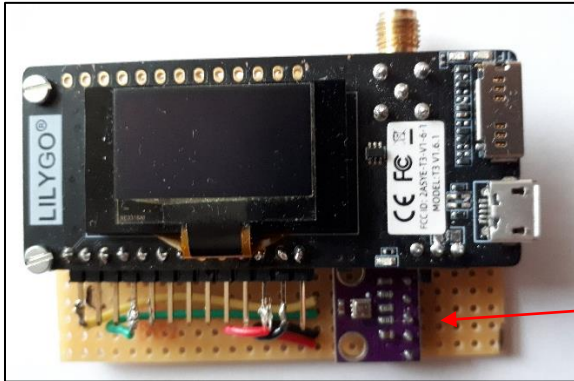


Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

iGate + BME280



BME280

von unten:

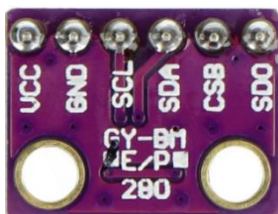
PINOUT TTGO LoRa32 433 Mhz V1.61

ADCL0	GPIO36	S_VP	36	21	SDA	GPIO21	V_SPLND
ADCL3	GPIO39	S_VN	39	TX0	CLX3	GPIO01	VO_TXD
ADCL6	GPIO34		34	RXD	CLX2	GPIO03	VO_RXD
ADCL7	GPIO35		35	22	SCL	GPIO22	V_SPLUP
TOUCH6	ADC2_6	GPIO14	14	23	GPIO23	V_SPLD	
TPUCHS	ADC2_5	GPIO12	12	19	GPIO19	V_SPLQ	VO_CTS
TPUCH4	ADC2_4	GPIO13	13	LoRa2			
TOUCH3	HSP_LCS0	GPIO15	15	LoRa1			
TOUCH2	HSP_LWP	ADC2_2	02	26	GPIO26	ADC2_9	DAC1
TOUCH1	ADC2_1	GPIO08	08	GND			LoRa_DIO
TOUCH0	HSP_LHD	ADC2_0	04	3.3V			
LED	DAC2	ADC2_8	01025	GND			
				5V			

MOSI	27
SCLK	5
CS	18
DIO	26
RST	23
MISO	19

SDA	21
SCL	22

CS	MOSI	SCK	MISO
IO13	IO15	IO14	IO02



Der **BME280** ist ein Temperatur-, Luftfeuchte- und Luftdrucksensor. Neben der Messung der Luftfeuchte mit $\pm 3\%$ Genauigkeit und der einer Temperaturmessung von $\pm 1,0^\circ\text{C}$ Genauigkeit stellt er auch einen Messwert für den Luftdruck mit ± 1 hPa absoluter Genauigkeit zur Verfügung. Er wird hier verwendet, um WX-Daten auszusenden. In der nachfolgenden Beschreibung wird erklärt, wie man hier auch einen relativen Luftdruck einstellen kann.

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 2 von 12

Der BME280 wird über eine I²C-Leitung an das iGate-Board angeschlossen. Er wird mit einer Betriebsspannung von 3,3V versorgt.

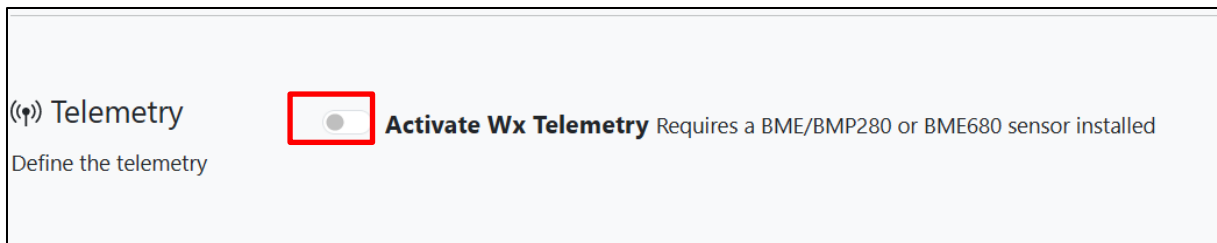
Es müssen also folgende Verbindungen hergestellt werden:

<u>BME280</u>	<u>iGate</u>
VCC	3,3V
GND	GND
SCL	SCL (Pin 22)
SDA	SDA (Pin 21)
CSB/SDO	--

Die I²C-Adresse ist standardmäßig: 0x76 (in der Firmware von CA2RXU bereits berücksichtigt)

In der Firmware müssen folgende Einstellungen vorgenommen werden:

- Aufrufen der IP-Adresse **192.168.2.101** mit dem Browser.
- Aktivierung der WX Telemetry



In der `igateConf.json` – falls man über VSC (Visual Studio Code) konfiguriert – muss bei „bme“ **true** statt **false** geschrieben werden.

```
59     "bme": {  
60         "active": true
```

Hinweis:

Da der Luftdrucksensor den absoluten Wert liefert, stimmt der angegebene Druck nicht mit den Werten der von den umliegenden Wetterstationen angegebenen relativen Werten überein.

In der `bme_utils.cpp` (Ordner `scr`) verwendet CA2RXU diese Formel für den angezeigten Luftdruck:

```
presStr = generatePresString(newPress + (HEIGHT_CORRECTION/CORRECTION_FACTOR));
```

Wobei „newPress“

```
140     newPress = (bme.readPressure() / 100.0F);
```

der vom Sensor gemessene absolute Wert (/100) in hPa ist.

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 3 von 12

Belässt man nun die **HEIGHT_CORRECTION** in Zeile 7 bei **0**

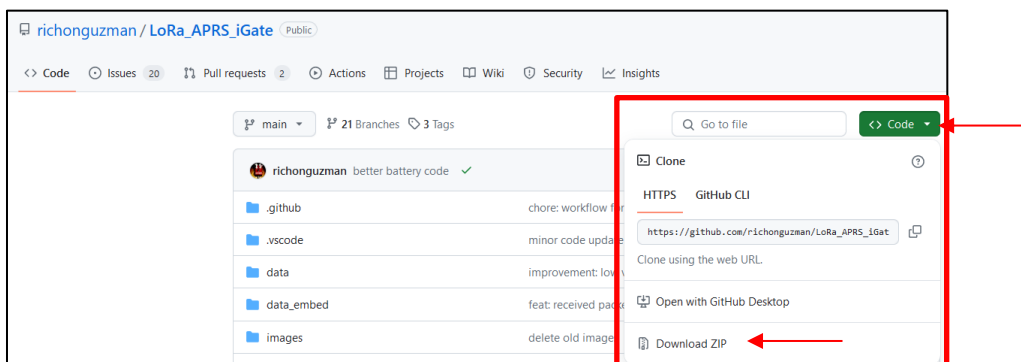
```
bme_utils.cpp ×
src > bme_utils.cpp > ...
1 #include "bme_utils.h"
2 #include "configuration.h"
3 #include "gps_utils.h"
4 #include "display.h"
5
6 #define SEALEVELPRESSURE_HPA (1013.25)
7 #define HEIGHT_CORRECTION 0 // in meters
8 #define CORRECTION_FACTOR (8.2296) // for meters
```

wird **nur dieser absolute** Wert angezeigt und ein Vergleich mit den Werten der umliegenden Wetterstationen ist somit nicht direkt möglich.

Im Moment (01.05.2024) finde ich keine Möglichkeit, dies beim Flashen der Firmware über die .bat-Dateien oder den Web Installer zu korrigieren. Nur, wenn man auch beim iGate den Weg über den VSC geht, kann eine Korrektur in der cpp-Datei vorgenommen werden. Damit wäre aber die einfache Firmware-Übertragung nicht möglich. Schade! Die einzige Lösung, die ich momentan sehe: Man verwendet einmalig – beim 1. Mal – den Weg über den VSC und macht danach nur Updates. Dann wird die bme_utils.cpp nicht verändert.

Anleitung, um den auf NN reduzierten Luftdruck einzustellen:

Lade von https://github.com/richonguzman/LoRa_APRS_iGate die zip-Datei herunter und entpacke sie:



Für den Fall, dass man diese Datei erneut benötigt und am Namen erkennen soll, welche Version das ist, benenne ich üblicherweise den entpackten Ordner so um, dass ich am Namen bereits das Versionsdatum erkenne. Dieses Datum findet sich im Unterordner **src** in **LoRa_APRS_iGate.cpp** als **String versionDate**.



Dieses Datum hänge ich an den Dateinamen an:



Jetzt öffnet man im VSC diesen umbenannten Ordner und findet wiederum im Unterordner **src** die **bme_utils.cpp**. In der Zeile 7 ist die Korrektur des Luftdrucks für höhergelegene Orte (über NN) zu ändern:

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 4 von 12

```
src > bme_utils.cpp > ...
1 #include "bme_utils.h"
2 #include "configuration.h"
3 #include "gps_utils.h"
4 #include "display.h"
5
6 #define SEALEVELPRESSURE_HPA (1013.25)
7 #define HEIGHT_CORRECTION 0 // in meters
8 #define CORRECTION_FACTOR (8.2296) // for meters

src > bme_utils.cpp > ...
1 #include "bme_utils.h"
2 #include "configuration.h"
3 #include "gps_utils.h"
4 #include "display.h"
5
6 #define SEALEVELPRESSURE_HPA (1013.25)
7 #define HEIGHT_CORRECTION 604 // in meters
8 #define CORRECTION_FACTOR (8.2296) // for meters
```

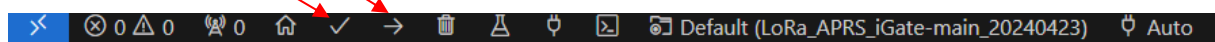
Die „604“ m gelten für meine Lage über NN.

Über Datei/Speichern wird das alles gesichert.

Nun kann man diese korrigierte Firmware **nach der entsprechenden Anleitung zur Installation über VSC** auf das Board flashen. (Hochladen der igate_conf.json und des Filesystems). Wie das geht, steht in https://github.com/richonguzman/LoRa_APRS_iGate/wiki/01.-Installation-Guide bzw. im nachfolgenden Text.

Das Board sollte spätestens jetzt über USB am PC angeschlossen werden.

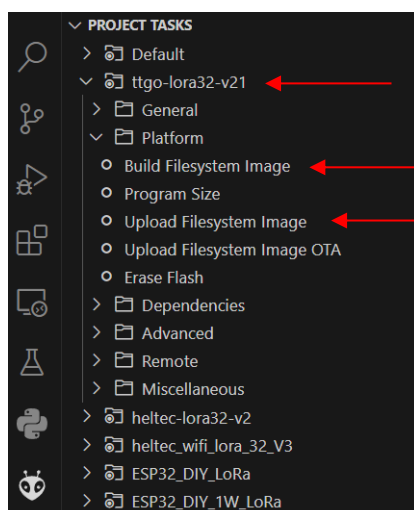
Mit dem **Häkchen** kann ein **Build** der Konfiguration erzeugt werden und wenn die positive Erfolgsmeldung kommt, kann mit dem Hochladen der Firmware begonnen werden. Dazu klickt man auf den **Pfeil nach rechts**.



Jetzt muss noch das Filesystem hochgeladen werden:

CA2RXU:

Auf der linken Seite des VSCODE-Bildschirms befinden sich einige Symbole. Suchen Sie das PlatformIO-Symbol (Kopf einer Ameise) und drücken Sie darauf. Ein neues Menü wird angezeigt. Suchen Sie nach **Project Task**, dann nach **Platform** und dort sollten Sie **Upload Filesystem Image** finden. Drücken Sie darauf, und es wird die JSON-Datei mit allen Daten hochgeladen, die Sie im **iGate-Konfigurationsteil** geändert haben. [Mit einem Klick auf **Build Filesystem Image** kann auch hier vorher ein Test erfolgen.]

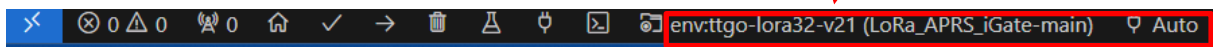


HINWEIS: Manchmal hat VSCODE hier einen Fehler. Wenn es mit dem Hochladen nicht funktioniert, schließen Sie VSCODE, öffnen Sie VSCODE erneut, und versuchen Sie den letzten Teil erneut. Wenn alles gut gelaufen ist, sollten Sie die Willkommensnachricht auf dem iGate-Bildschirm sehen.

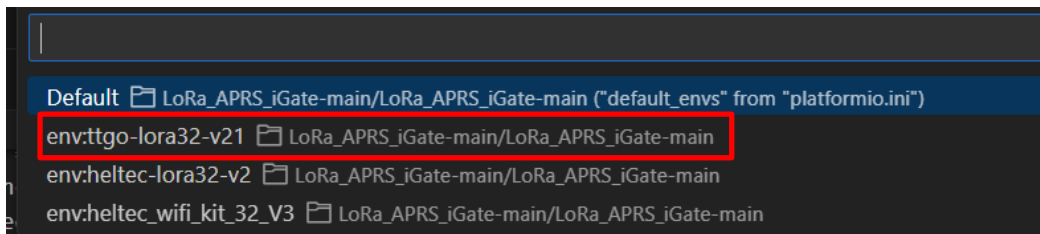
Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 5 von 12

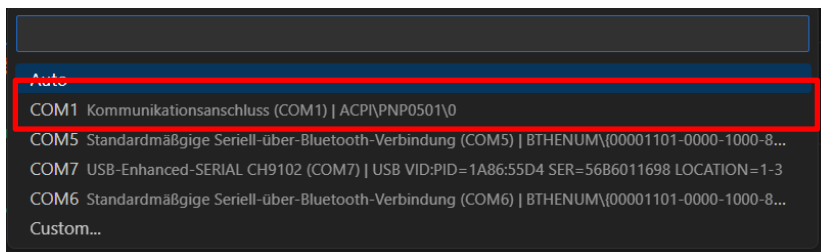
In jedem Fall muss vorher darauf geachtet werden, dass das richtige Board verwendet wird und statt „Auto“ auch der richtige COM-Port eingestellt ist.



Falls beim Board noch „Default“ steht, einfach draufklicken und in der sich oben auf der Seite öffnenden Auswahl die „ttgo-lora32-v21“ auswählen. Entsprechendes gilt für den Port.



[Der richtige Port (hier COM7) wird erst oben angezeigt, wenn das iGate an USB angeschlossen ist.]



Seitdem Ricardo auf den Auto-AP (s. igateConf.json) umgestellt hat und hier keine direkten Konfigurationen eines WiFi-Netzes mehr möglich sind, erscheint auf dem iGate-Bildschirm Folgendes:



NOCALL-10 ist die SSID des Auto-AP-Netzwerkes.
Bei WiFi steht nur „AP“ und bei APRS-IS steht „OFF“. Die IP-Adresse ist: **192.168.4.1**.
Im Gegensatz zur bisherigen Installation (vor März 2024) fehlt in der `igate_config.json` zunächst die Eintragung eines WiFi-APs.

Wie es ab hier weiter geht, ist in meiner Dokumentation (ab Seite 2) zum Flashen mit Hilfe der Batch-Dateien zu finden: [DD1GS_LoRa_firmware_CA2RXU_iGate_install_bat.pdf](#)

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 6 von 12

OLED 1.3“-Display für Tracker

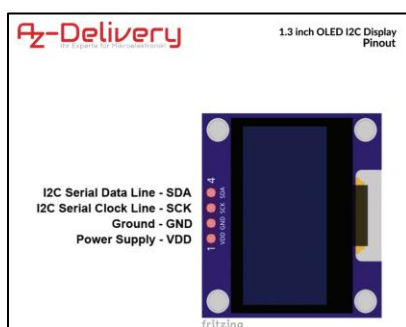
Es ist möglich, statt des üblichen 0.96“ OLED-Displays auch ein 1.3“-Display zu verwenden.

- Änderungen sind in `src/display.cpp` vorzunehmen:

```
10 //#define ssd1306 //comment this line with "//" when using SH1106 screen instead of SSD1306
```

```
49 #else  
50 Adafruit_SH1106G display(128, 64, &Wire, -1;  
51 #endif  
52
```

Die „-1“ ist notwendig, wenn kein 5. Pin **RST** vorhanden ist.



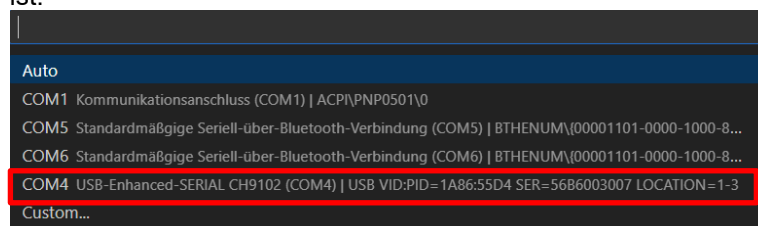
Hinweis:

Am besten ein OLED mit eingebautem Pegelwandler (5V – 3,3V) und nur 4 Pins verwenden. Dann hat man keine Probleme mit dem Anschluss an Arduinos, TTGO-Boards usw., die mit einem 5V-Pegel arbeiten.

- Damit hat man eine **spezielle** Version für das OLED 1.3“ (mit den Anschlüssen: SDA, SCK, GND, VDD).
- Um zu testen, ob alles korrekt ist, kann vor dem Flashen ein Build erstellt werden. In jedem Fall muss vorher darauf geachtet werden, dass das richtige Board verwendet wird und statt „Auto“ auch der richtige COM-Port eingestellt ist.

```
env:ttgo-t-beam-v1_2 (LoRa_APRS_Tracker-main) COM4
```

Falls beim Board noch „Default“ steht, einfach draufklicken und in der sich oben öffnenden Auswahl die „ttgo-t-beam-v1_2“ auswählen. Entsprechendes gilt für den Port. Der richtige Port (hier COM4) wird erst oben angezeigt, wenn der Tracker an USB angeschlossen ist:



- Ob die aktuellste Version auch verwendet wurde, ist hier `src/LoRa_APRS_Tracker.cpp` zu finden (oder beim Booten zu sehen):

```
LoRa_APRS_Tracker.cpp 37 String versionDate = "2024.01.22";
```

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 7 von 12

LEDs und Buzzer für Tracker

Meine beiden Tracker habe ich um 2 LEDs (für TX und Nachrichten) und einen Buzzer erweitert. Der Buzzer läuft ebenfalls wie das OLED-Display über den I²C-Bus, In der **tracker_config.json** kann er aktiviert werden:

```
> .vscode      84      },
  > data        85      "notification": {
    { tracker_config.json 86      "ledTx": true,
  > extra       87      "ledTxPin": 13,
  > images      88      "ledMessage": true,
  > lib         89      "ledMessagePin": 2,
  > src         90      "ledFlashlight": false,
  >             91      "ledFlashlightPin": 14,
  >             92      "buzzerActive": true,
  >             93      "buzzerPinTone": 33,
  >             94      "buzzerPinVcc": 25,
  >             95      "bootUpBeep": true,
  >             96      "txBeep": true,
  >             97      "messageRxBeep": true,
  >             98      "stationBeep": true,
  >             99      "lowBatteryBeep": true
  >             100     }
```

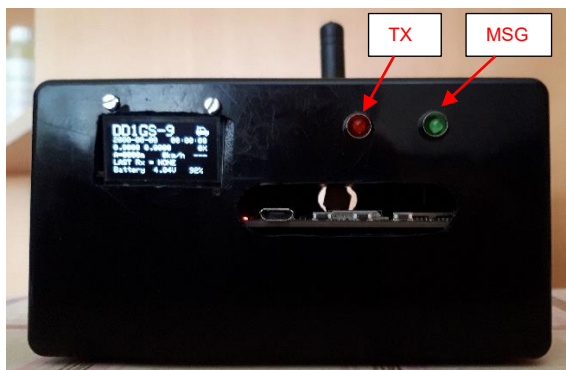
Die LEDs sind an die entsprechenden Pins anzuschließen und müssen auch aktiviert werden. Bei einem zweiten Tracker habe ich noch andere Ergänzungen/Umbauten vorgenommen:

- PA mit ca. 2W
- Anschluss für die externe Stromversorgung
- Anschluss für eine aktive größere GPS-Antenne mit Magnetfuß (fürs Autodach)
- Anschluss für eine HF-Antenne ebenfalls mit Magnetfuß (fürs Autodach)

Bei den Bildern am Schluss, werden diese Erweiterungen gezeigt. Ein einfacher „Schaltplan“ ist ebenfalls dargestellt.

Um die in die Gehäuse eingebauten Teile wieder leicht entfernen zu können, erfolgen die Verbindungen mit Jumper Wires. Das Ganze ist auf Lochraster-Platinen (Streifen) montiert, so dass letztlich das bloße Board separat entnommen werden kann.

Tracker 1 mit Buzzer und LEDs

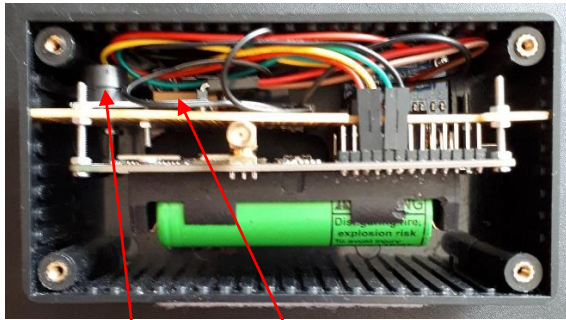


Vorderseite

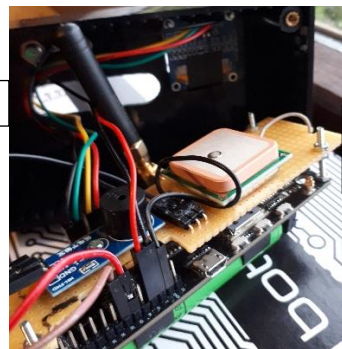
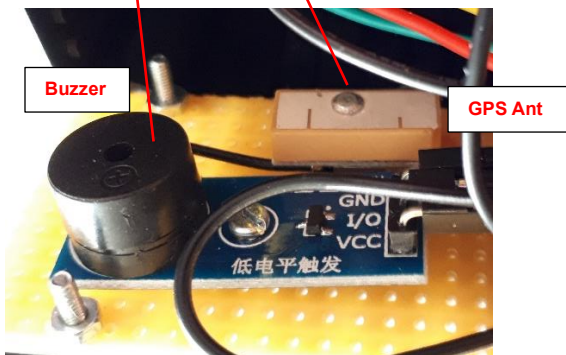
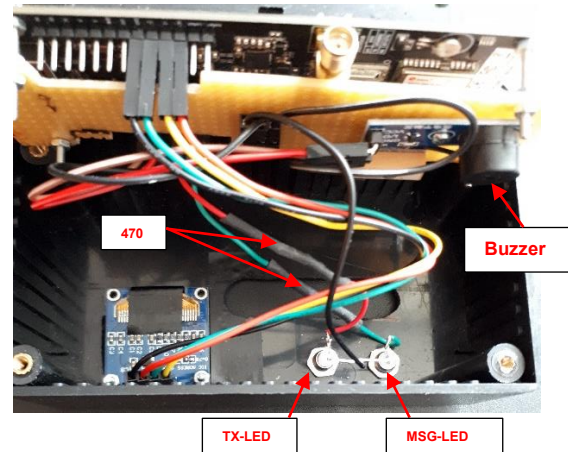


Rückseite

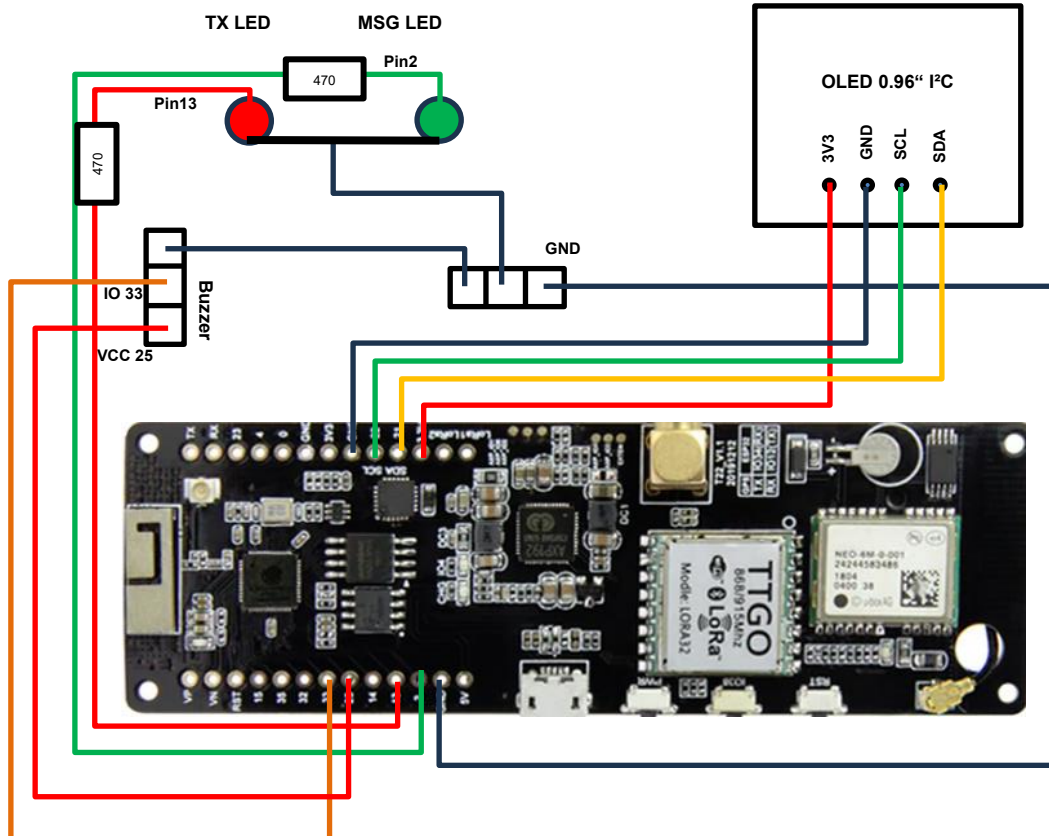
Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU



Eingebaut

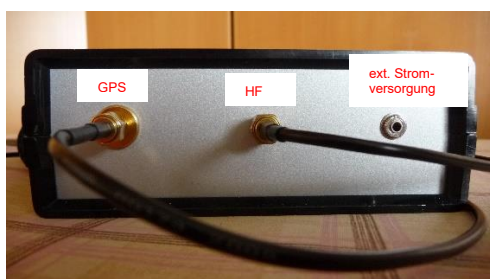
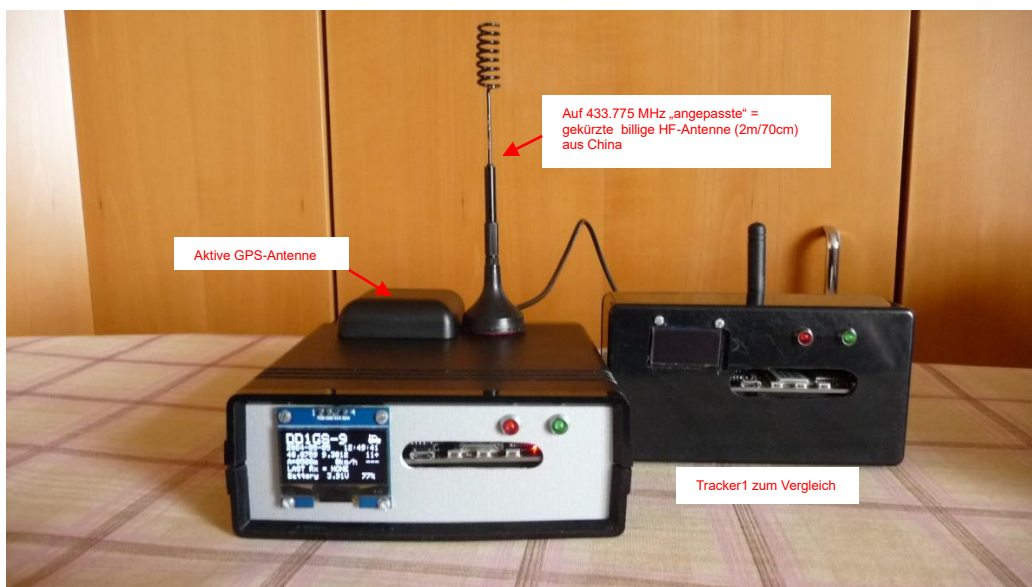
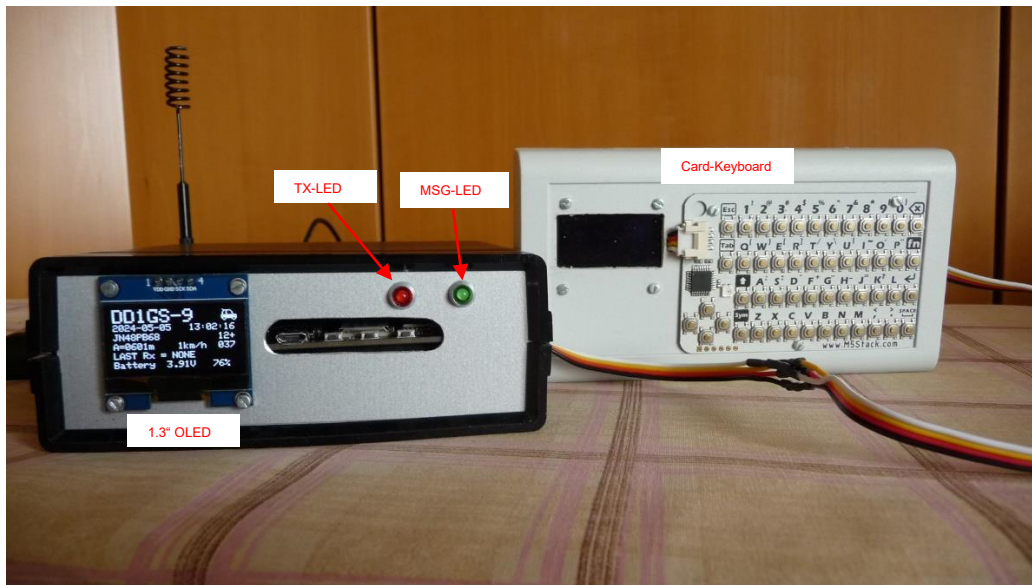


„Schaltplan“:

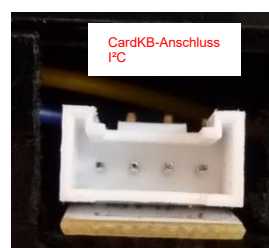


Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Tracker 2 mit Buzzer, LEDs, CardKB



Rückseite

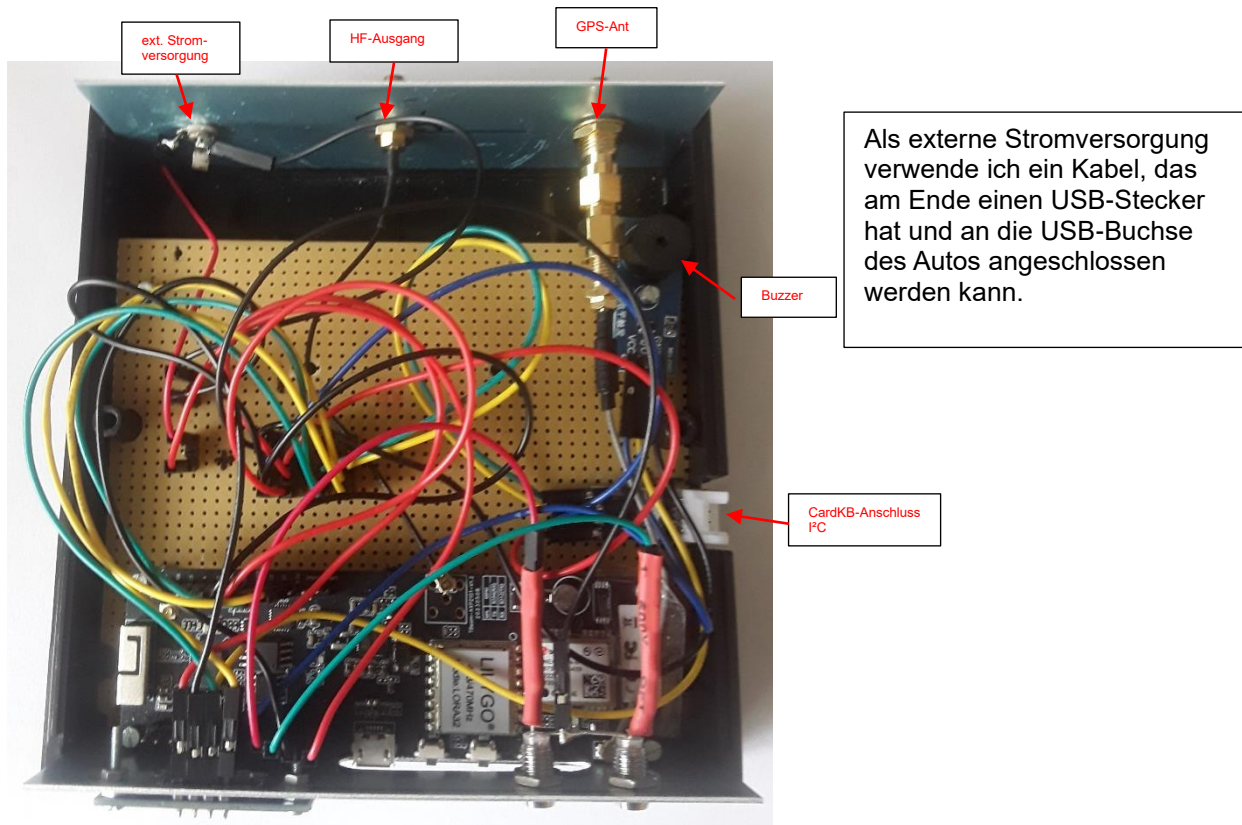


seitlich

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 10 von 12

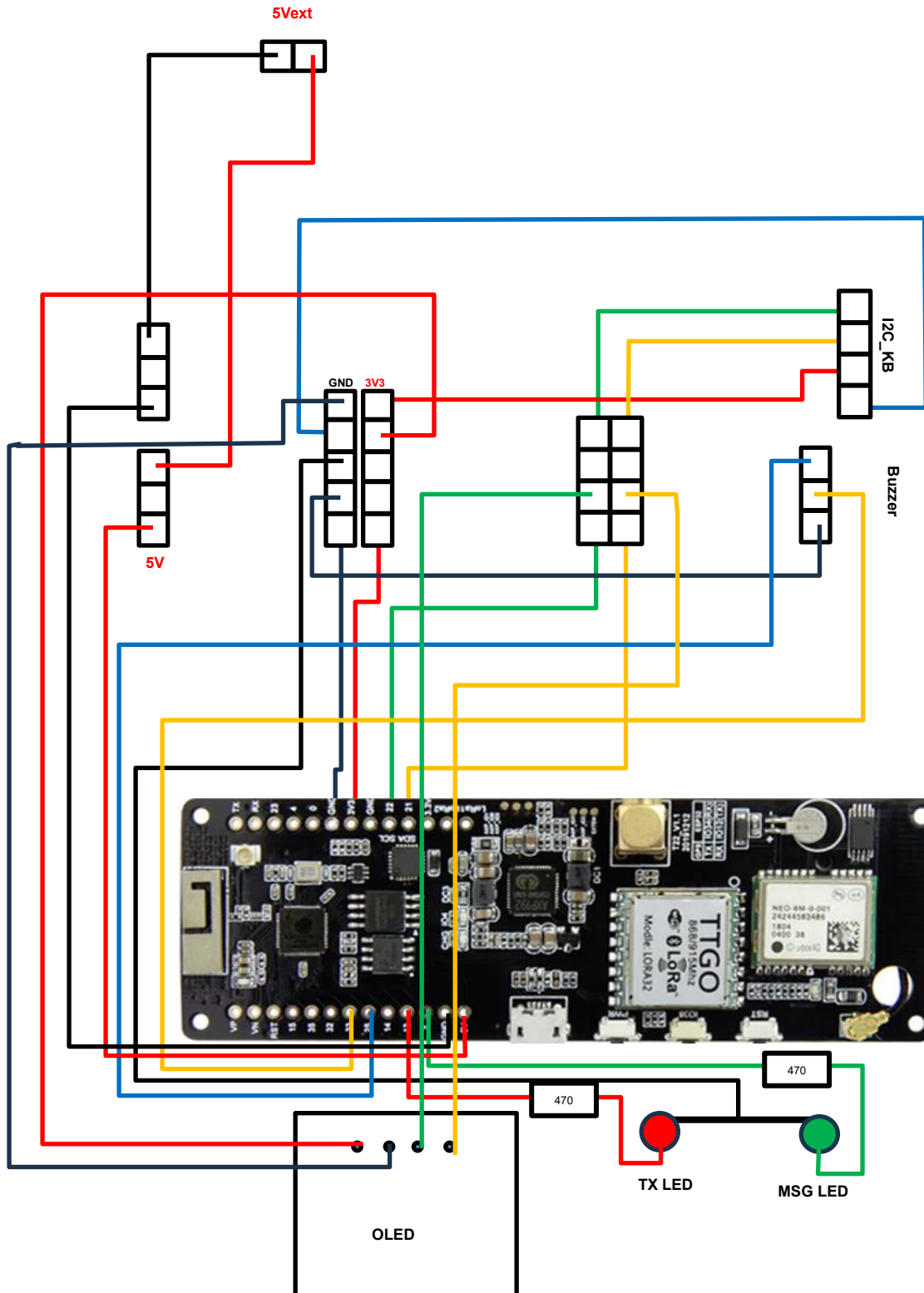
Messungen der ursprünglich zusätzlich eingebauten PA mit einem TinySA Ultra haben gezeigt, dass diese im wahrsten Sinne des Wortes „billige“ PA bei der 2. Harmonischen lediglich einen Wert von ca. -16dBc erreichte. Deshalb habe ich diese PA wieder ausgebaut. Diverse Fahrten (Fahrrad) mit dem Tracker1 haben mir gezeigt, dass die Leistung, die das Board ohne PA liefert, völlig ausreichend ist. Für eine gute Streckenaufzeichnung ist das Vorhandensein von naheliegenden iGates viel entscheidender. Auch eine gegenüber der originalen GPS-Antenne etwas größere trägt auch dazu bei. Wer mehr Geld investieren will, kann es ja mal mit einer „besseren“ PA versuchen. Ich glaube aber nicht, dass sich das lohnt.



Die Verkabelung sieht „wild“ aus. hi
Das ist aber durch die Verwendung von Jumper Wires bedingt. Was eben den Vorteil hat, das Ganze völlig auseinandernehmen zu können. Für alle + und – Leitungen und die Leitungen des I²C-Buses wurden dieselben Farben verwendet. Die zusammengehörenden Kabel wurden auf Stiftleisten, die auf die Lochrasterplatine (Streifen) gelötet sind, aufgesteckt. HF-Leitungen sind Koaxkabel und SMA-Steckverbinder.

Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

„Schaltplan“



Hardware-Erweiterungen für Tracker und iGate mit Firmware von CA2RXU

Seite 12 von 12

Selten passiert es, dass das GPS-Modul auf dem Tracker-Board nicht mehr richtig funktioniert. Dafür gibt es folgendes Programm, das unter VSC läuft:

[https://github.com/richonguzman/TTGO T BEAM GPS RESET](https://github.com/richonguzman/TTGO_T_BEAM_GPS_RESET)

Zum Abschluss der Dokumentation noch eine sehr schöne Zusammenstellung von **Klaus DL1GKR** zu den Menü-Befehlen über die Drucktasten des Boards. Mit dem Card-Keyboard lässt sich das natürlich auch bedienen. Nachrichten sendet man aber am einfachsten mit einem Smartphone, das per Bluetooth mit APRSDroid und dem Tracker als TNC verbunden ist. CA2RXU beschreibt das in seinen Anleitungen:

https://github.com/richonguzman/LoRa_APRS_Tracker/wiki/06.-Bluetooth-Connection-%23-Conexi%C3%B3n-Bluetooth

Hier die Zusammenstellung der Menü-Befehle von Klaus DL1GKR:

TTGO T-Beam v1.2 --- Tastenbedienung und LEDs

Tasten:

linke Taste (PWR)	zum Einschalten des Moduls	1x kurz drücken.
linke Taste (PWR)	zum Ausschalten des Moduls	7s drücken.
mittlere Taste (IO38)	zum Ausschalten des Moduls	3x kurz drücken.
mittlere Taste (IO38)	um Display zu reaktivieren	1x kurz drücken.
mittlere Taste (IO38)	um APRS-Symbol umzuschalten	1x lang drücken.
mittlere Taste (IO38)	um Bake manuell zu senden	1x kurz drücken.
mittlere Taste (IO38)	um ins Menü zu kommen	2x kurz drücken.
mittlere Taste (IO38)	um im Menü zu navigieren:	
	- weiter	1x kurz
	- zurück	2x kurz
	- bestätigen	1x lang
rechte Taste (RST)	um Modul neu zu starten	1x kurz drücken.

LEDs:

rote LED (unter Display)	leuchtet rot, wenn Modul eingeschaltet ist.
blaue LED (unter Display)	leuchtet blau, wenn Akku geladen wird.
rote LED (unterhalb GPS)	blinkt rot, wenn gültige GPS-Position vorliegt.

Wichtig:

- Display schaltet nach einstellbarem Timeout (z.B. 1 min) aus.
- Display-Reaktivierung mit mittlerer Taste.
- im Menü automatische Rückkehr ins Hauptdisplay nach 30s oder mittlere Taste 2x kurz drücken.
- Display-Helligkeit (MIN/MAX) nur über Menü umstellbar:
Configuration > Display > Brightness > MAX
z.B. mittlere Taste: 2k 1k lang 1k 1k lang 1k lang
- Akku wird auch bei ausgeschaltetem Modul geladen, doch ohne LED-Kontrolle!