

LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

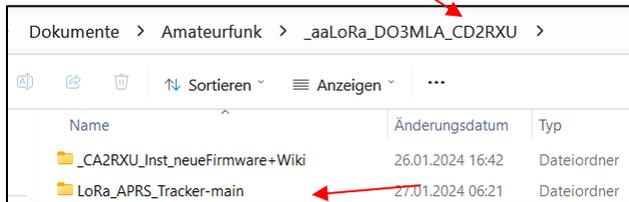
Seite 1 von 8

Neue Firmware-Zip (https://github.com/richonguzman/LoRa_APRS_Tracker) aus dem Download-Ordner

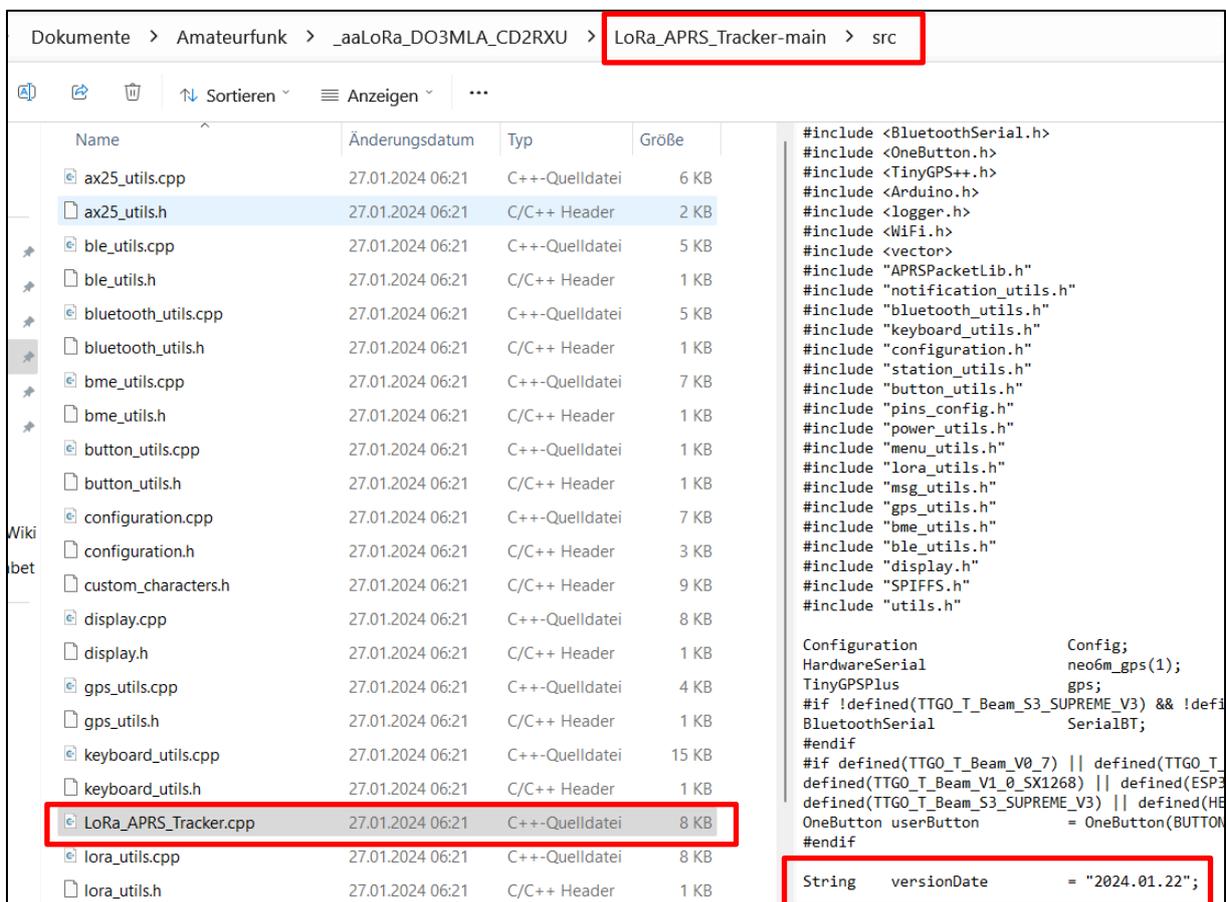


in den Ordner*) entpacken

*) bei mir



Im entpackten Ordner im Ordner **LoRa_APRS_Tracker-main/src** nach dem Versionsdatum suchen (Datei **LoRa_APRS_Tracker.cpp**) und den Namen des Ordners entsprechend umbenennen/ergänzen.

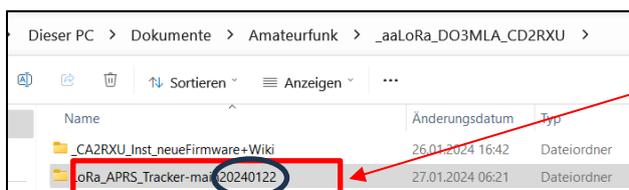


Name	Änderungsdatum	Typ	Größe
ax25_utils.cpp	27.01.2024 06:21	C++-Quelldatei	6 KB
ax25_utils.h	27.01.2024 06:21	C/C++ Header	2 KB
ble_utils.cpp	27.01.2024 06:21	C++-Quelldatei	5 KB
ble_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
bluetooth_utils.cpp	27.01.2024 06:21	C++-Quelldatei	5 KB
bluetooth_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
bme_utils.cpp	27.01.2024 06:21	C++-Quelldatei	7 KB
bme_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
button_utils.cpp	27.01.2024 06:21	C++-Quelldatei	1 KB
button_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
configuration.cpp	27.01.2024 06:21	C++-Quelldatei	7 KB
configuration.h	27.01.2024 06:21	C/C++ Header	3 KB
custom_characters.h	27.01.2024 06:21	C/C++ Header	9 KB
display.cpp	27.01.2024 06:21	C++-Quelldatei	8 KB
display.h	27.01.2024 06:21	C/C++ Header	1 KB
gps_utils.cpp	27.01.2024 06:21	C++-Quelldatei	4 KB
gps_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
keyboard_utils.cpp	27.01.2024 06:21	C++-Quelldatei	15 KB
keyboard_utils.h	27.01.2024 06:21	C/C++ Header	1 KB
LoRa_APRS_Tracker.cpp	27.01.2024 06:21	C++-Quelldatei	8 KB
lora_utils.cpp	27.01.2024 06:21	C++-Quelldatei	8 KB
lora_utils.h	27.01.2024 06:21	C/C++ Header	1 KB

```
#include <BluetoothSerial.h>
#include <OneButton.h>
#include <TinyGPS++.h>
#include <Arduino.h>
#include <logger.h>
#include <WiFi.h>
#include <vector>
#include "APRSPacketLib.h"
#include "notification_utils.h"
#include "bluetooth_utils.h"
#include "keyboard_utils.h"
#include "configuration.h"
#include "station_utils.h"
#include "button_utils.h"
#include "pins_config.h"
#include "power_utils.h"
#include "menu_utils.h"
#include "lora_utils.h"
#include "msg_utils.h"
#include "gps_utils.h"
#include "bme_utils.h"
#include "ble_utils.h"
#include "display.h"
#include "SPIFFS.h"
#include "utils.h"

Configuration          Config;
HardwareSerial         neo6m_gps(1);
TinyGPSPlus           gps;
#if !defined(TTGO_T_Beam_S3_SUPREME_V3) && !defi
BluetoothSerial       SerialBT;
#endif
#if defined(TTGO_T_Beam_V0_7) || defined(TTGO_T
defined(TTGO_T_Beam_V1_0_SX1268) || defined(ESP3
defined(TTGO_T_Beam_S3_SUPREME_V3) || defined(HE
OneButton userButton = OneButton(BUTTON
#endif

String versionDate = "2024.01.22";
```



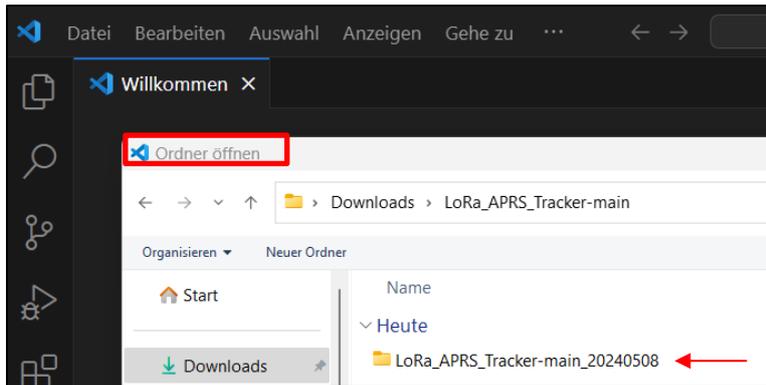
LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

Seite 2 von 8

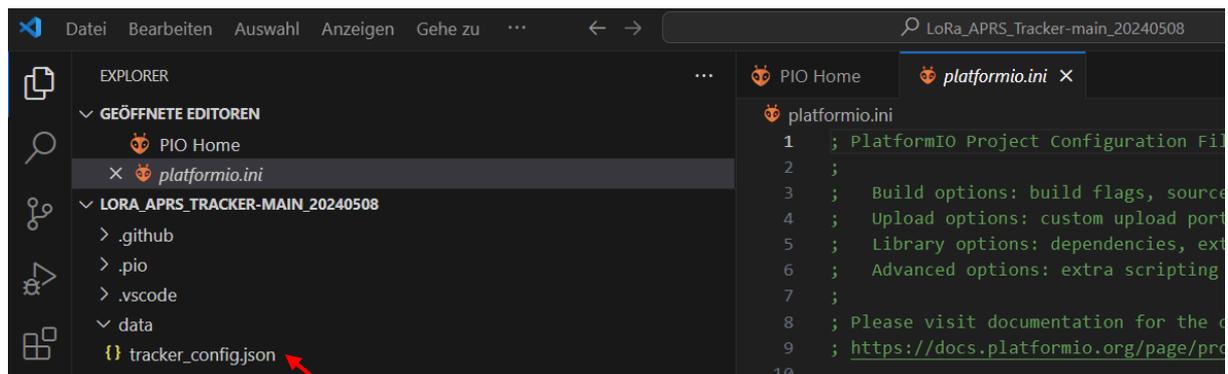
Ich denke, Ricardo (CA2RXU) wird künftig auch für die Tracker-Firmware einen einfacheren Weg für die Installation/Konfiguration statt den Weg über VSC entwickeln, wie es bereits bei der iGate-Firmware geschehen ist.

Hier meine Zusammenfassung der Vorgehensweise mit Hilfe von VSC. Die englische Beschreibung findet sich auf der oben angegebenen GitHub-Seite.

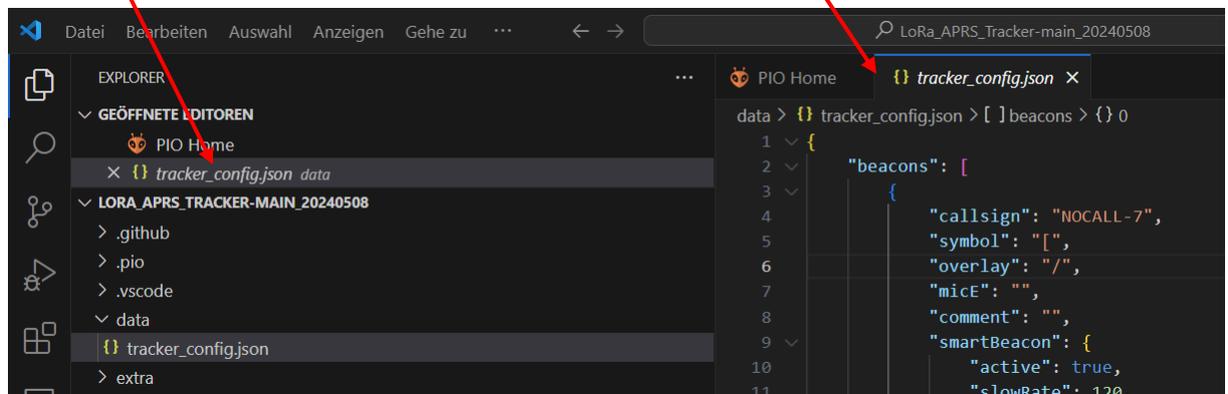
Ordner mit neuer, entpackter Firmware (s.o.) in VSC öffnen.



Es erscheint:



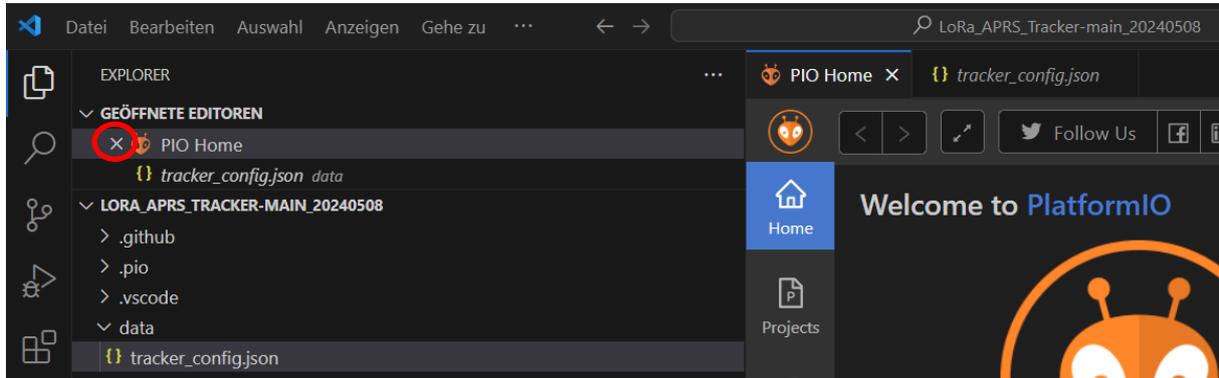
Auf **tracker_config.json** klicken. Statt **platformio.ini** erscheint bei den geöffneten Editoren **tracker_config.json data**. Sie ist auch auf der rechten Bildschirm-Hälfte zu sehen und enthält noch viele Werte, die noch nicht angepasst sind, z.B. „callsign“: „NOCALL-7“.



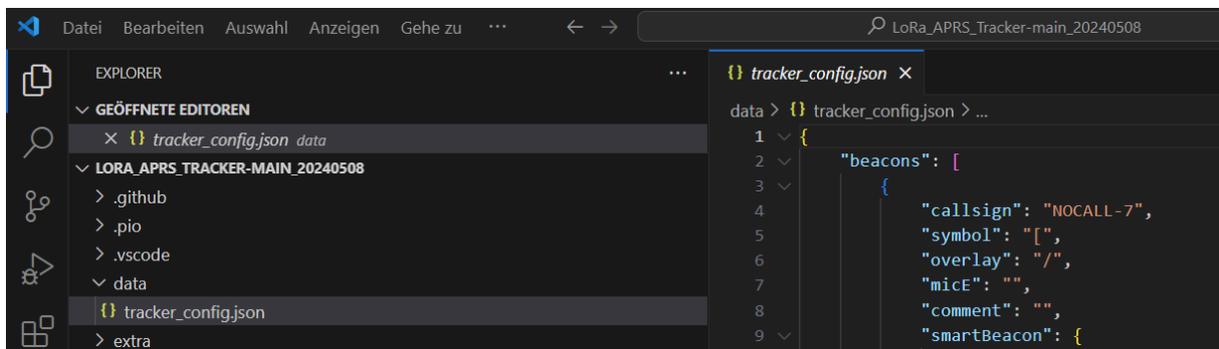
LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

Seite 3 von 8

Jetzt klickt man auf das **x** bei PIO Home, um diese Datei aus den Editoren zu entfernen, damit sie beim weiteren Vorgehen nicht stört.



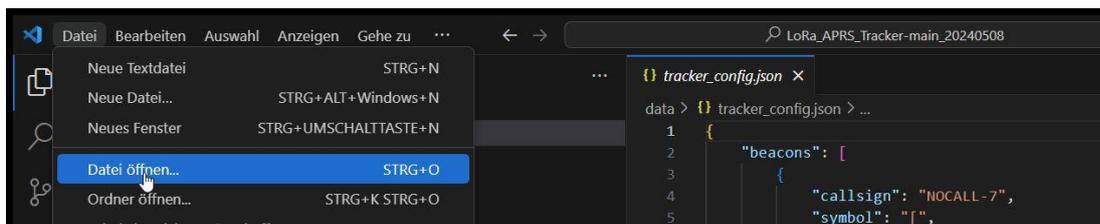
Es ist nur noch der Editor mit der **tracker_config.json data** geöffnet.



Es gibt nun zwei Wege, die Daten einer früheren Version auf diese neue **tracker_config.json** zu übertragen:

Man hat vorher die alte Version markiert, kopiert und diese Kopie ausgedruckt oder man öffnet neben der neuen Version zusätzlich die alte in den Editoren. Nachfolgend die Beschreibung der zweiten Möglichkeit:

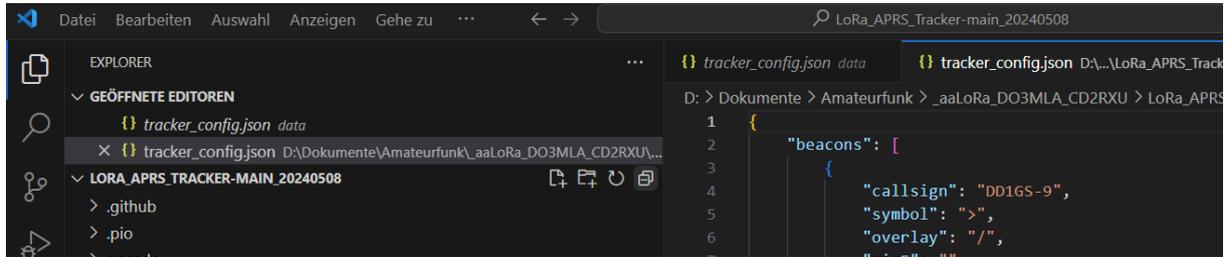
Über **Datei/Datei öffnen** wird die Vorgängerkonfigurationsdatei den geöffneten Editoren hinzugefügt.



LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

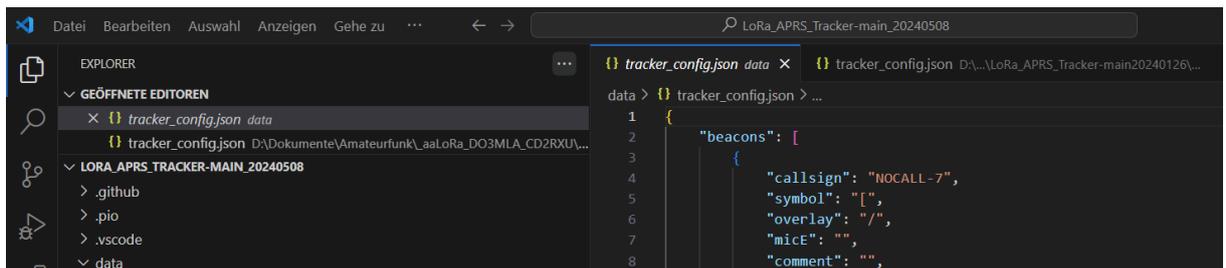
Seite 4 von 8

Diese hinzugefügte Datei erscheint jetzt mit Pfadangabe bei den Editoren. Rechts sieht man den Inhalt mit dem früher ausgefüllten Call „DD1GS-9“.



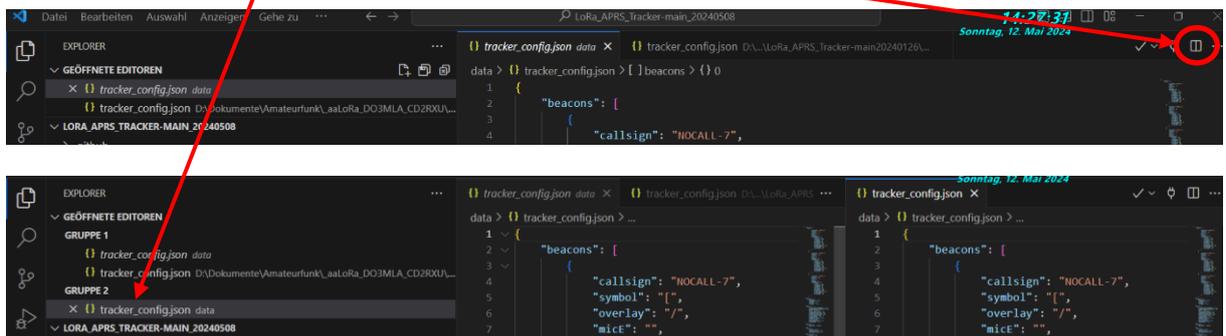
```
1 {
2   "beacons": [
3     {
4       "callsign": "DD1GS-9",
5       "symbol": ">",
6       "overlay": "/",
7       "micE": ""
8     }
9   ]
10 }
```

Klickt man auf **tracker_config.json data** ist man wieder in der neuen Firmware.



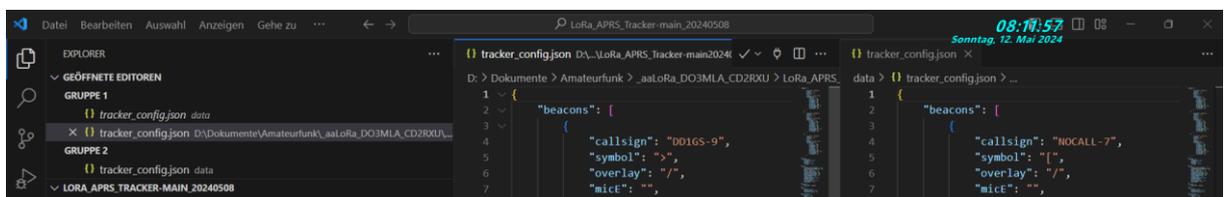
```
1 {
2   "beacons": [
3     {
4       "callsign": "NOCALL-7",
5       "symbol": "[",
6       "overlay": "/",
7       "micE": "",
8       "comment": ""
9     }
10  ]
11 }
```

Klickt man nun auf das kleine Symbol „Editor rechts teilen“, entsteht bei den Editoren eine Gruppe 2 mit **tracker_config.json data**



```
1 {
2   "beacons": [
3     {
4       "callsign": "NOCALL-7",
5       "symbol": "[",
6       "overlay": "/",
7       "micE": ""
8     }
9   ]
10 }
```

Mit Klick auf die Vorgängerdatei ändert sich der Inhalt der 1. Spalte auf der rechten Bildschirmhälfte. Die 1. Spalte enthält die Werte der Vorgängerversion. Diese werden nun sukzessiv in die 2. Spalte eingetragen. Man achte genau darauf, dass man keine Zeile vergisst. Wichtig sind z.B. „symbol“ und die nachfolgenden Werte für smart beaconing („smartBeacon“).



```
1 {
2   "beacons": [
3     {
4       "callsign": "DD1GS-9",
5       "symbol": ">",
6       "overlay": "/",
7       "micE": ""
8     }
9   ]
10 }
```

Ich habe folgende 3 Konfigurationen für die unterschiedliche Verwendung des Trackers (PKW, Fahrrad, Fußgänger) bei mir eingetragen. Die Werte für smart beaconing sind teilweise experimentell ermittelt, bzw. habe ich von **Wolfgang DL1GWW** übernommen. Natürlich kann man sie durch weiteres Ausprobieren eventuell noch optimieren bzw. den eigenen individuellen Wünsche anpassen.

LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

Hier meine persönlichen Eintragungen als Anregung:

```
{
  "beacons": [
    {
      "callsign": "DD1GS-9",
      "symbol": ">",
      "overlay": "/",
      "micE": "",
      "comment": "Willi P29",
      "smartBeacon": {
        "active": true,
        "slowRate": 120,
        "slowSpeed": 10,
        "fastRate": 60,
        "fastSpeed": 70,
        "minTxDist": 100,
        "minDeltaBeacon": 8,
        "turnMinDeg": 10,
        "turnSlope": 80
      }
    },
    {
      "callsign": "DD1GS-8",
      "symbol": "b",
      "overlay": "/",
      "micE": "",
      "comment": "Willi P29",
      "smartBeacon": {
        "active": true,
        "slowRate": 120,
        "slowSpeed": 5,
        "fastRate": 20,
        "fastSpeed": 25,
        "minTxDist": 50,
        "minDeltaBeacon": 12,
        "turnMinDeg": 12,
        "turnSlope": 60
      }
    },
    {
      "callsign": "DD1GS-7",
      "symbol": "[",
      "overlay": "/",
      "micE": "",
      "comment": "Willi P29",
      "smartBeacon": {
        "active": true,
        "slowRate": 120,
        "slowSpeed": 2,
        "fastRate": 30,
        "fastSpeed": 5,
        "minTxDist": 20,
        "minDeltaBeacon": 20,
        "turnMinDeg": 3,
        "turnSlope": 40
      }
    }
  ]
}
```

LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

```
"display": {
  "showSymbol": true,
  "ecoMode": false,
  "timeout": 4,
  "turn180" : false
},
"other": {
  "simplifiedTrackerMode": false,
  "sendCommentAfterXBeacons": 10,
  "path": "WIDE1-1",
  "nonSmartBeaconRate": 15,
  "rememberStationTime": 30,
  "maxDistanceToTracker": 30,
  "standingUpdateTime": 15,
  "sendAltitude": true,
  "sendBatteryInfo": false,
  "bluetoothType": 1,
  "bluetoothActive": true,
  "disableGPS": false
},
"winlink": {
  "password": "ABCDEF"
},
"bme": {
  "active": false,
  "sendTelemetry": false,
  "heightCorrection": 0
},
"notification": {
  "ledTx": true,
  "ledTxPin": 13,
  "ledMessage": true,
  "ledMessagePin": 2,
  "ledFlashlight": false,
  "ledFlashlightPin": 14,
  "buzzerActive": true,
  "buzzerPinTone": 33,
  "buzzerPinVcc": 25,
  "bootUpBeep": true,
  "txBeep": true,
  "messageRxBeep": true,
  "stationBeep": true,
  "lowBatteryBeep": true
},
"lora": {
  "frequency": 433775000,
  "spreadingFactor": 12,
  "signalBandwidth": 125000,
  "codingRate4": 5,
  "power": 20
},
"pttTrigger": {
  "active": false,
  "io_pin": 4,
  "preDelay": 0,
  "postDelay": 0,
  "reverse": false
}
}
```

Da ich LEDs für TX und MSG und einen Buzzer eingebaut habe, stehen hier viele Werte auf „true“.

In dieser älteren Version vom 26.01.2024 steht hier nur die Frequenz 433775000. In der neueren Version vom 08.05.2024 stehen hier 2 zusätzliche für andere Länder/Regionen. Die richtige wird automatisch ausgewählt. **Bitte die anderen nicht löschen! Im Display erscheint LoRa[EU].**

LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

Seite 7 von 8

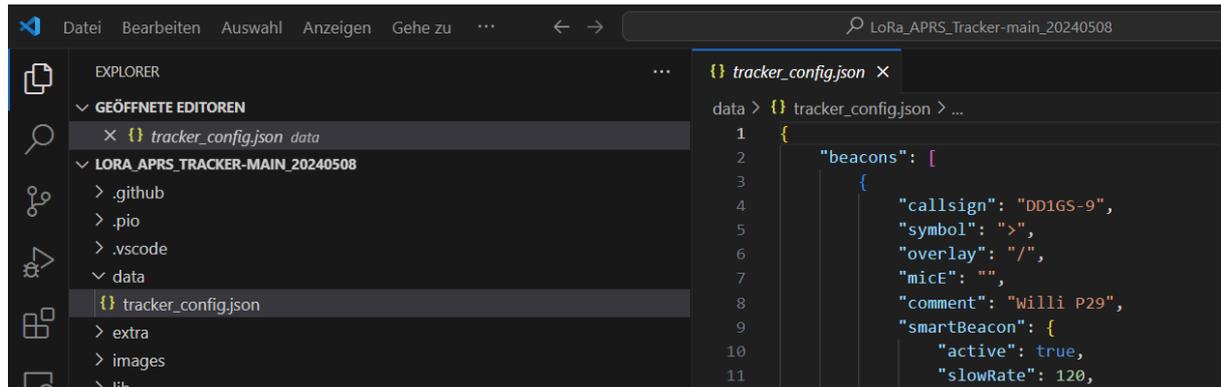
Im Editor auf *tracker_config.json* klicken und speichern.

Ordner schließen. Damit ist in der neuen Firmware die Vorläufer-Konfiguration übertragen.

Wie diese konfigurierte Firmware mit VSC auf das Board übertragen wird, steht in den Anleitungen (WIKI) von CA2RXU bzw. hier in Kurzform:

Fertige Konfiguration flashen:

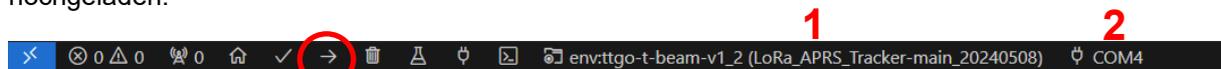
Wenn die Konfiguration vollständig ist,



```
data > {} tracker_config.json > ...
1  {
2    "beacons": [
3      {
4        "callsign": "DD1GS-9",
5        "symbol": ">",
6        "overlay": "/",
7        "micE": "",
8        "comment": "Willi P29",
9        "smartBeacon": {
10         "active": true,
11         "slowRate": 120,
```

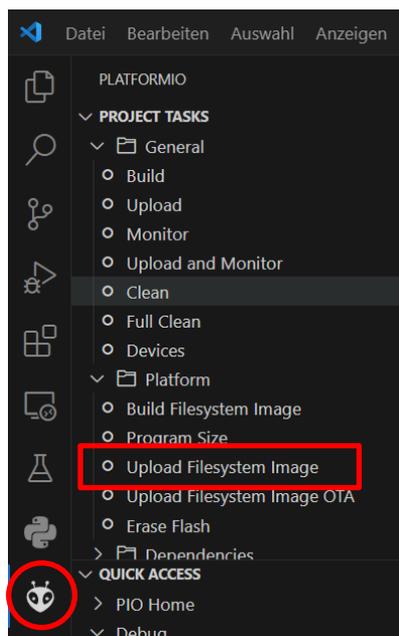
erfolgt das Hochladen der Firmware und des Filesystems:

Klicke in der unteren Zeile auf den Pfeil nach rechts. Die Dateien werden kompiliert und auf das Board hochgeladen.



Ist der Vorgang erfolgreich abgeschlossen, kann das Filesystem hochgeladen werden:

Klicke dazu zuerst auf den „Ameisenkopf“ und wähle dann „Upload Filesystem Image“:



Sollte das nicht auf Anhieb funktionieren, kontrolliere erst das ausgewählte Board **1** und den COM-Port **2**. Wenn die Meldung kommt, dass der COM-Port nicht erkannt wird, hilft es, das USB-Kabel – eventuell mehrmals – aus- und wieder einzustecken.

LoRa-Tracker-Firmware-Installation und -Upgrade (mit Hilfe von VSC)

Seite 8 von 8

1.3"-Display [von mir bei einem 2. Tracker verwendet – siehe auch „Hardware-Erweiterungen“]

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.

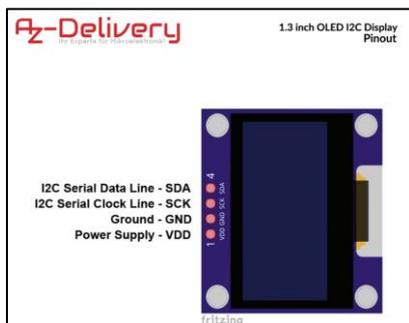
Erstelle eine Kopie dieser konfigurierten neuen Version, nimm die OLED13-Änderungen in der Kopie vor und versieh den Ordnernamen mit „OLED13“.

LoRa_APRS_Tracker-main_v20240122 **OLED13** 22.01.2024 22:30 Dateiordner

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.

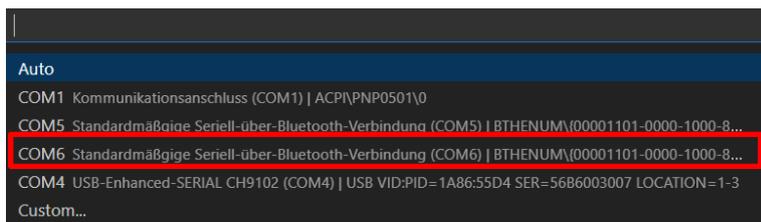


Damit hat man eine **spezielle** Version für das OLED 1.3" (mit den Anschlüssen: SDA, SCK, GND, VDD).

Um zu testen, ob alles korrekt ist, kann vor dem Flashen ein Build (Häkchen) 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";
```