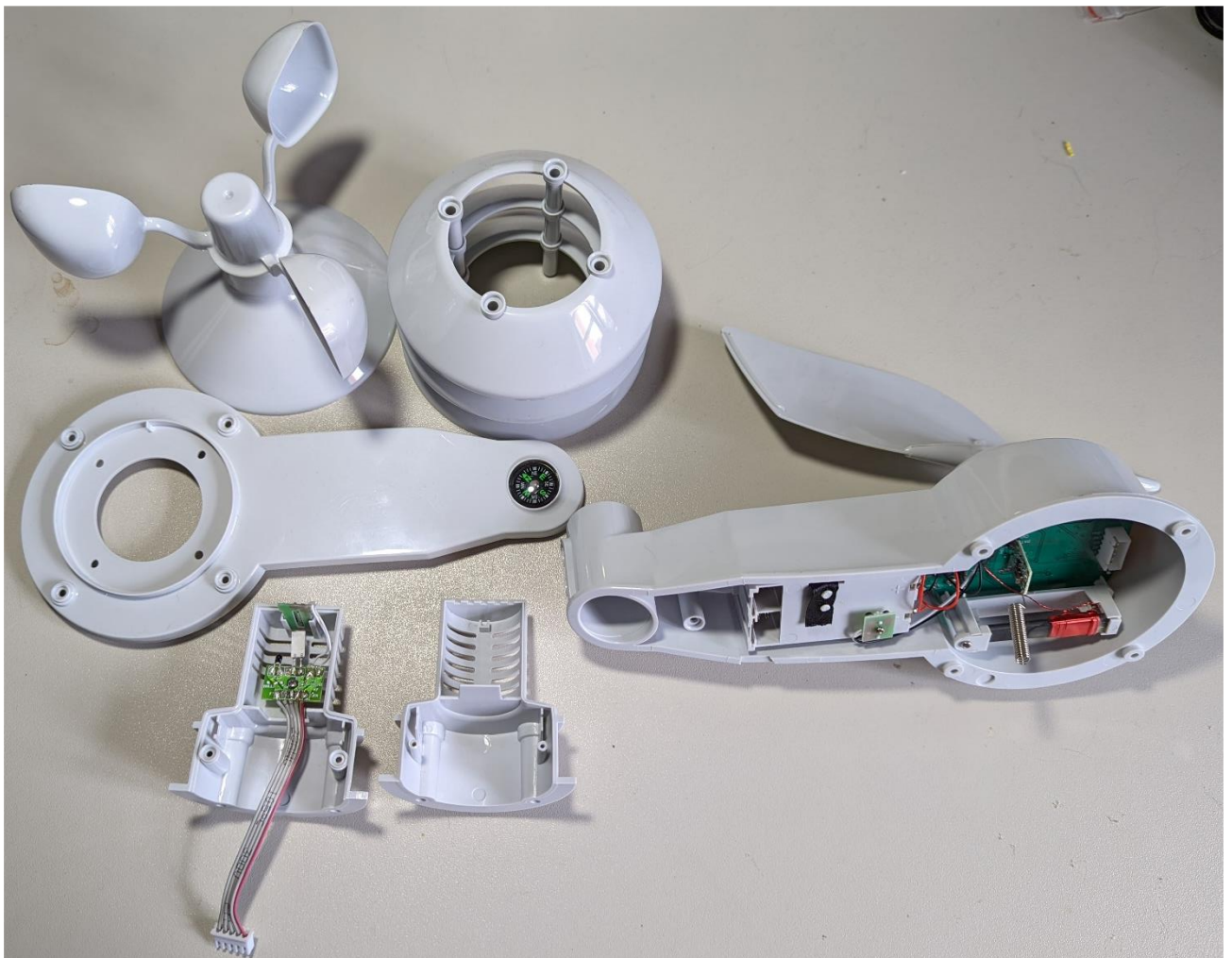
ID	Menge	Bezeichnung	Material	Bemerkung
Mechanikteile				
1	1	EMOS E06016		
3	1	Quadmagnet 5x5x3	Neodym	Auch ein diametraler Scheibenmagnet ist möglich
Elektronik				
4	1	ESP32		Das Pinout der Entwicklerbords unterscheidet sich
5	1	AS5600-Platine		Es gibt mehrere Platinen. Für diesen Umbau wird folgende Platine verwendet: https://www.seeedstudio.com/Grove-12-bit-Magnetic-Rotary-Position-Sensor-AS5600-p-4192.html
6	1	Widerstand 10k Ohm		
7	1	Widerstand 4,7M Ohm	t	
2	11	Lüsterklemmen		Möglichst klein
6	1	DC/DC-Wandler		von benötigter Eingangsspannung auf 5V
7	15	Jumperkabel		
NMEA 2000				
8	1	SN65HVD230 CAN BUS Modul		
9	4	Jumperkabel		
Software				
10	1	Firmware		https://github.com/jukolein/Emos/tree/main/Firmware
11	1	ESP32-Flasher		https://github.com/doayee/esptool-esp32-gui

2.2. Zerlegen des Gehäuses

Hierfür sind die sechs Schrauben auf der Unterseite zu lösen und das Kabel, welches in den oberen Bereich führt, vorsichtig von der Platine zu trennen.



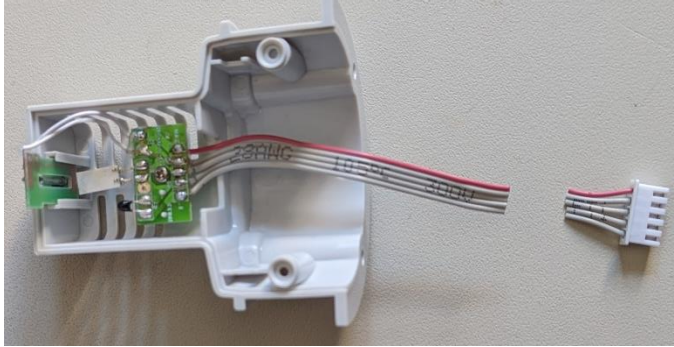
Anschließend werden die vier langen Schrauben im oberen Teil entfernt. Nun noch die zwei Schrauben, die das Gehäuse der Sensoren zusammenhalten, lösen, und man sollte diese Einzelteile erhalten:



2.3. Umbau der Sensoreinheit

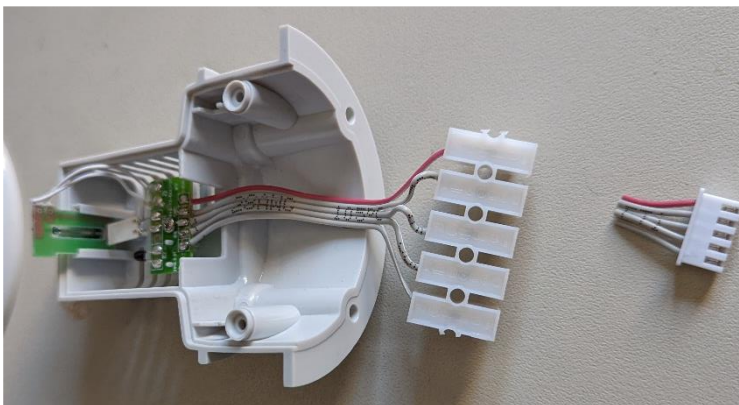
2.3.1) Durchtrennen des Kabels

Das Kabel wird etwa einen Zentimeter oberhalb des Steckers abgeschnitten.

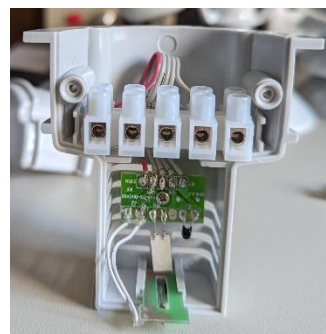


2.3.2) Einsetzen der Lüsterklemmen

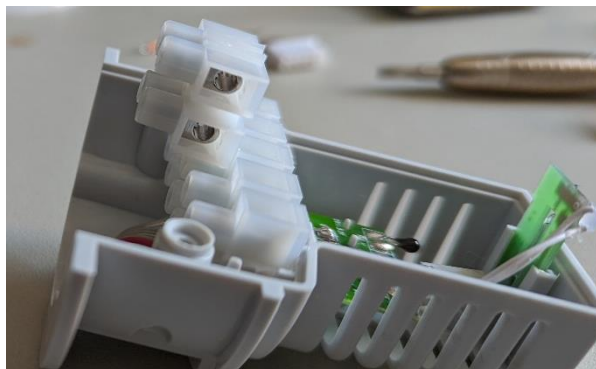
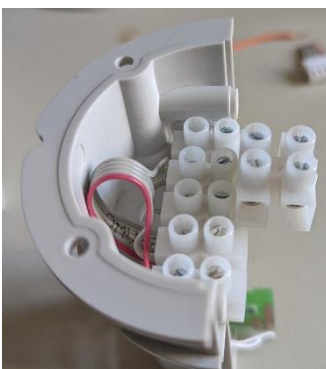
Die Kabelenden werden jeweils in einer Lüsterklemme fixiert. Hier bietet es sich an, eine zusammenhängende 5er-Reihe zu verwenden. Beim Verbinden ist darauf zu achten, dass die Schrauben, wie auf dem Bild gezeigt, nach unten zeigen.



Anschließend wird die Lüsterklemme in das Gehäuse eingeklebt.



Auf diese 5er-Reihe wird nun noch eine 2er-Reihe geklebt.

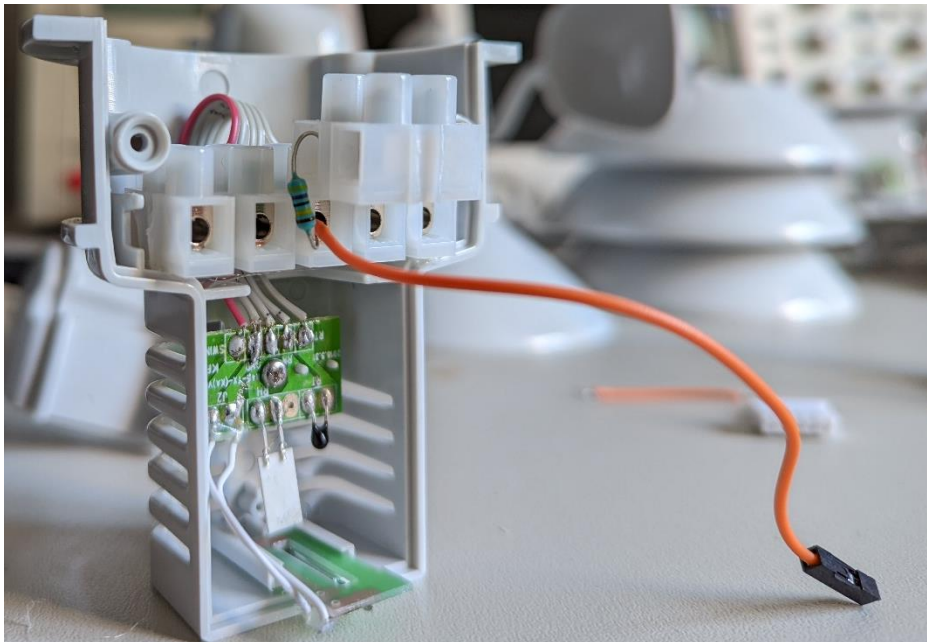


2.3.3) Einbau der Widerstände

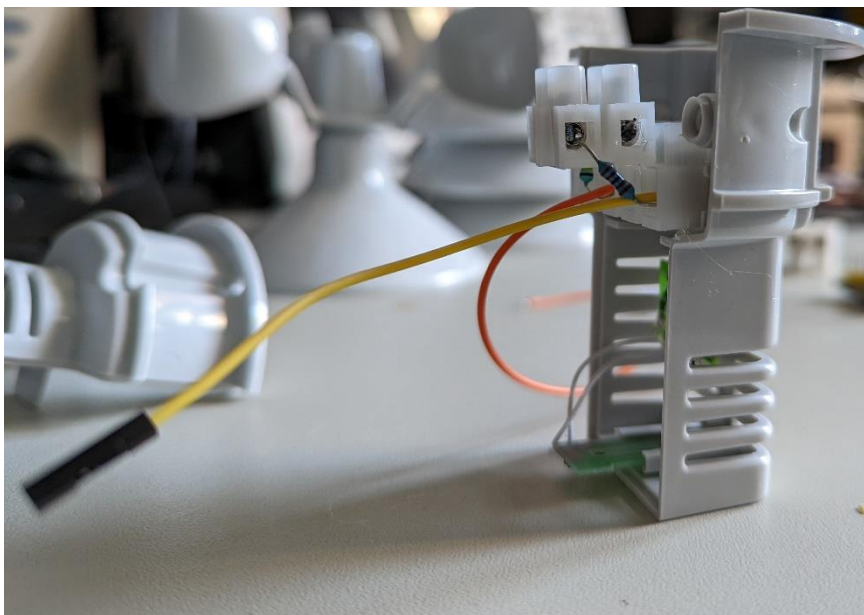
Der ESP32 ist in der Lage, mithilfe eines Analog-Digital-Converters (ADC) Spannungen im Bereich zwischen 0V und 3.3V zu messen. Diese werden dann einem Wert zwischen 0 und 4095 zugeordnet. Diese Spannung kann in einem Spannungsteiler dazu verwendet werden um den Widerstandswert des Feuchtigkeits- sowie des Temperatursensors zu berechnen. Da die hier verwendeten Sensoren ihren Widerstandswert in Abhängigkeit von Temperatur beziehungsweise Feuchtigkeit ändern, lässt sich daraus dann wiederum die Temperatur und die Luftfeuchtigkeit berechnen.

Damit der Spannungsteiler funktioniert, braucht es einen bekannten, festen Widerstand als Vergleichswert. Diese Widerstände werden nun eingebaut. Der Widerstand für den Temperatursensor hat einen Wert von 10k Ohm, der für den Feuchtigkeitssensor von 4,7M Ohm.

Der 4,7M Ohm-Widerstand wird mit der mittleren unteren Lüsterklemme und der inneren oberen Lüsterklemme verbunden. Zusätzlich wird ein Jumperkabel mit der unteren mittleren Lüsterklemme verbunden.

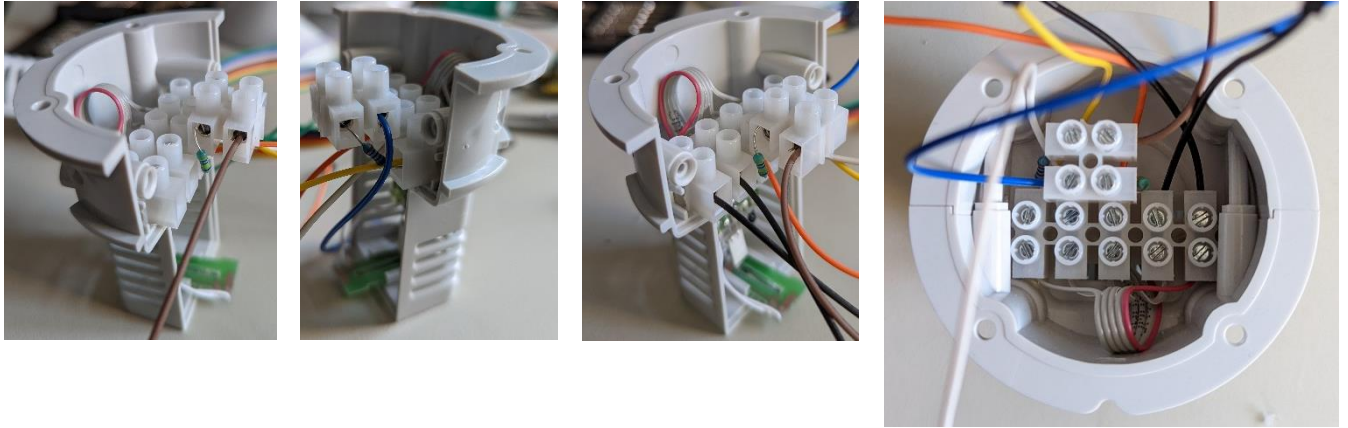


Der 10k Ohm-Widerstand wird mit der Lüsterklemme verbunden die sich, von vorne betrachtet, ganz rechts befindet, sowie mit der oberen, linken Lüsterklemme. Auch hier wird ein Jumperkabel mit der unteren, ganz rechten Lüsterklemme verbunden.



2.3.4) Anbringen der übrigen Jumperkabel und Schließen der Sensoreinheit

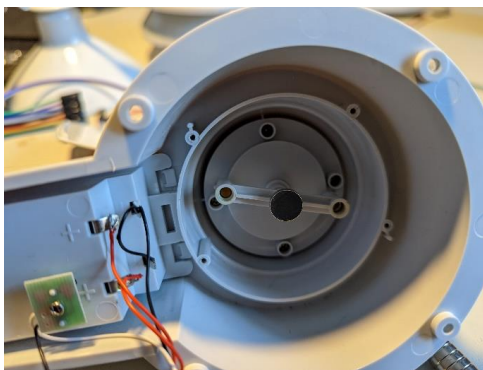
Abschließend müssen noch die verbleibenden fünf Ausgänge der Lüsterklemmen mit Jumperkabeln versehen werden. Ist dies geschehen, können beide Hälften der Sensoreinheit wieder verschraubt werden.



2.4. Entfernen der Platinen und Anbringen des Magneten

Die beiden Platinen werden nach dem Umbau nicht mehr benötigt und werden, samt Kabeln, entfernt.

Die Konstruktion dieses Windsensors macht es sehr einfach, einen Magneten geeignet zu platzieren. Und zwar mittig über dem Stift, der das Windrad mit den beiden sich drehenden Magneten verbindet.



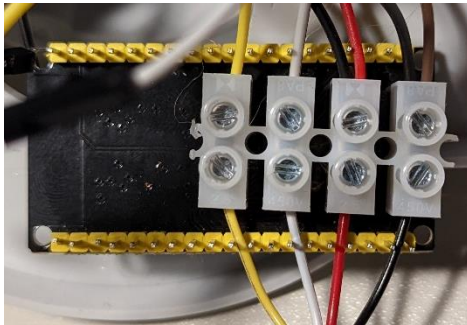
2.5. Verbinden und Befestigen der Richtungssensor-Lüsterklemmen

Das Kabel des AS5600-Magnetsensors wird etwa bei einem Drittel der Länge gekürzt und die Enden in der Lüsterklemme fixiert.



Auf der anderen Seite werden Jumperkabel verbunden.

Die Lüsterklemmen werden anschließend auf die Unterseite des ESP32 geklebt. Diese sollten sich möglichst nahe an dem USB-Anschluss befinden



Hinweis: Auf diesem Bild befindet sich noch ein Fehler; Das schwarze Kabel, welches mit der zweiten Lüsterklemme von rechts verbunden ist, muss mit der ganz rechten Klemme verbunden werden.

2.6. Verschrauben des oberen Teils und Anschließen der Jumperkabel für NMEA0183

An dieser Stelle kann nun der obere Teil des Sensors, also die Sensoreinheit, die drei Ringe und das Schalenrad, wieder verschraubt werden. Dies erleichtert im Folgenden die Handhabung des Sensors.

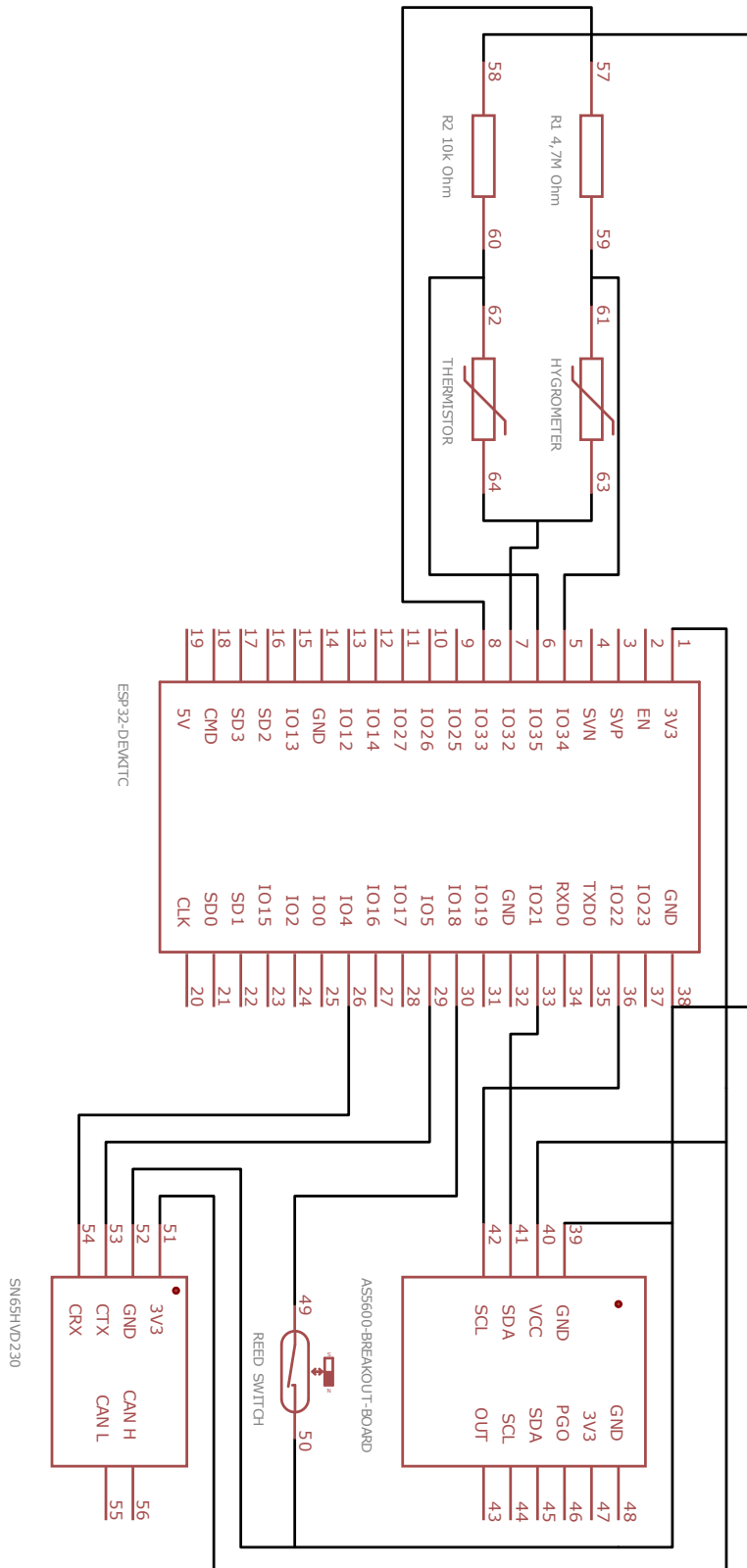
Anschließend werden nun sämtliche Kabel verbunden und angeschlossen, die für NMEA0183 benötigt werden.

„Farbe“ bezieht sich dabei auf die in dieser Dokumentation verwendeten und auf den Bildern sichtbaren Farben der Jumperkabel. Da es zu farblichen Dopplungen kommt, bezeichnet der Zusatz „AS5600“, dass es sich um die Jumperkabel handelt, die mit dem AS5600-Magnetsensor verbunden werden. Der Zusatz „Sensor“ bedeutet, dass die Jumperkabel zu dem Feuchtigkeits- Temperatur- und Hallsensor im oberen Bereich des EMOS führen.

Farbe	Pin am ESP32
Schwarz AS5600	GND
Rot AS5600	3v3
Weiß AS5600	21
Gelb AS5600	22
Orange Sensor	34
Gelb Sensor	35
Weiß Sensor	32
Blau Sensor	33
Braun Sensor	GND (kann in der Lüsterklemme mit Schwarz AS5600 verbunden werden)
1. Schwarz Sensor	GND (kann in der Lüsterklemme mit Schwarz AS5600 verbunden werden)
2. Schwarz Sensor	18

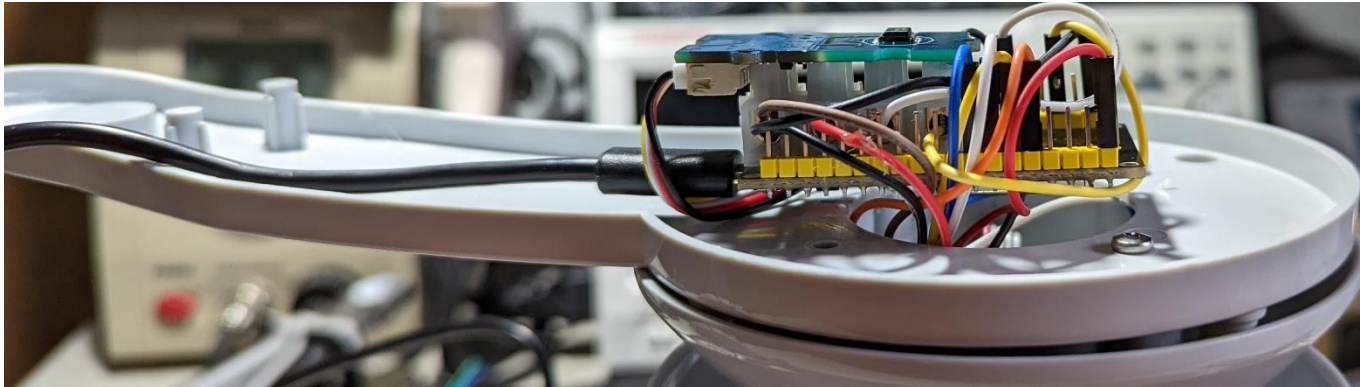
2.7. Schaltplan

Auf diesem Schaltplan ist auch der Anschluss des CAN-Busses für die Verwendung von NMEA2000 abgebildet. Der tatsächliche Einbau ist zwar auf keinem der Bilder dieser Anleitung zu sehen, da es sich dabei jedoch „nur“ um das Anschließen von vier Jumperkabeln handelt, sollte dies jedoch hoffentlich kein all zu großes Problem darstellen.



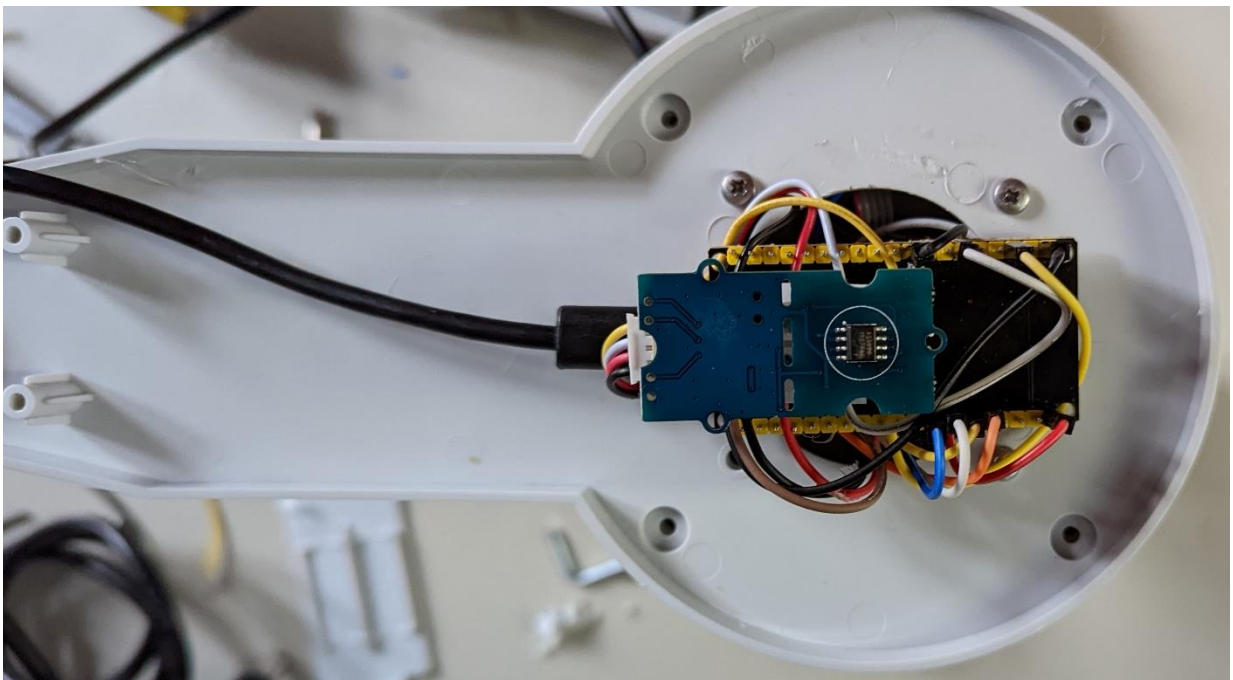
2.8. Anbringen des AS5600

Um eine kompakte Einheit zu erhalten, wird der AS5600 nun auf den Lüsterklemmen auf dem ESP32 fixiert. Dabei muss darauf geachtet werden, dass sich der AS5600 möglichst im hinteren Drittel (also weg vom USB-Anschluss) befindet. Dadurch kann die ganze Einheit im nächsten Schritt etwas nach oben versetzt fixiert werden, um mehr Platz für den USB-Anschluss zu lassen.



2.9. Fixieren der Einheit

Anschließend wird der Verbund aus ESP32, Lüsterklemmen und AS5600 auf dem Gehäuse verklebt. Der AS5600 muss sich dabei exakt über dem Magneten befinden, wenn beide Hälften wieder zusammengebaut werden.



2.10. Zusammenführen der beiden Hälften

Das USB-Kabel kann beispielsweise durch den Batterieschacht nach außen geführt werden



Anschließend können beide Hälften wieder aufeinandergesetzt und verschraubt werden.

Der Umbau des EMOS ist damit, zumindest für die Verwendung von NMEA 0183, abgeschlossen.

Um die Daten auch per NMEA2000 auszugeben, müssen lediglich noch die vier Jumperkabel für 3v3, GND, CTX und CRX angeschlossen werden (siehe 2.7, „Schaltplan“).

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.

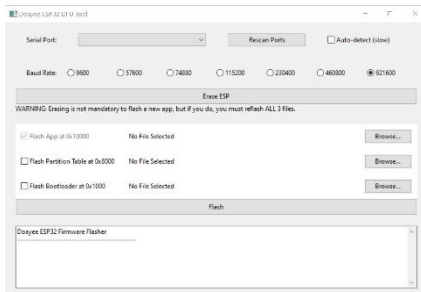
3.4. Herunterladen der aktuellen Firmware

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

3.5. Aufspielen der aktuellen Firmware

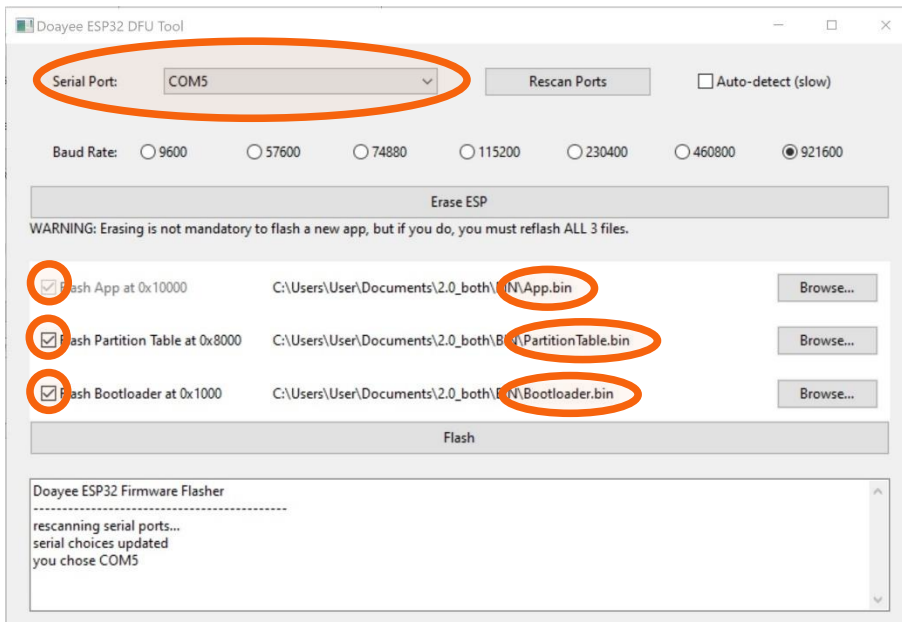
Dieser Schritt ist, wie bereits angesprochen, beim ESP32 etwas komplexer, wird jedoch durch die grafische Benutzeroberfläche des Programms stark vereinfacht.

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

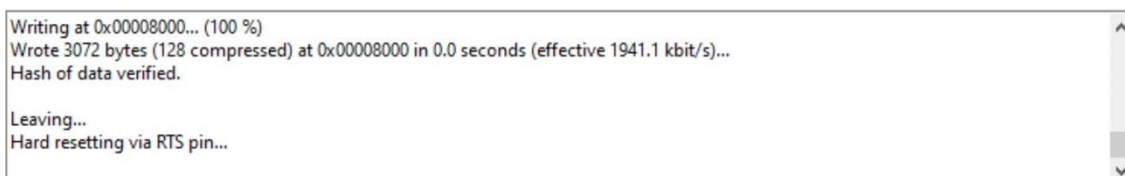


Hierbei müssen nun der korrekte Serielle Port und alle drei Dateien ausgewählt und die Haken davor gesetzt werden.

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.

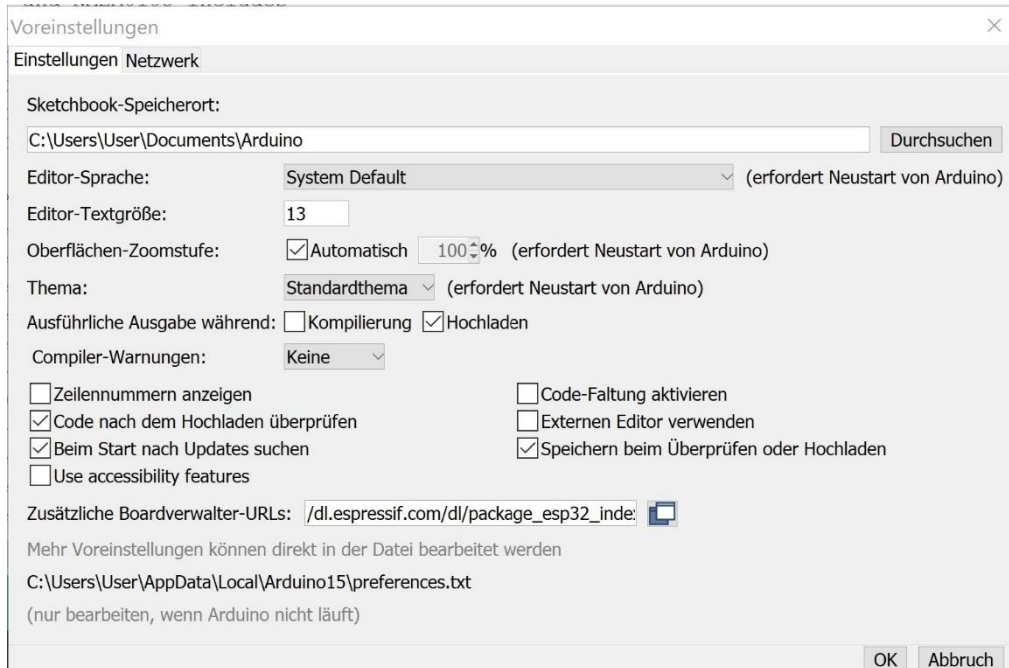
Nähere Informationen zur Verwendung der Arduino IDE folgen im nächsten Kapitel.

4. Nutzung der Arduino-IDE

4.1. Den Boardmanager für den ESP32 hinzufügen

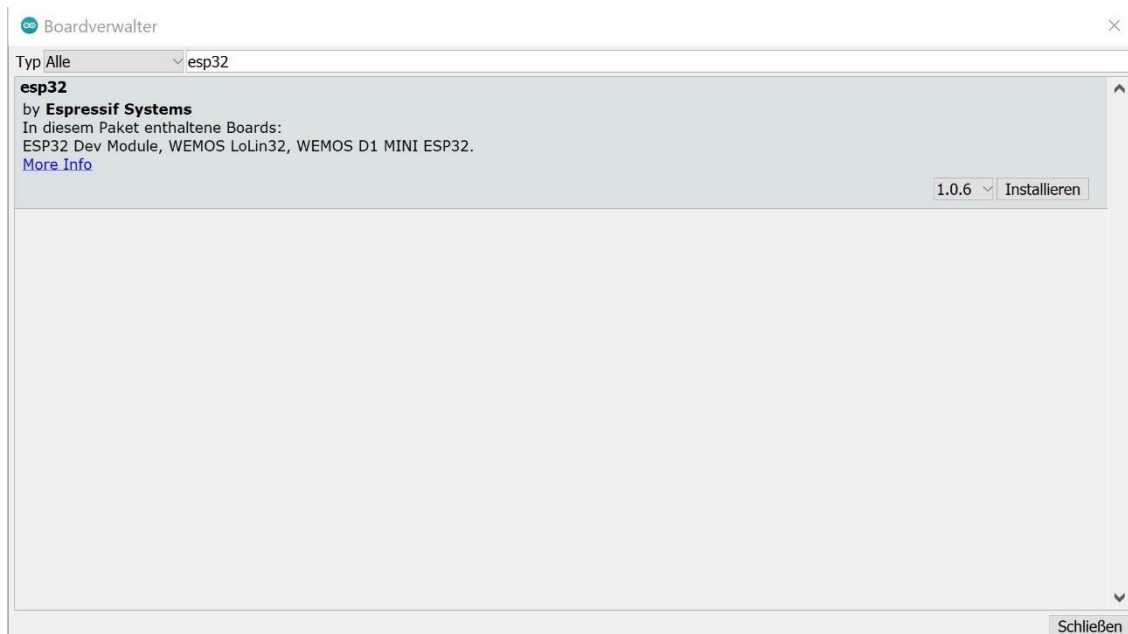
Hierfür muss unter „Datei --> Voreinstellungen --> zusätzliche Boardverwalter-URLs“ folgende URL eingegeben werden: https://dl.espressif.com/dl/package_esp32_index.json

Anschließend mit „OK“ bestätigen.



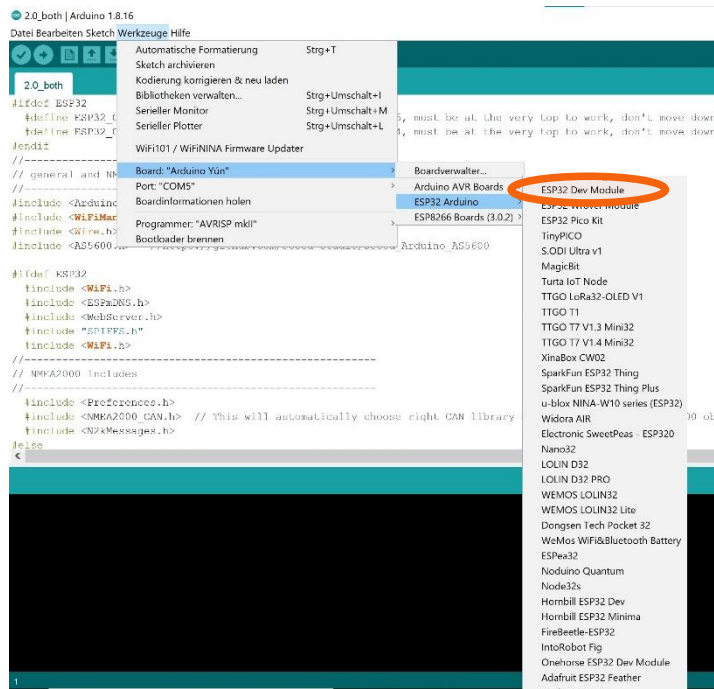
4.2. Die ESP32-Boards über den Boardmanager hinzufügen

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

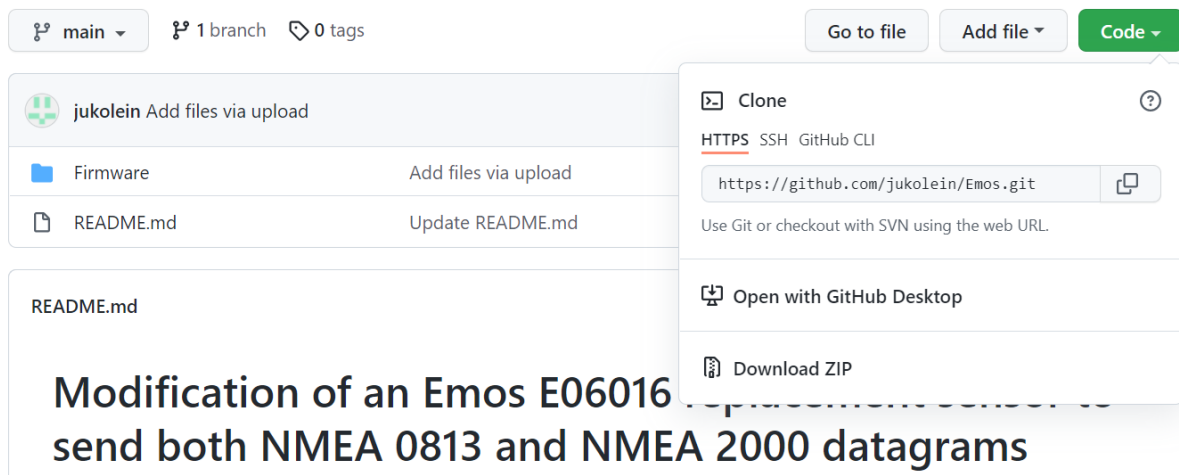
Es werden vier zusätzliche Bibliotheken verwendet: „WiFiManager“, „AS5600“, „NMEA2000“ und „NMEA2000_esp32“

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.

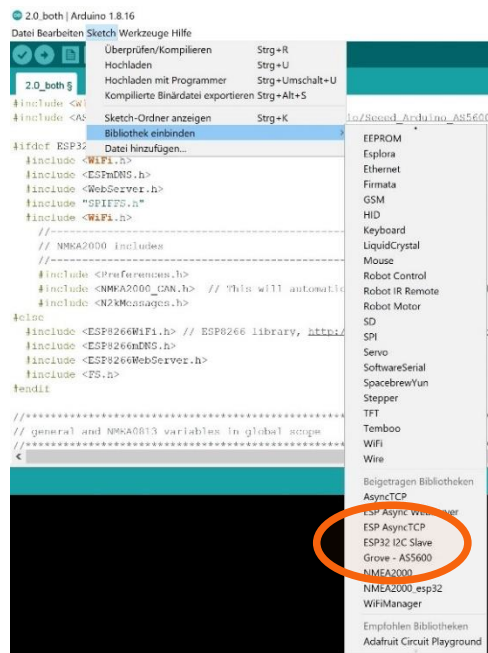


4.6. Die benötigten Bibliotheken hinzufügen

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.

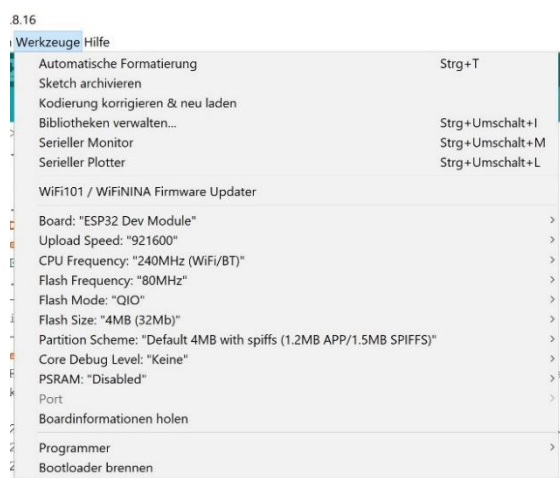
Anschließend sollten diese unter „Sketch --> Bibliotheken einbinden“ im Abschnitt „Beigetragene Bibliotheken“ aufgelistet sein.



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“.

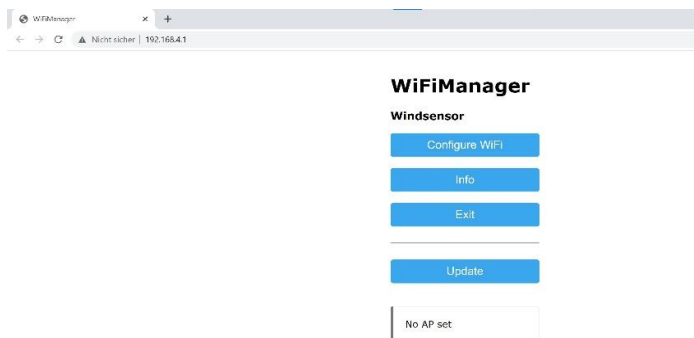


Nach erfolgreicher Verbindung mit diesem Netzwerk sollte man automatisch auf „192.168.4.1“ weitergeleitet werden. Ist dies nicht der Fall, so muss die IP manuell aufgerufen werden. Wie es danach weiter geht, hängt davon ab, welcher Nutzungsmodus (Integration in ein bestehendes Netzwerk oder Standalone) 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.

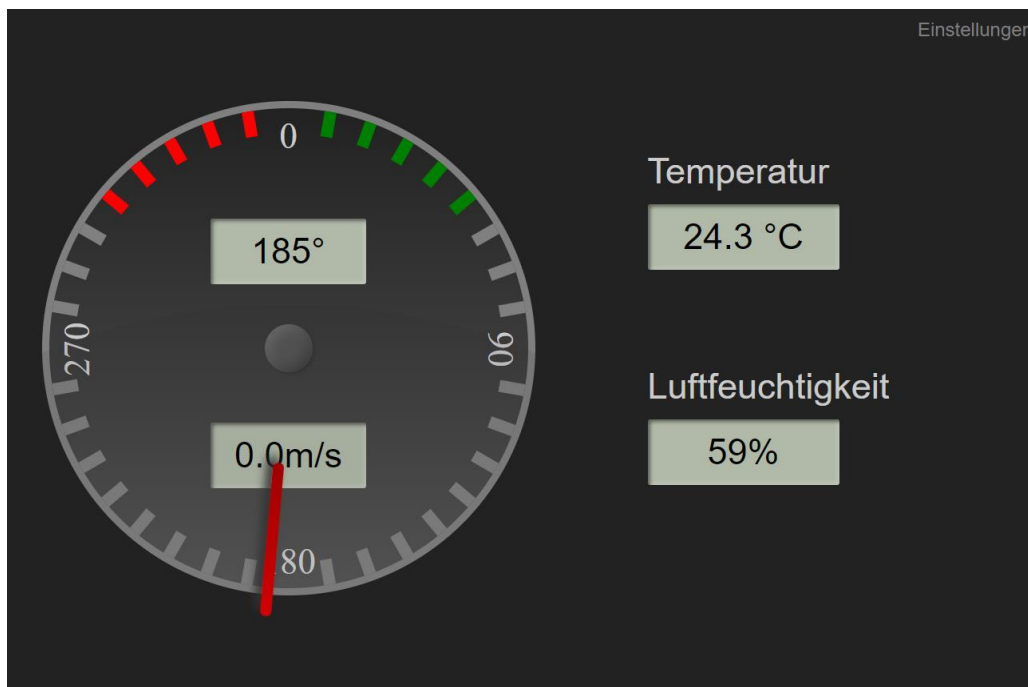
Um den Sensor dauerhaft im Standalone-Modus zu halten, muss dies noch in der Einstellungsseite aktiviert werden.

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.

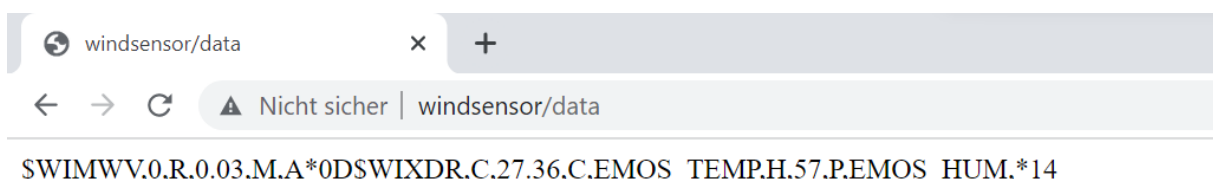


Angezeigt wird der scheinbare Wind in Grad sowie die aktuelle Windgeschwindigkeit in m/s, km/h oder kt. Außerdem werden, falls aktiviert, auch die Luftfeuchtigkeit und die Temperatur angezeigt.

Vielen Dank an ziegenhagel.com für das Programmieren der Oberfläche.

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

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.



6.2. Einstellungsseite

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

Emos

Einstellungen

Aktuell ausgewählter Windsensor-Typ: Emos

Vorzeichenbehafteter Offset der Windrichtung in Grad:

linearer Korrekturfaktor der Windgeschwindigkeit:

Mittlung der Windrichtung über Werte

Mittlung der Windgeschwindigkeit über Werte

Aktuelle Einheit der Windgeschwindigkeit: Meter pro Sekunde

Aktuell wird die IP beim Start nicht geblinkt

Beim nächsten Start verbindet sich der Windsensor mit einem bestehendem Netzwerk

NMEA2000-Daten werden nicht erstellt oder ausgegeben

Sprache / Language

Temperatur und Luftfeuchtigkeit werden gemessen

Mittelwertbildung der Temperatur über Werte

Mittelwertbildung der Luftfeuchtigkeit über Werte

Korrekturfaktor für die Temperatur:

Offset für die Temperatur

Korrekturfaktor für die Luftfeuchtigkeit:

Offset für die Luftfeuchtigkeit

[Zur Windanzeige](#)

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

6.2.1. Windsensor-Typ

Hier kann der aktuell verwendete Windsensor ausgewählt werden. Dies hat Einfluss auf die Voreinstellungen zur Berechnung der Windgeschwindigkeit sowie auf die angezeigten Optionen, da einige Einstellungen nur für einige Typen sinnvoll sind.

6.2.2. 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.

6.2.3. 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.

6.2.4. Mittelwertbildung der Windrichtung/Windgeschwindigkeit

Hier kann ein ganzzahliger Wert zwischen 1 und 500 eingegeben werden. Über die angegebene Anzahl von Werten wird der Mittelwert gebildet und entsprechend ausgegeben. Große Werte sorgen also für ein „glatteres“, dafür aber auch trägeres Resultat.

6.2.5. 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.

6.2.6. 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.

6.2.7. 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.

6.2.8. 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.

6.2.9. Sprache/Language

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

6.2.10. Messung von Temperatur und Luftfeuchtigkeit

Nur für den Sensor „Emos“ verfügbar. Falls aktiviert, werden die Temperatur und die Luftfeuchtigkeit über die integrierten Sensoren gemessen und ausgegeben.

6.2.11. Mittelwertbildung der Temperatur/Luftfeuchtigkeit

Hier kann je ein ganzzahliger Wert zwischen 1 und 500 eingegeben werden. Über die angegebene Anzahl von Werten wird der Mittelwert gebildet und entsprechend ausgegeben. Große Werte sorgen also für ein „glatteres“, dafür aber auch trägeres Resultat.

6.2.12. Korrekturfaktor für die Temperatur/Luftfeuchtigkeit

Hier kann je eine beliebige Fließkommazahl eingegeben werden. Die ermittelte Temperatur/Luftfeuchtigkeit wird dann mit diesem Wert multipliziert und kann somit auf den gegebenen Sensor angepasst werden.

6.2.13. Offset für die Temperatur/Luftfeuchtigkeit

Hier kann je ein vorzeichenbehafteter ganzzahliger Offset für die Temperatur und die Luftfeuchtigkeit angegeben werden. Dieser wird nach der Berechnung auf das Ergebnis addiert, falls es eine positive Zahl war, oder subtrahiert, falls eine negative Zahl eingegeben wurde.