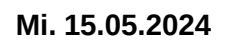


- Workshop 1 von 2
- Das I-Gate



LoRa APRS

- LoRa-APRS, eine kurze Einführung
- Visual Studio Code (VSC) & Platform.IO
- Installation der Software
- APRS Passcode
- Download und Programmierung der I-Gate Firmware
- Die Parameter
- Komponenten

LoRa-APRS, eine kurze Einführung

LoRa APRS ist eine Betriebsart die es ermöglicht Informationspakete mit geringer Leistung über größere Entfernungen zu übertragen.

Die verwendeten TRX'e (Tracker) haben in der Regel eine Ausgangsleistung von ca. 100 mW (20dBm)

Es können fertige Geräte für geringe Kosten verwendet werden (Chinaware)

Die benötigte Software unterliegt ständigen Weiterentwicklungen ist „**open Source**“ und bei GitHub erhältlich. (Youtube / Manuel Lausmann)

Es ist mit einfachen Mitteln möglich ein Internet Gateway (I-Gate) aufzubauen welches die Daten ins Internet überträgt

LoRa-APRS, eine kurze Einführung

Frequenzen für APRS Betrieb:

Bereich	Frequenz	Betriebsart	Region
70-Zentimeter-Band	432,500 MHz	FM mit 1200 Baud AFSK	Deutschland, Luxemburg, Lothringen, Tests in Wien ^[8]
	433,800 MHz	FM mit 1200 Baud AFSK	Testbetrieb seit September 2012 im Innviertel
	430,5125 MHz	FM mit 1200 Baud AFSK	Testbetrieb in Holland
	433,775 MHz	LoRa mit 128 kbps	Testbetrieb in OE und DL

LoRa-APRS, eine kurze Einführung

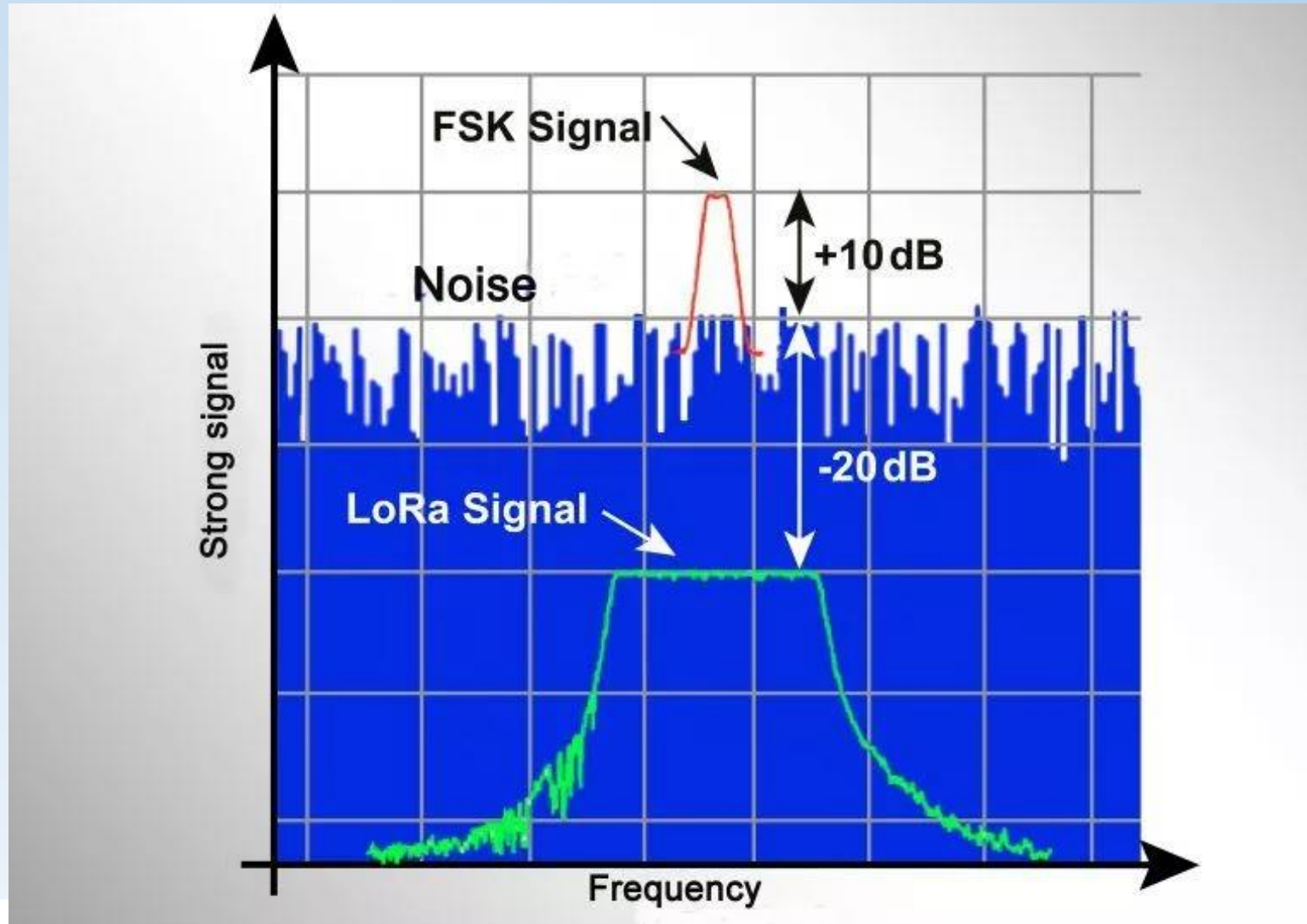
Was ist das besondere an LoRa?

Es wird ein Frequenzspreizverfahren als Modulationstechnik verwendet, das sogenannte CSS (Chirp Spread Spectrum).

Chirp-Impulse werden als Daten gesendet, um eine größere Funkreichweite als bei herkömmliche Modulationstypen wie FSK zu erreichen.

Der LoRa-Empfänger kann ein nützliches Signal bis zu 20 dB unter dem Geräuschpegel erfolgreich empfangen und decodieren.

LoRa-APRS, eine kurze Einführung



LoRa-APRS, eine kurze Einführung

Welche Anwendungen können u. a. damit realisiert werden?

1. Tracker (KFZ, Fahrrad, Wandern etc.)
2. Übertragung von Messwerten (Wasserstände etc.)
3. Einsatz als Wetterstationen (DG1EAD-15)
- 4. Internet Gateway (I-Gates) !**

LoRa-APRS, eine kurze Einführung



Visual Studio Code (VSC) & PlatformIO

Code editing.
Redefined.

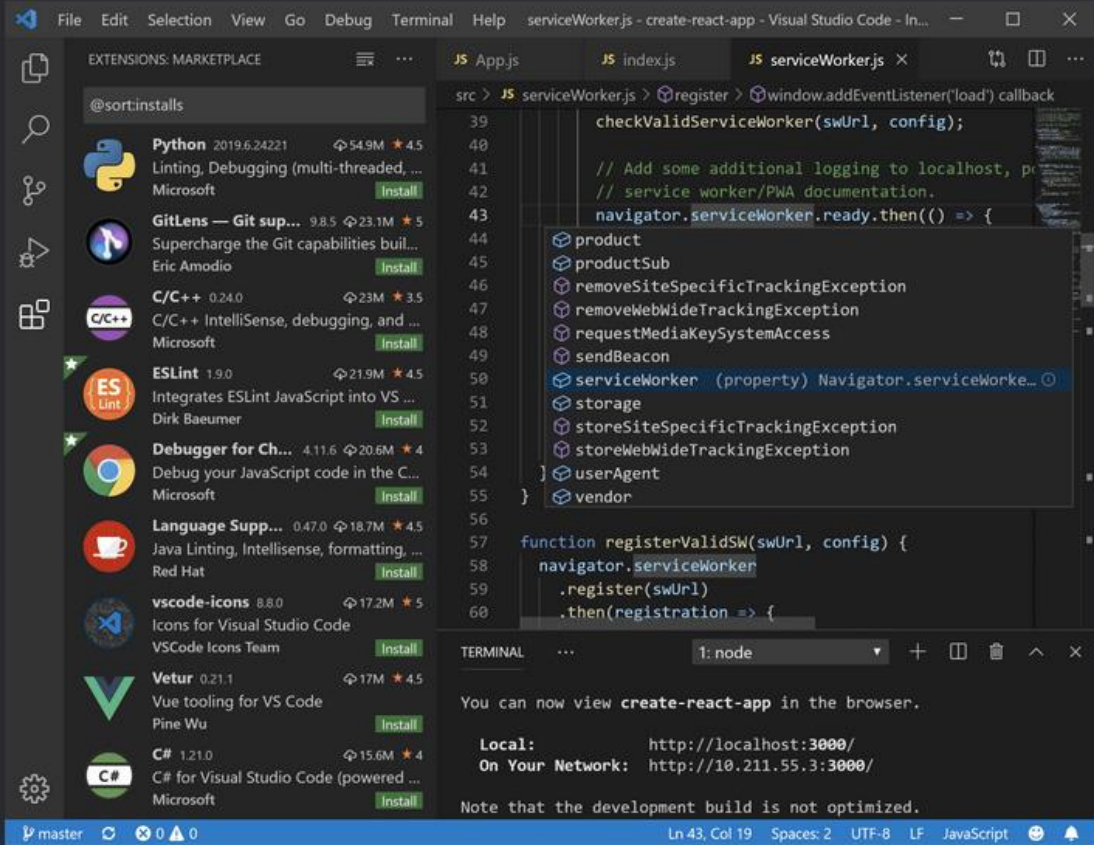
Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its
[license and privacy statement.](#)





The screenshot displays the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' sidebar lists various extensions like Python, GitLens, C/C++, ESLint, and others. The main editor area shows a JavaScript file named 'serviceWorker.js' with code for registering a service worker. A dropdown menu is visible over the code, listing properties of the 'navigator.serviceWorker' object. At the bottom, a terminal window shows the output of a command, indicating that the application is running on localhost:3000.

Visual Studio Code (VSC) & PlatformIO

<https://code.visualstudio.com>

The image shows a composite of two parts. On the left is a promotional graphic for Visual Studio Code with the text 'Code editing. Redefined.' and 'Free. Built on open source. Runs everywhere.' Below this is a 'Download for Windows' button with a dropdown arrow, and a table of download links for various operating systems. A red arrow points from the dropdown arrow to the VS Code interface on the right. The interface shows the 'EXTENSIONS: MARKETPLACE' sidebar with a list of installed and available extensions, the main editor area showing a JavaScript file, and a terminal window at the bottom.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build

		Stable	Insiders
macOS	Universal	↓	↓
Windows x64	User Installer	↓	↓
Linux x64	.deb	↓	↓
	.rpm	↓	↓

[Other downloads or open on web](#)

EXTENSIONS: MARKETPLACE

@sortinstalls

- Python 2019.6.24221 54.9M 4.5
Linting, Debugging (multi-threaded, ...
Microsoft [Install](#)
- GitLens — Git sup... 9.8.5 23.1M 5
Supercharge the Git capabilities built...
Eric Amodio [Install](#)
- C/C++ 0.24.0 23M 3.5
C/C++ IntelliSense, debugging, and ...
Microsoft [Install](#)
- ESLint 1.9.0 21.9M 4.5
Integrates ESLint JavaScript into VS ...
Dirk Baumer [Install](#)
- Debugger for Ch... 4.11.6 20.6M 4
Debug your JavaScript code in the C...
Microsoft [Install](#)
- Language Supp... 0.47.0 18.7M 4.5
Java Linting, IntelliSense, formatting, ...
Red Hat [Install](#)
- vscode-icons 8.8.0 17.2M 5
Icons for Visual Studio Code
VSCode Icons Team [Install](#)
- Vetur 0.21.1 17M 4.5
Vue tooling for VS Code
Pine Wu [Install](#)
- C# 1.21.0 15.6M 4
C# for Visual Studio Code (powered ...
Microsoft [Install](#)

JS App.js JS index.js JS serviceWorker.js

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41
42 // Add some additional logging to localhost, p...
43 // service worker/PWA documentation.
44 navigator.serviceWorker.ready.then(() => {
45
46   product
47   productSub
48   removeSiteSpecificTrackingException
49   removeWebWideTrackingException
50   requestMediaKeySystemAccess
51   sendBeacon
52   serviceWorker (property) Navigator.serviceWorke...
53   storage
54   storeSiteSpecificTrackingException
55   storeWebWideTrackingException
56   userAgent
57   vendor
58
59 function registerValidSW(swUrl, config) {
60   navigator.serviceWorker
61     .register(swUrl)
62     .then(registration => {
```

TERMINAL ... 1: node

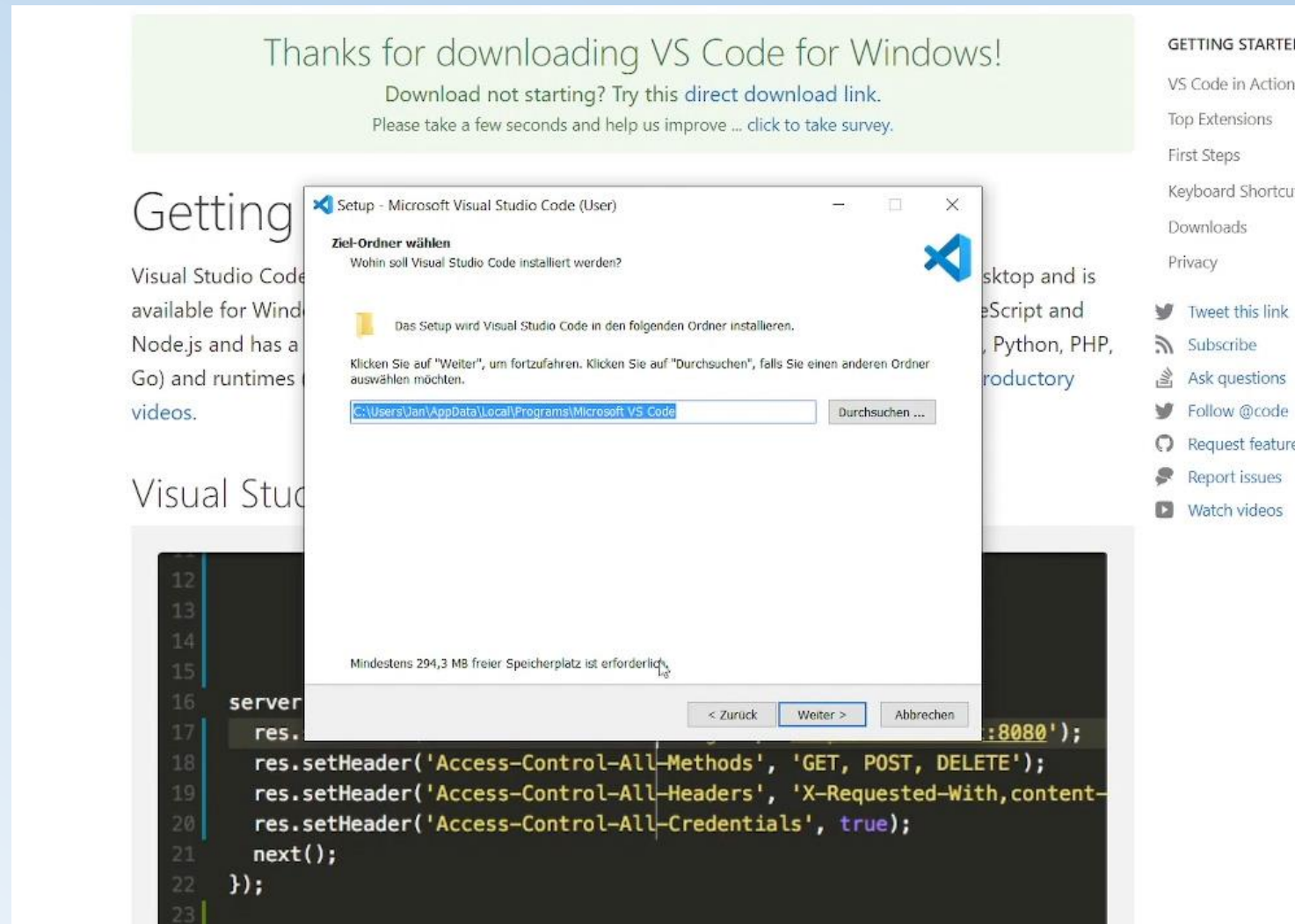
You can now view create-react-app in the browser.

Local: http://localhost:3000/
On Your Network: http://10.211.55.3:3000/

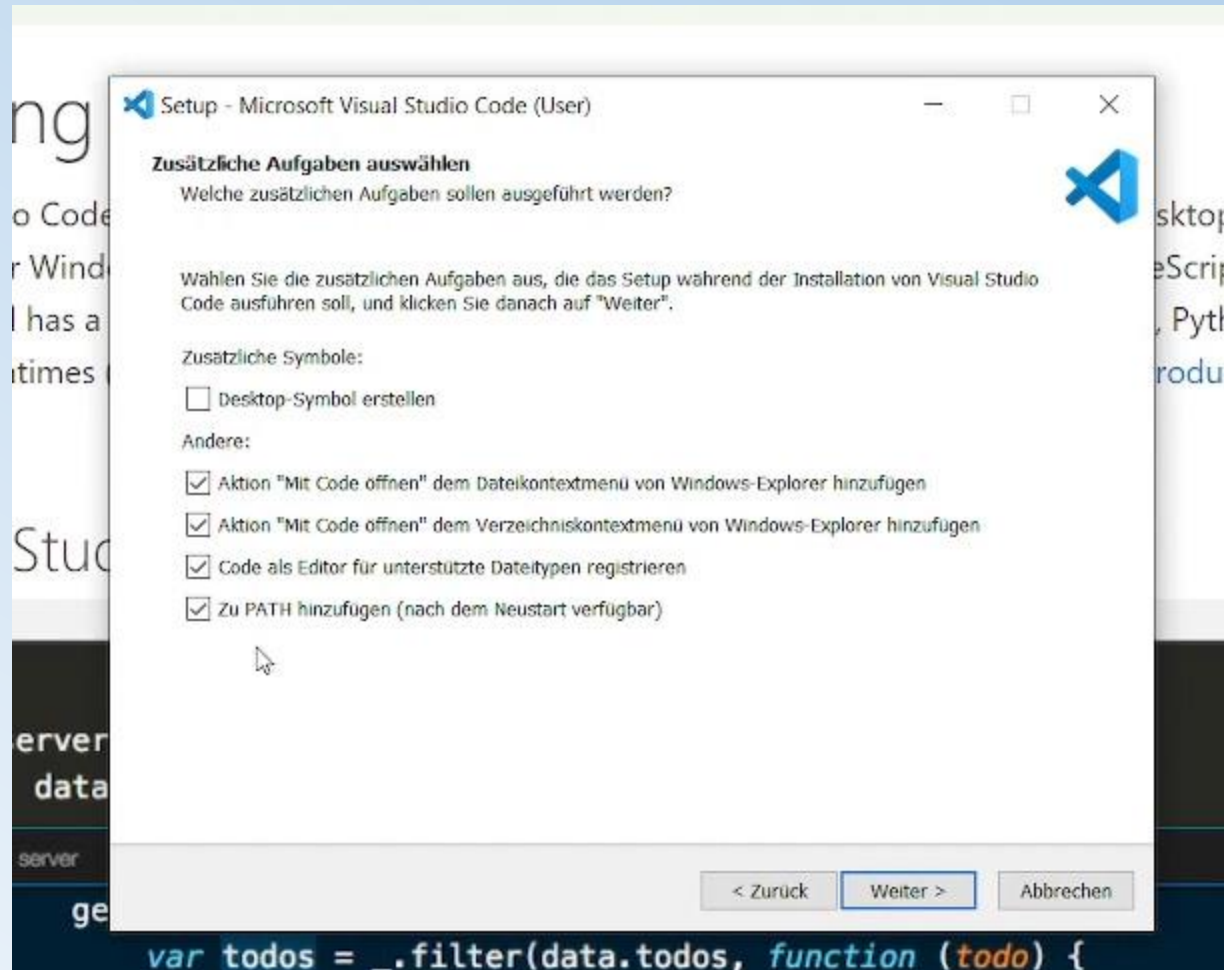
Note that the development build is not optimized.

10

Visual Studio Code (VSC) & PlatformIO

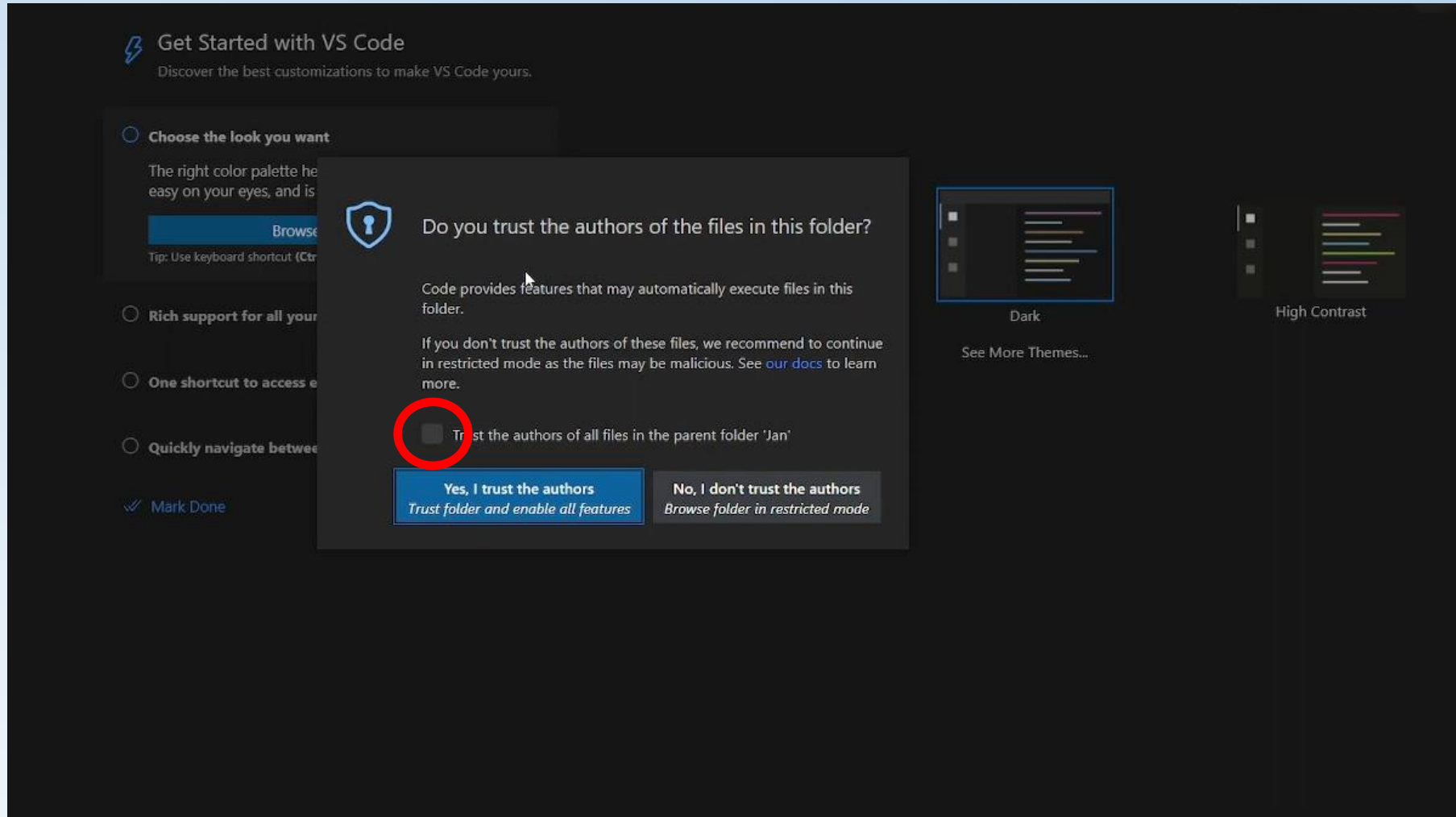


Visual Studio Code (VSC) & PlatformIO

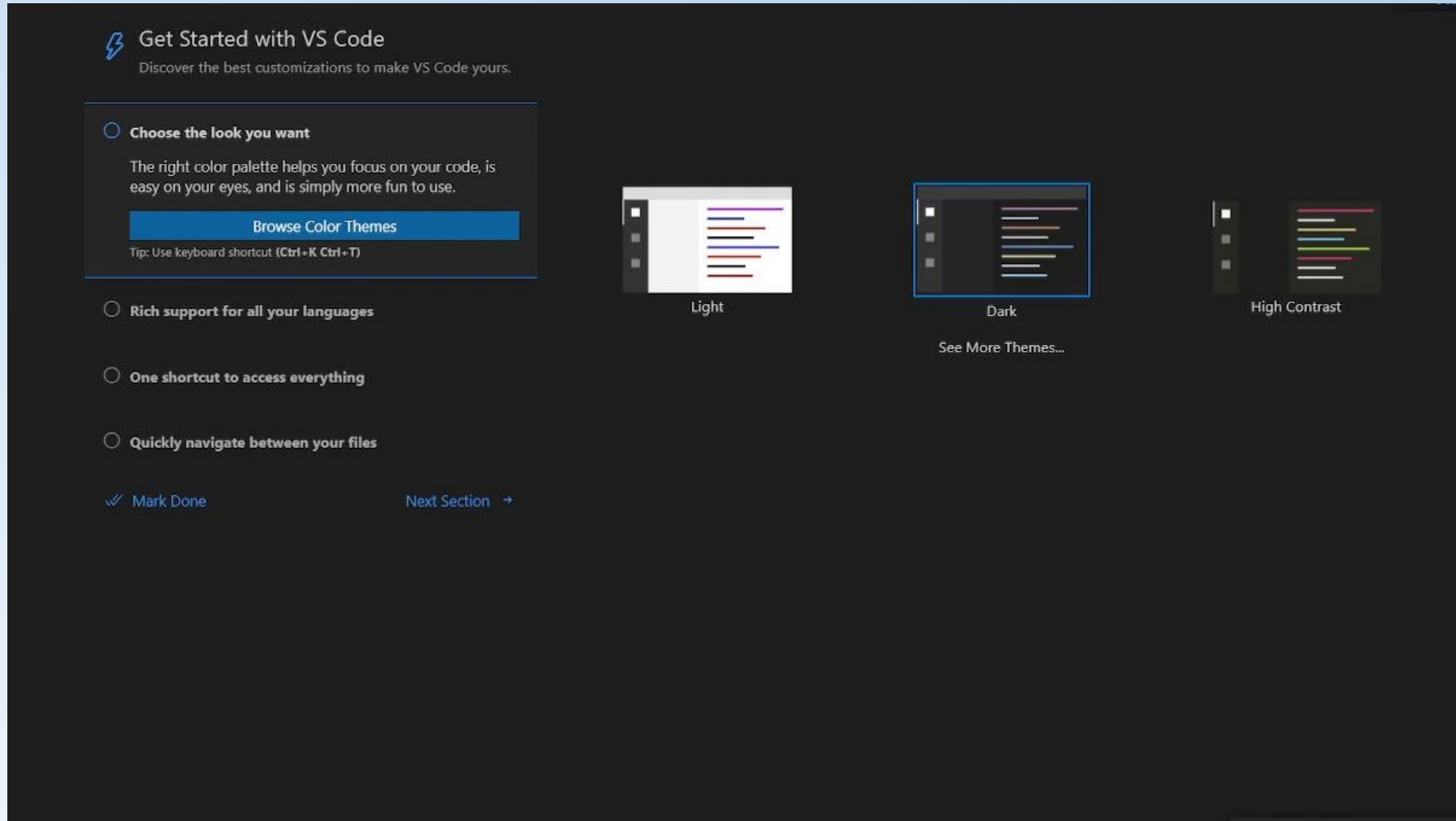


12

Visual Studio Code (VSC) & PlatformIO



Visual Studio Code (VSC) & PlatformIO



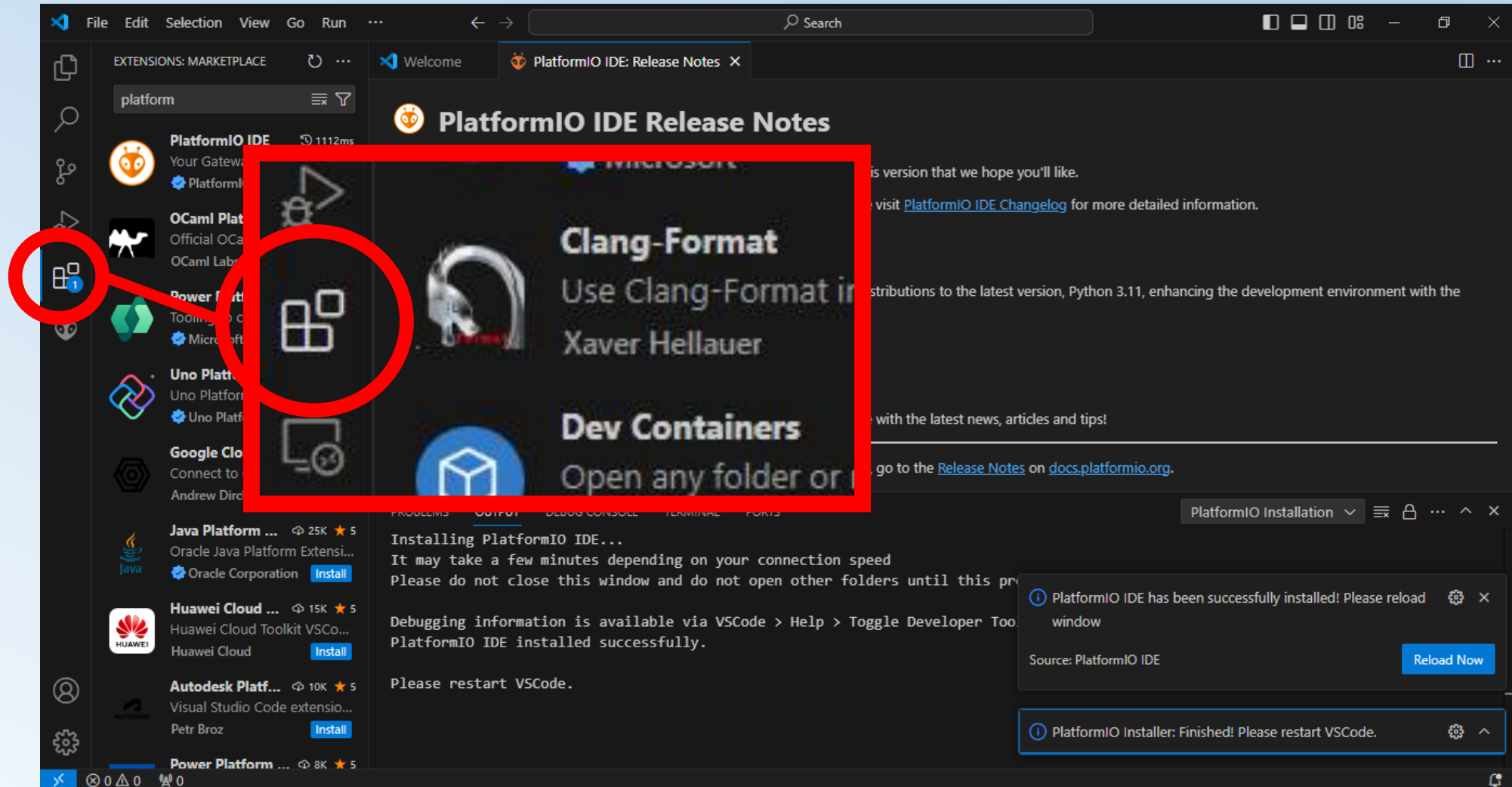
14

Visual Studio Code (VSC) & PlatformIO

The screenshot shows the Visual Studio Code interface during the installation of the PlatformIO IDE extension. The left sidebar displays the 'EXTENSIONS: MARKETPLACE' with a search for 'platform'. The 'PlatformIO IDE' extension is highlighted, showing its details and an 'Install' button. A red arrow points to this extension. The main editor area shows the 'PlatformIO IDE Release Notes' for version 3.3.3 (2024-01-26). The notes include a welcome message, release history, and instructions for staying in touch. The bottom status bar shows the 'PlatformIO Installation' progress, with a message: 'Installing PlatformIO IDE... It may take a few minutes depending on your connection speed. Please do not close this window and do not open other folders until this process is complete.' A notification bubble in the bottom right corner states: 'PlatformIO IDE has been successfully installed! Please reload window. Source: PlatformIO IDE. Reload Now'. Another notification bubble at the bottom states: 'PlatformIO Installer: Finished! Please restart VSCode.'

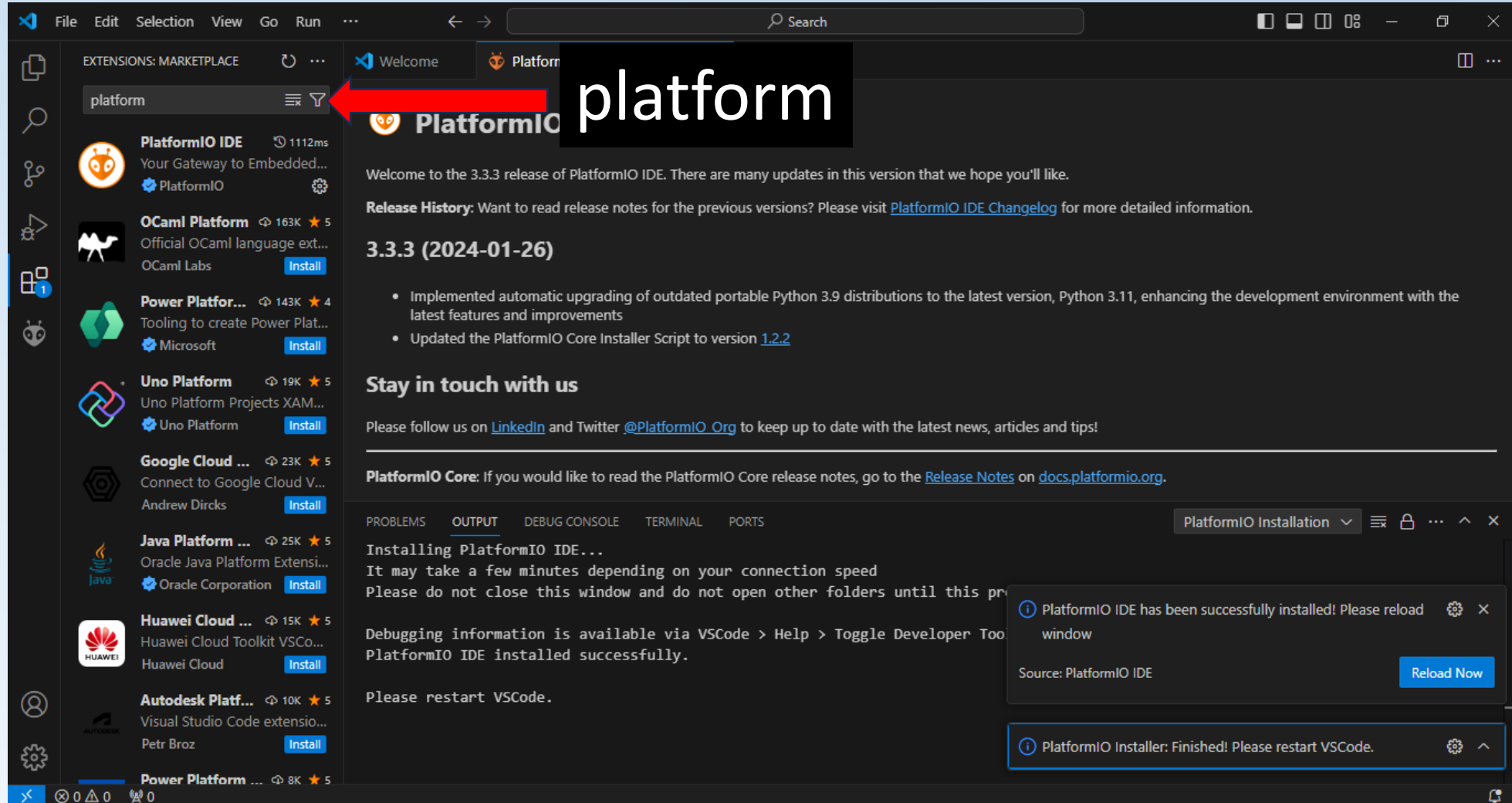
15

Visual Studio Code (VSC) & PlatformIO



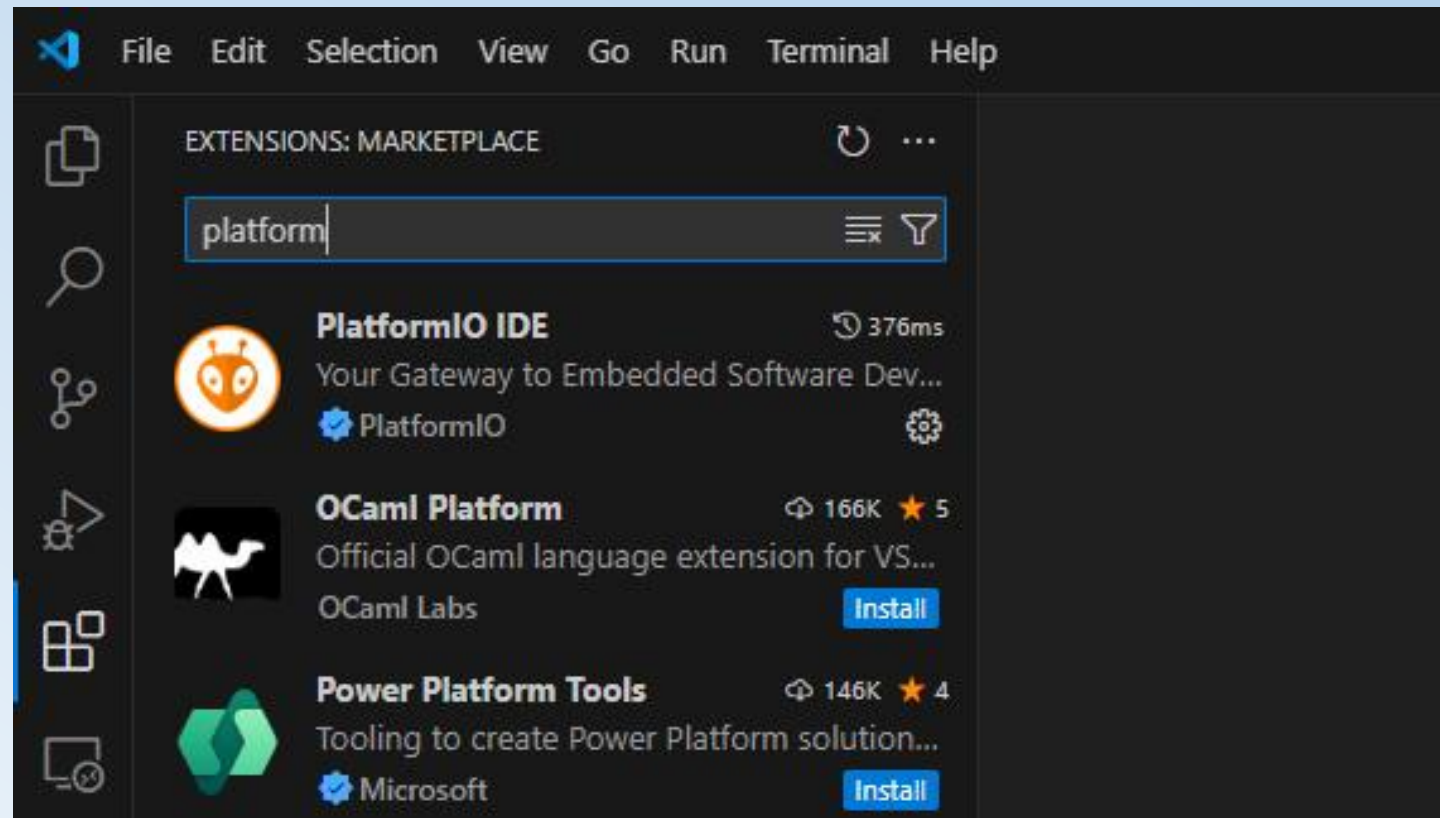
16

Visual Studio Code (VSC) & PlatformIO



17

Visual Studio Code (VSC) & PlatformIO



Visual Studio Code Layout

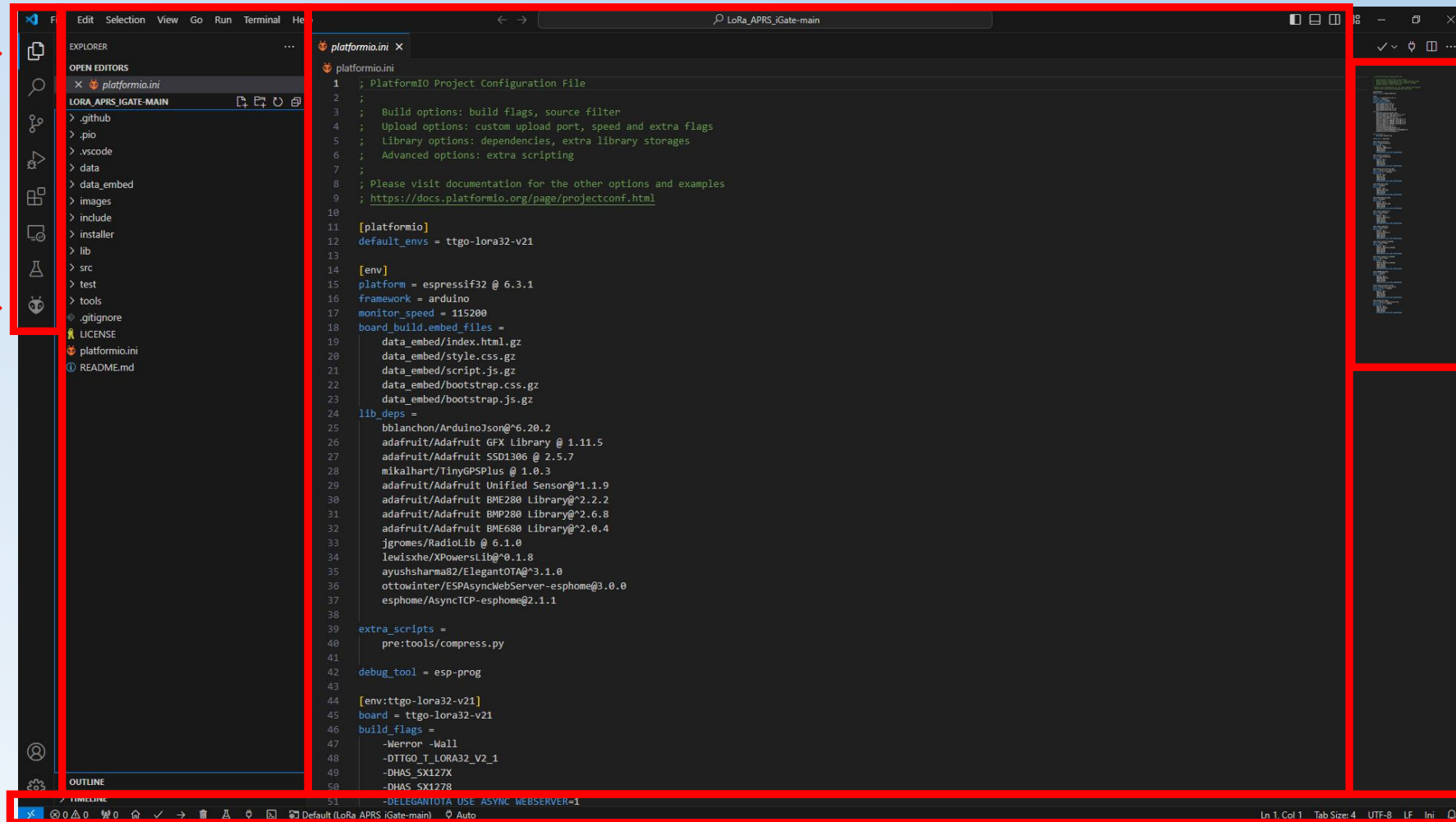
Explorer

Editor

Activity
Bar

Übersicht

Status Bar



19

APRS Passcode

<https://apps.magicbug.co.uk/passcode/>

APRS Passcode Generator

Important: Do not use fake callsigns to inject content into the APRS-IS network you will likely get banned.

Callsign

Technical Example of Passcode Generation using PHP

Source code available on [Github](#)

This tool was developed by Peter Goodhall [2M0SQL](#) who also runs [magicbug](#)

20

APRS Passcode

<https://apps.magicbug.co.uk/passcode/>

Beispiel für das Rufzeichen: DG1EAD

APRS Passcode Generator

Your Passcode: 18340

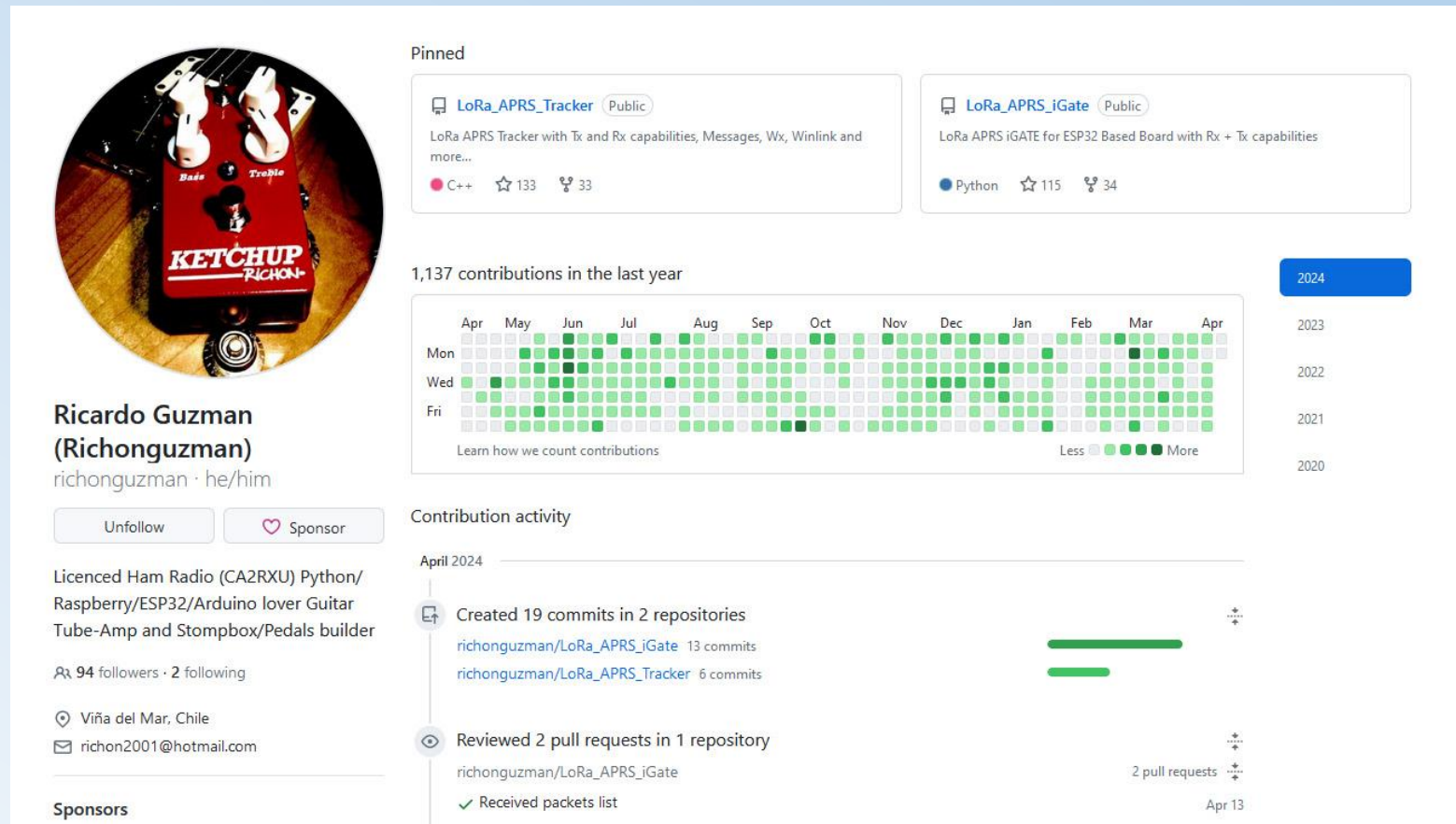
[Generate a new Passcode](#)

Technical Example of Passcode Generation using PHP

Source code available on [Github](#)

This tool was developed by Peter Goodhall [2M0SQL](#) who also runs [magicbug](#)

Download und Programmierung der I-Gate Firmware



The screenshot shows the GitHub profile of Ricardo Guzman (Richonguzman). The profile includes a circular profile picture of a red guitar pedal labeled 'KETCHUP RICHON'. The bio states: 'Licenced Ham Radio (CA2RXU) Python/ Raspberry/ESP32/Arduino lover Guitar Tube-Amp and Stompbox/Pedals builder'. It also shows 94 followers and 2 following. The location is Viña del Mar, Chile, and the email is richon2001@hotmail.com. There are buttons for 'Unfollow' and 'Sponsor'. The 'Pinned' section shows two repositories: 'LoRa_APRS_Tracker' (C++, 133 stars, 33 forks) and 'LoRa_APRS_iGate' (Python, 115 stars, 34 forks). The '1,137 contributions in the last year' section shows a calendar heatmap with a legend for 'Less' and 'More' contributions. The 'Contribution activity' section for April 2024 shows: 'Created 19 commits in 2 repositories' (13 for LoRa_APRS_iGate, 6 for LoRa_APRS_Tracker) and 'Reviewed 2 pull requests in 1 repository' (LoRa_APRS_iGate). A green checkmark indicates 'Received packets list'.

Profile: Ricardo Guzman (Richonguzman) richonguzman · he/him

Bio: Licenced Ham Radio (CA2RXU) Python/ Raspberry/ESP32/Arduino lover Guitar Tube-Amp and Stompbox/Pedals builder

Stats: 94 followers · 2 following

Location: Viña del Mar, Chile

Email: richon2001@hotmail.com

Pinned Repositories:

- LoRa_APRS_Tracker** (Public) C++ 133 stars 33 forks
- LoRa_APRS_iGate** (Public) Python 115 stars 34 forks

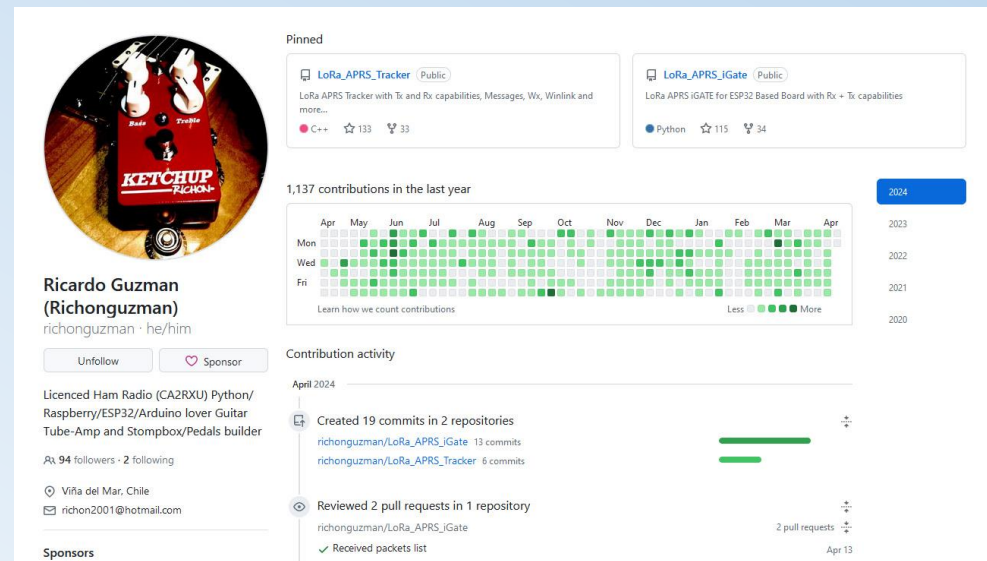
Contributions: 1,137 contributions in the last year

Contribution activity (April 2024):

- Created 19 commits in 2 repositories
 - richonguzman/LoRa_APRS_iGate 13 commits
 - richonguzman/LoRa_APRS_Tracker 6 commits
- Reviewed 2 pull requests in 1 repository
 - richonguzman/LoRa_APRS_iGate
- Received packets list

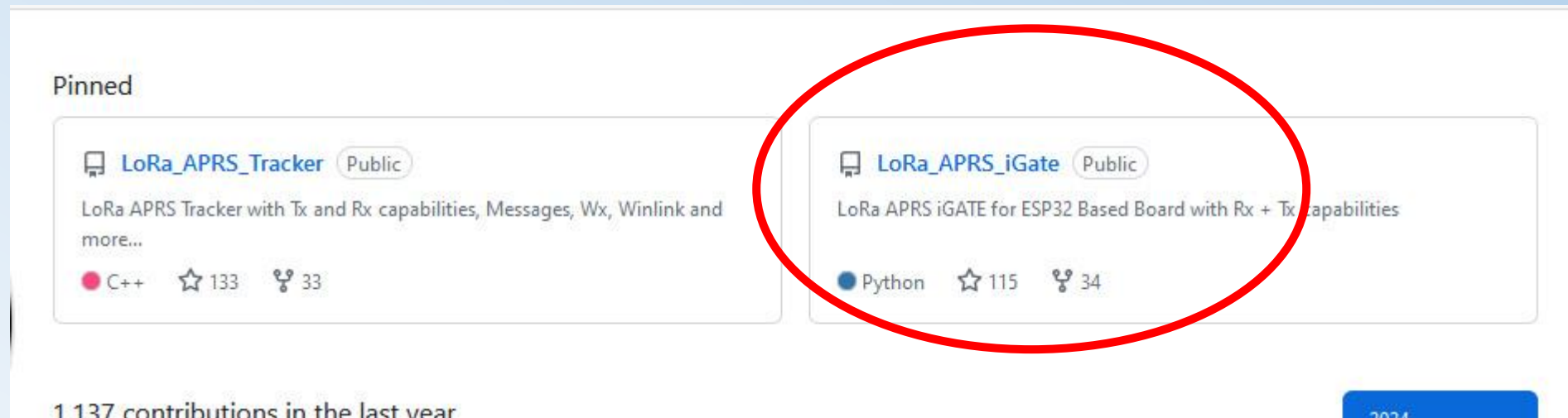
Download und Programmierung der I-Gate Firmware

<https://github.com/richonguzman>



Download und Programmierung der I-Gate Firmware

<https://github.com/richonguzman>



Download und Programmierung der I-Gate Firmware

LoRa_APRS_iGate Public

Sponsor Watch 20 Fork 34 Star 115

main 15 Branches 2 Tags

Go to file Add file Code

richonguzman just small typo ✓ 0fb2c95 · 11 minutes ago 471 Commits

.github	fix: rename heltec wireless lite to ht-ct62	3 weeks ago
.vscode	minor code update	last month
data	improvement: low voltage cut off as web ui option	3 weeks ago
data_embed	feat: received packets list	2 days ago
images	new picture	3 weeks ago
include	first-beta1	last year
installer	fix: av false positive	last month
lib	first-beta1	last year
src	just small typo	11 minutes ago
test	first-beta1	last year

About

LoRa APRS iGATE for ESP32 Based Board with Rx + Tx capabilities

Readme MIT license Activity 115 stars 20 watching 34 forks Report repository

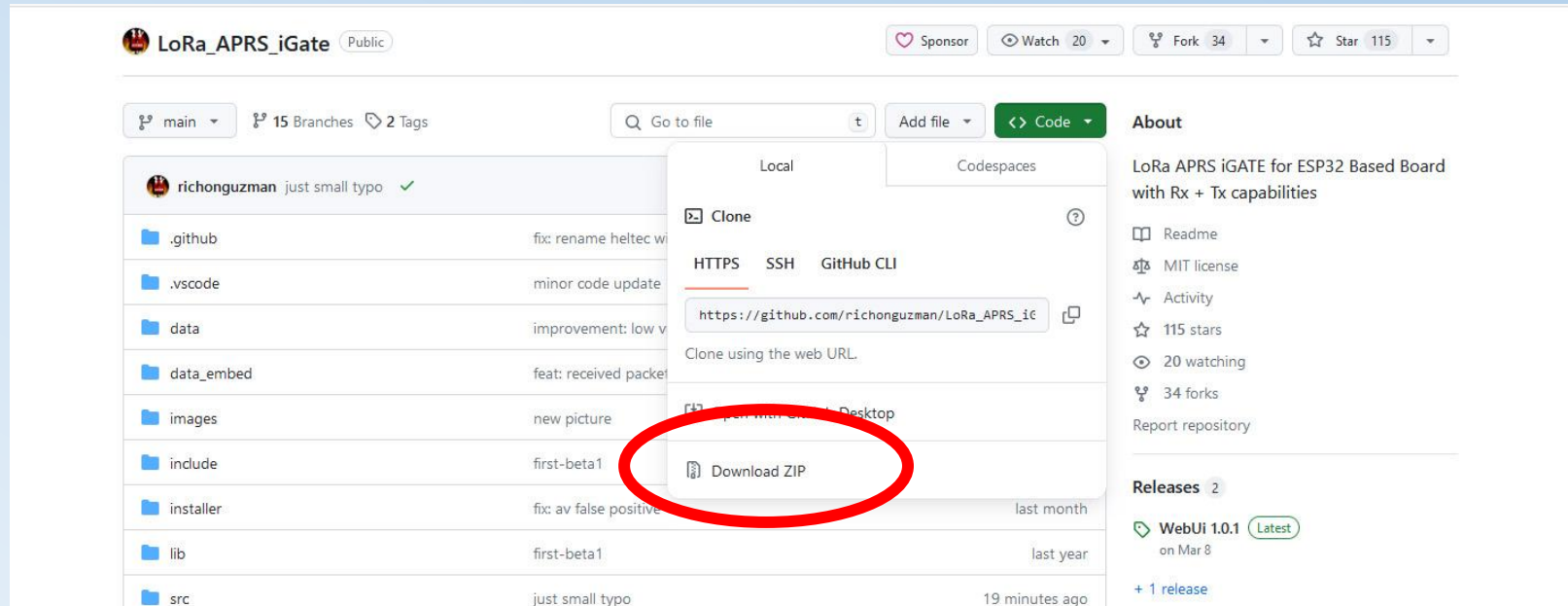
Releases 2

WebUi 1.0.1 Latest on Mar 8

+ 1 release

25

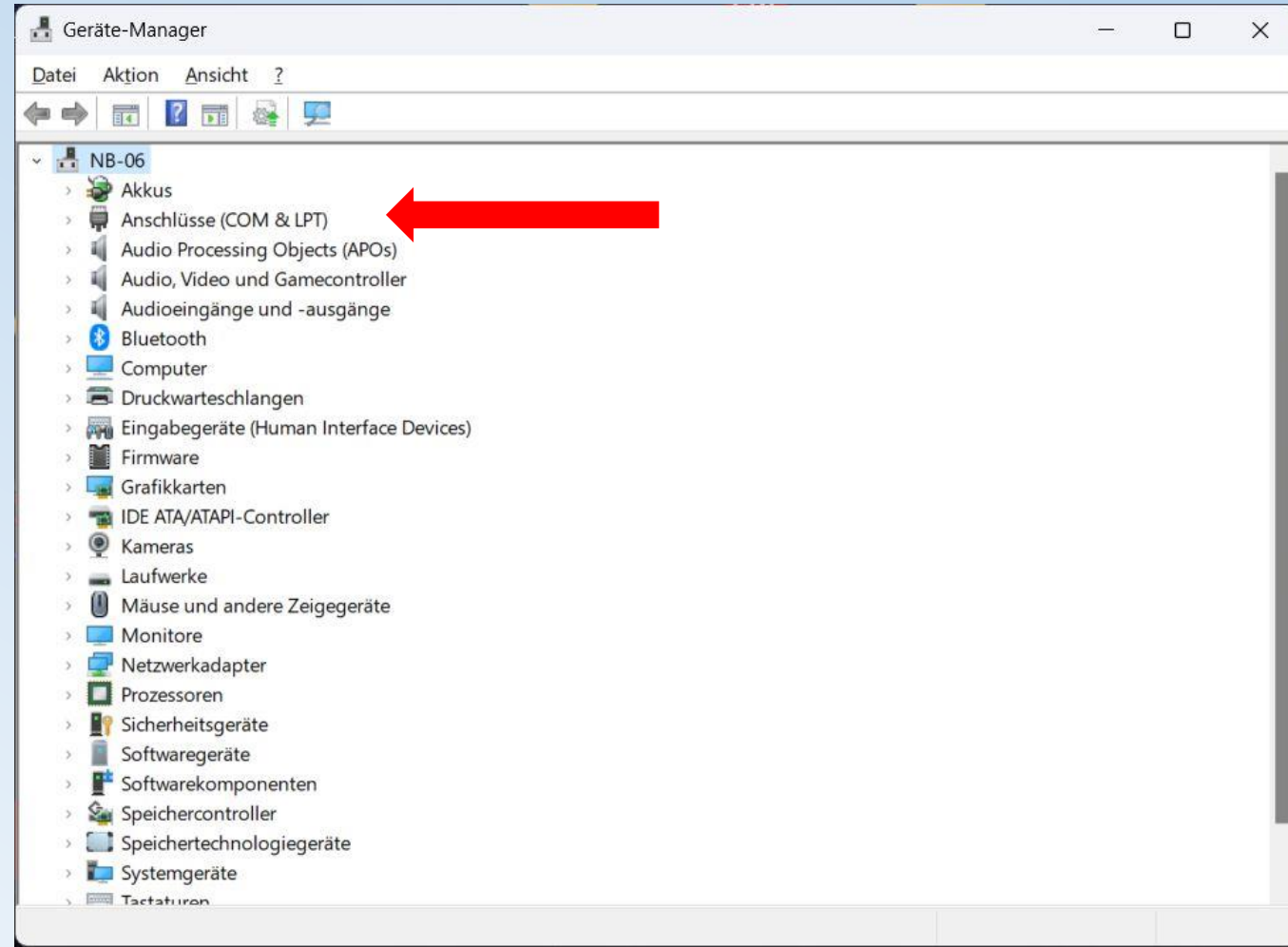
Download und Programmierung der I-Gate Firmware



Abspeichern im bevorzugten Ordner und in dem Ordner dann entpacken.

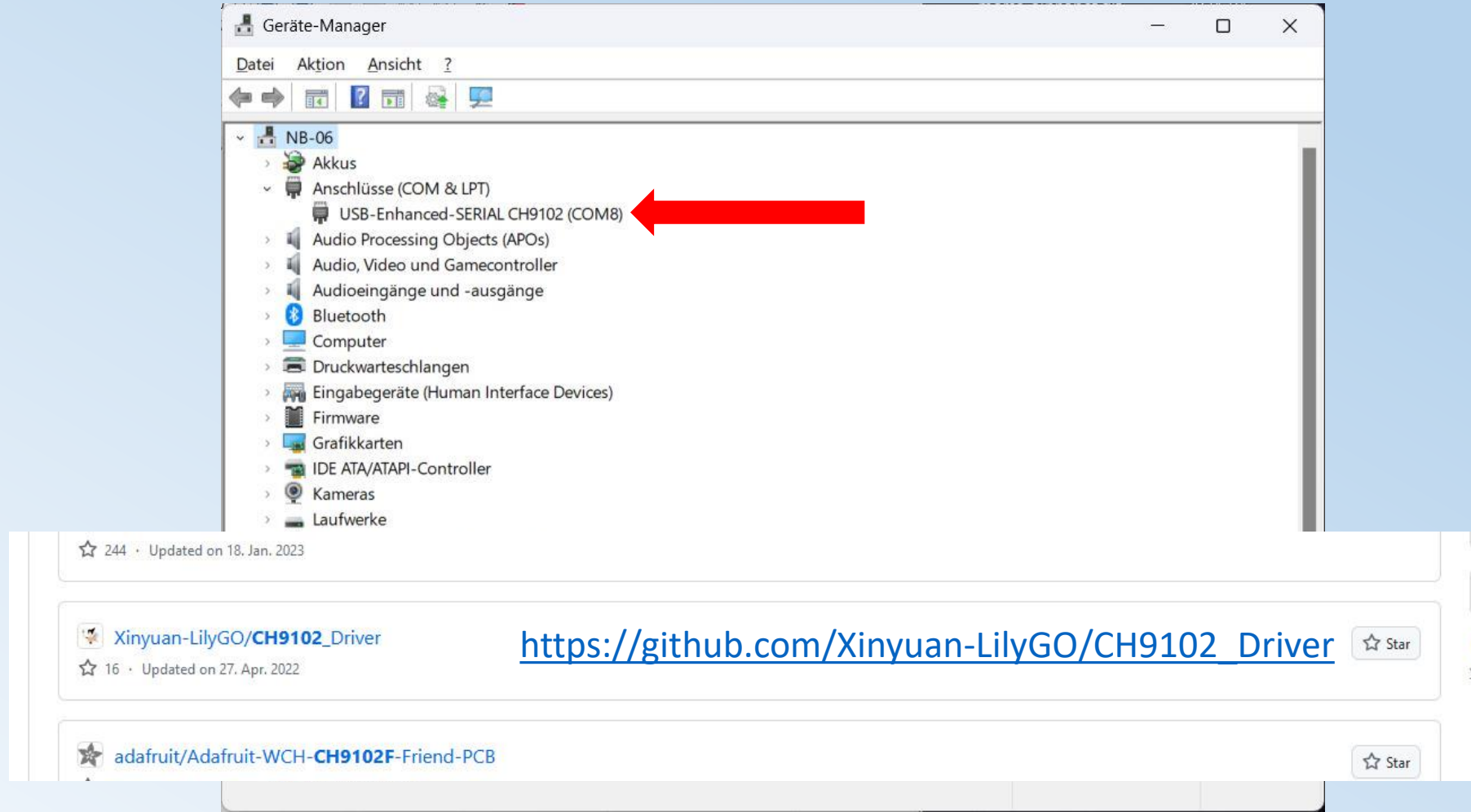
Wichtig!! **immer** eine Arbeitskopie erstellen, damit im Notfall auf das Original zurückgegriffen werden kann, wenn man die Programmierung “zerschossen” hat.

Anschluss TTGO ESP32

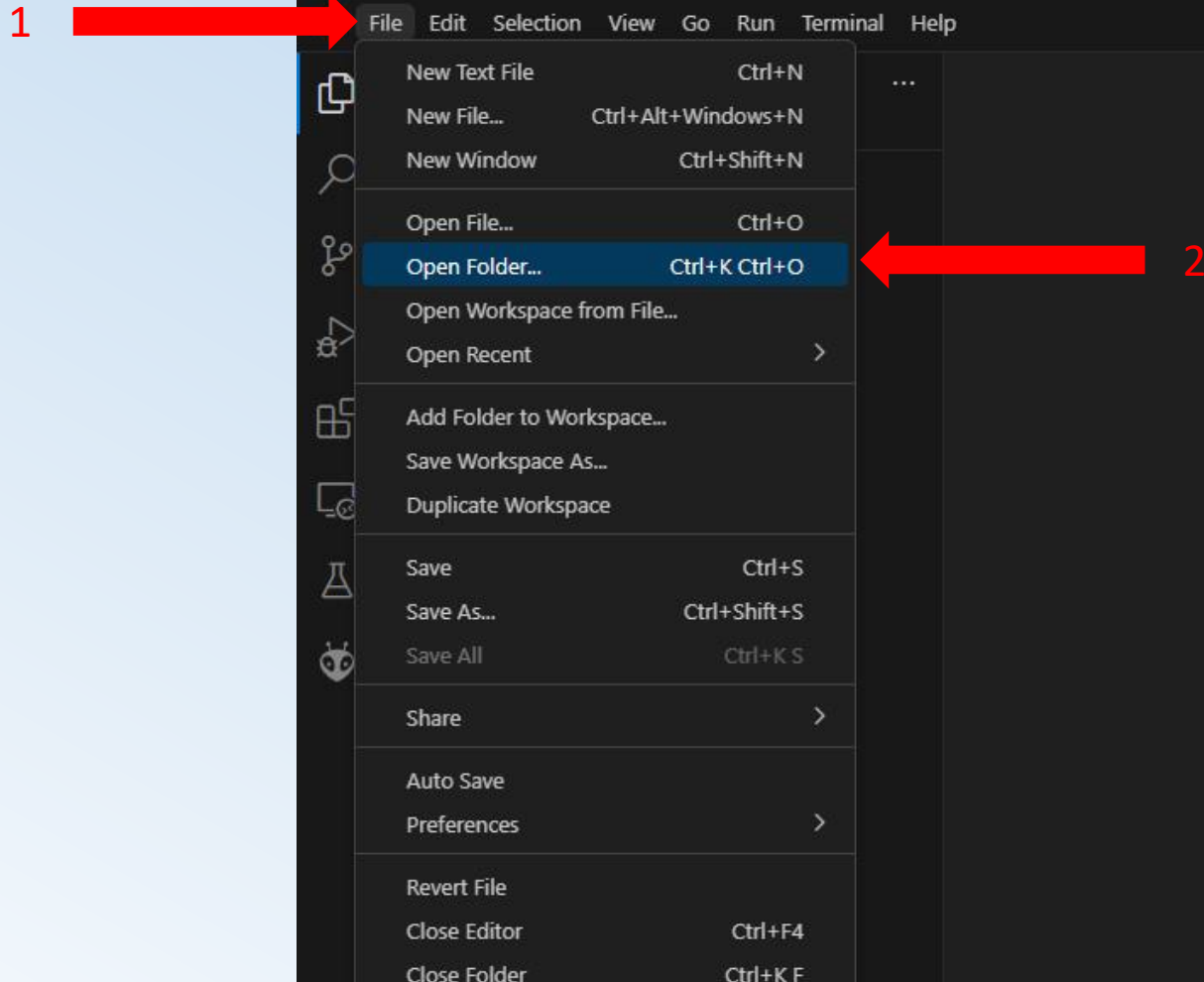


27

Anschluss TTGO ESP32

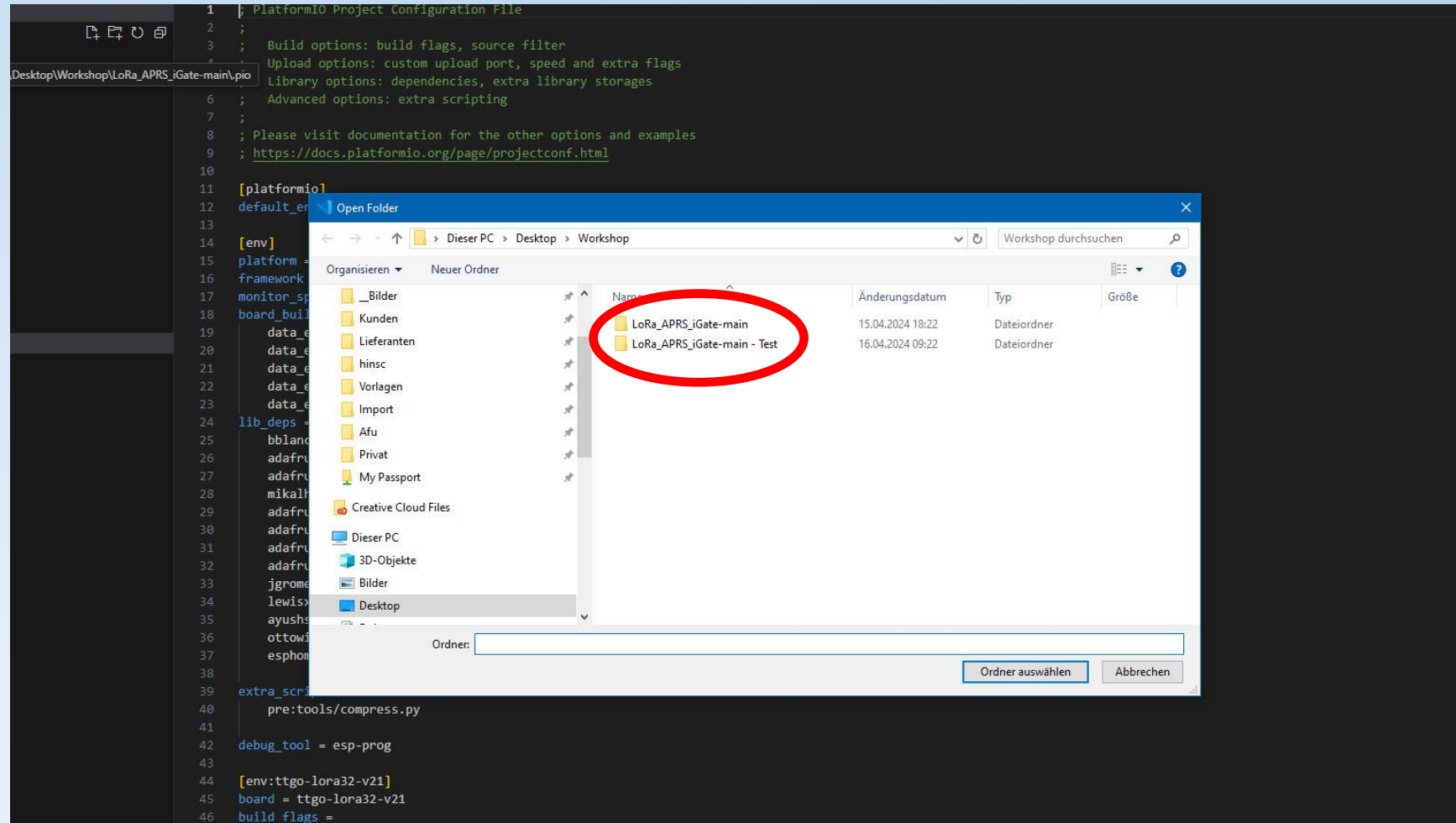


Programmierung der I-Gate Firmware



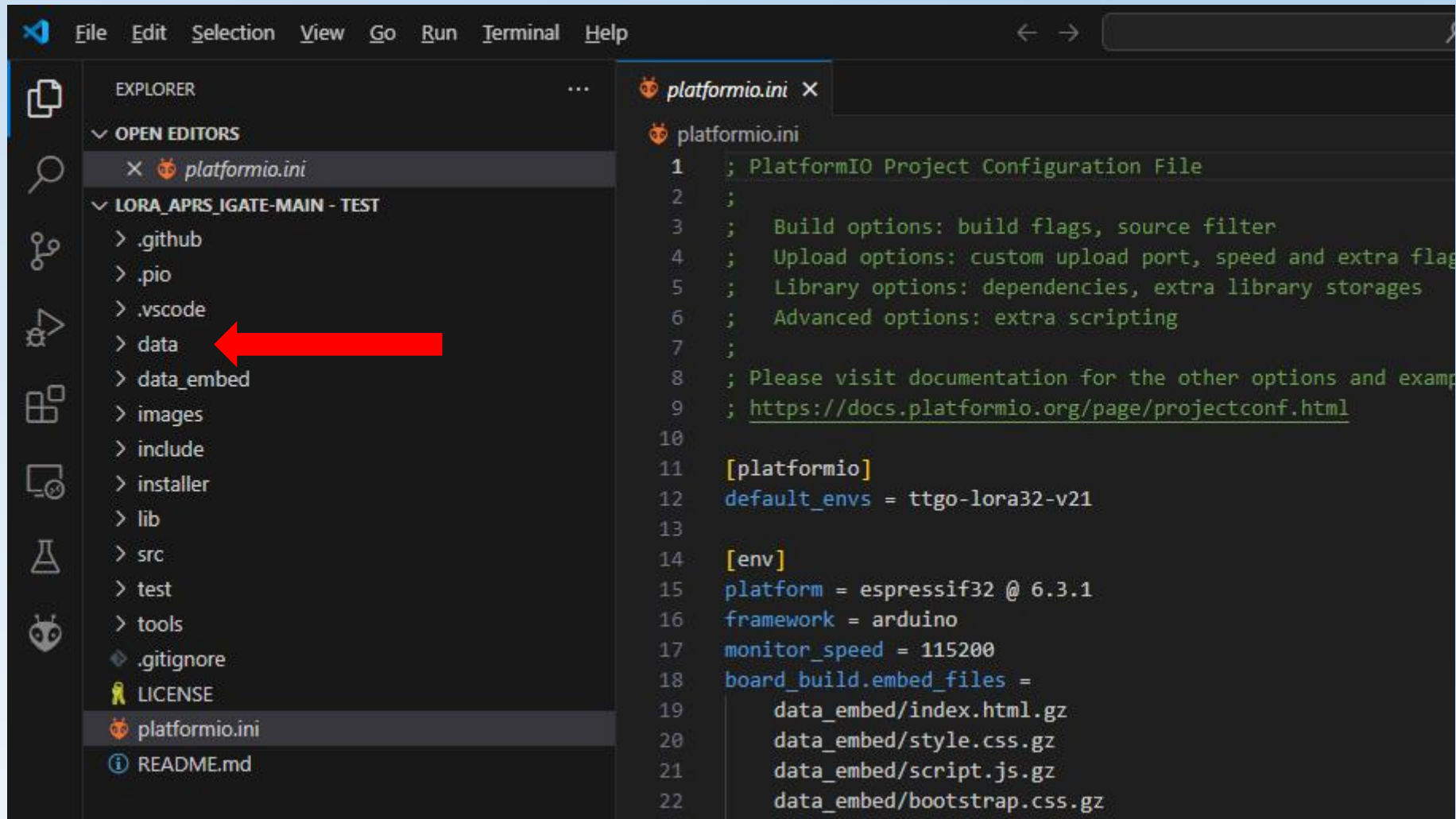
29

Programmierung der I-Gate Firmware



30

Programmierung der I-Gate Firmware

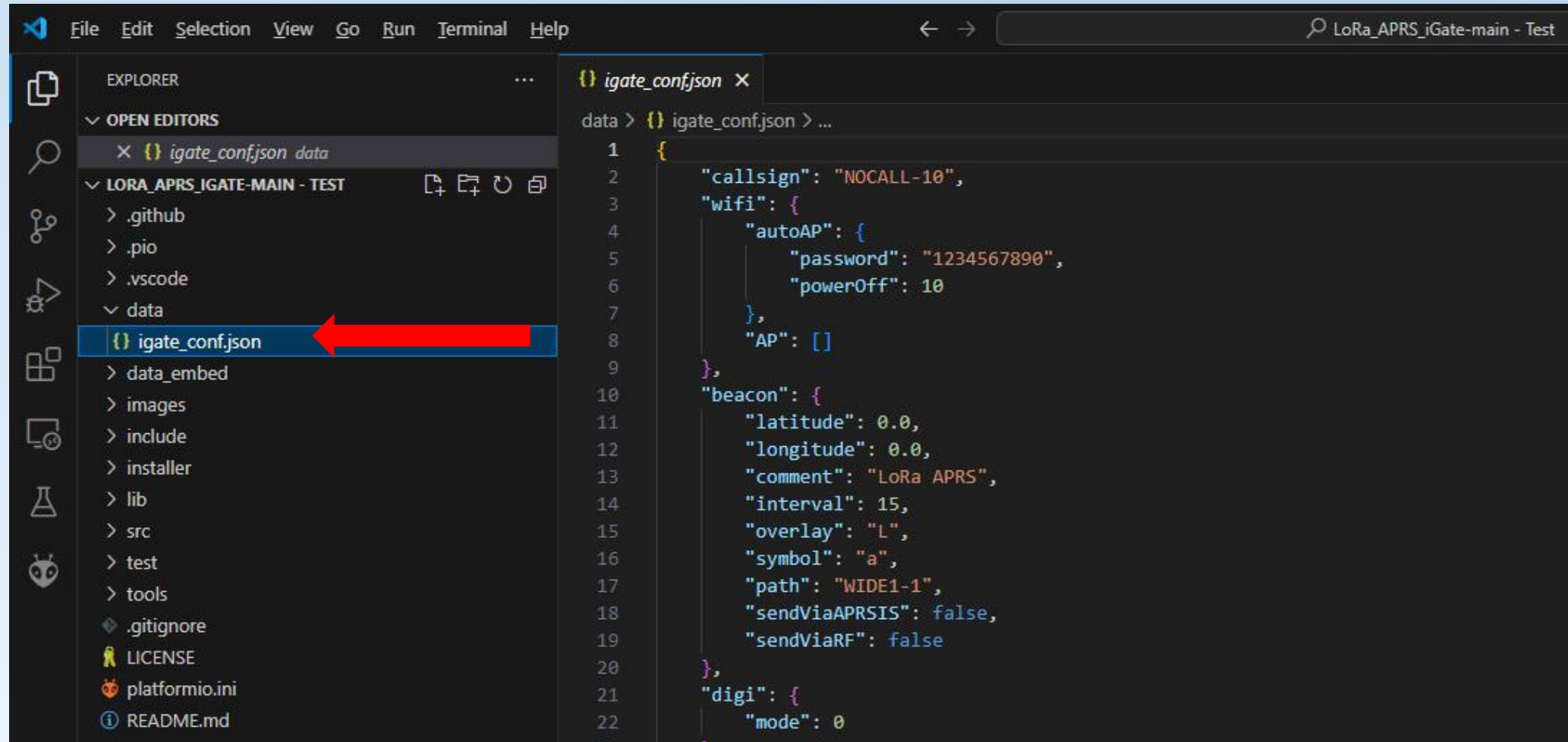


The screenshot shows the Visual Studio Code interface with the PlatformIO extension. The Explorer sidebar on the left displays the project structure for 'LORA_APRS_IGATE-MAIN - TEST'. A red arrow points to the 'data' folder. The main editor window shows the 'platformio.ini' file, which contains configuration for the PlatformIO project.

```
platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [platformio]
12 default_envs = ttgo-lora32-v21
13
14 [env]
15 platform = espressif32 @ 6.3.1
16 framework = arduino
17 monitor_speed = 115200
18 board_build.embed_files =
19     data_embed/index.html.gz
20     data_embed/style.css.gz
21     data_embed/script.js.gz
22     data_embed/bootstrap.css.gz
```

31

Programmierung der I-Gate Firmware



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'LoRa_APRS_iGate-main - Test'. The 'data' folder is expanded, and 'igate_conf.json' is selected, highlighted with a red arrow. The main editor area shows the contents of 'igate_conf.json', which is a JSON configuration file. The file is currently open in a tab labeled 'igate_conf.json'. The JSON content is as follows:

```
data > {} igate_conf.json > ...
1  {
2    "callsign": "NOCALL-10",
3    "wifi": {
4      "autoAP": {
5        "password": "1234567890",
6        "powerOff": 10
7      },
8      "AP": []
9    },
10   "beacon": {
11     "latitude": 0.0,
12     "longitude": 0.0,
13     "comment": "LoRa APRS",
14     "interval": 15,
15     "overlay": "L",
16     "symbol": "a",
17     "path": "WIDE1-1",
18     "sendViaAPRSIS": false,
19     "sendViaRF": false
20   },
21   "digi": {
22     "mode": 0
```

32

Die Parameter

```
{} igate_conf.json X
data > {} igate_conf.json > ...
1  {
2    "callsign": "NOCALL-10",
3    "wifi": {
4      "autoAP": {
5        "password": "1234567890",
6        "powerOff": 10
7      },
8      "AP": [
9        {
10         "ssid": "WIFI_1",
11         "password": "WIFI_1_passwort",
12         "latitude": 0.000000,
13         "longitude": 0.000000
14       },
15       {
16         "ssid": "WIFI_2",
17         "password": "WIFI_2_passwort",
18         "latitude": 0.000000,
19         "longitude": 0.000000
20       }
21     ]
22   },
23 }
```

Ändern in eigenes Call & SSID (1-15)

Zugangsdaten des LoRa wenn kein Wlan empfangen wird
zusätzliche Erklärung folgt später

Hier kommt die SSID und das Passwort des
Wlan-Zugangs von zu Hause rein

hier kann ein weiterer WLAN-Zugang
eingetragen werden

Die Parameter

```
20 },
21 "beacon": {
22   "latitude": 0.0,
23   "longitude": 0.0,
24   "comment": "LoRa APRS",
25   "interval": 15,
26   "overlay": "L",
27   "symbol": "a",
28   "path": "WIDE1-1",
29   "sendViaAPRSIS": false,
30   "sendViaRF": false
31 },
32 "digi": {
33   "mode": 0
34 },
35 "tnc": {
36   "enableServer": false,
37   "enableSerial": false,
38   "acceptOwn": false
39 },
```

Ändern in Koordinaten wie bei Google Maps
z. B.: 51.1699697, 6.8808112 (7 Dezimalstellen)

Bakentext, z.B.: LoRa APRS I-Gate Duesseldorf

Bakenintervall in Minute, bitte nicht kleiner als 15

Symboloverlay, nicht ändern

Symbol der Station:
Der Standardwert ist „a“
„#“ ist ein grüner Stern,
„&“ ist eine schwarze Raute,
„a“ ist eine rote Raute,
„_“ ist ein blauer Kreis (erforderlich für Wx-Daten).

34

Die Parameter

```
20 },
21 "beacon": {
22     "latitude": 0.0,
23     "longitude": 0.0,
24     "comment": "LoRa APRS",
25     "interval": 15,
26     "overlay": "L",
27     "symbol": "a",
28     "path": "WIDE1-1",
29     "sendViaAPRSIS": false,
30     "sendViaRF": false
31 },
32 "digi": {
33     "mode": 0
34 },
35 "tnc": {
36     "enableServer": false,
37     "enableSerial": false,
38     "acceptOwn": false
39 }
```

Standardwert ist WIDE1-1, nicht ändern

Senden der Bake über APRS-IS auf „true“ stellen!!!

Aktiviert das Senden der Bake über LoRa

Aktiviert das Digipeating über LoRa

? keine Beschreibung

Die Parameter

```
39  },
40  "aprs_is": {
41    "active": false,
42    "passcode": "XYZVW",
43    "server": "rotate.aprs2.net",
44    "port": 14580,
45    "filter": "m/10",
46    "toRF": false
47  },
48  "loras": {
49    "txFreq": 433775000,
50    "rxFreq": 433775000,
51    "spreadingFactor": 12,
52    "signalBandwidth": 125000,
53    "codingRate4": 5,
54    "power": 20,
55    "txActive": false,
56    "rxActive": true
57  },
58  "display": {
59    "alwaysOn": true,
60    "timeout": 4,
61    "turn180": false
62  },
63  }
```

auf "true" stellen

Aktiviert das Senden der empf. LoRa Pakete zu APRS-IS.

Hier kommt der Passcode rein

ändern auf: "germany.aprs2.net"

Standard, vorerst so lassen

Standard, vorerst so lassen

Sendeleistung bei Digipeating in dBm
20dBm = 100mW
18dBm = 63mW
15dBm = 32mW
10dBm = 10mW

36

Die Parameter

```
39 },
40 "aprs_is": {
41     "active": false,
42     "passcode": "XYZVW",
43     "server": "rotate.aprs2.net",
44     "port": 14580,
45     "filter": "m/10",
46     "toRF": false
47 },
48 "lora": {
49     "txFreq": 433775000,
50     "rxFreq": 433775000,
51     "spreadingFactor": 12,
52     "signalBandwidth": 125000,
53     "codingRate4": 5,
54     "power": 20,
55     "txActive": false,
56     "rxActive": true
57 },
58 "display": {
59     "alwaysOn": true,
60     "timeout": 4,
61     "turn180": false
62 },
```

Standard, vorerst so lassen

immer an, „false“ schaltet das Display nach „timeout“ ab
Zeit in Sek. wann das Display aus geht
Horizontale Drehung des Displays, so lassen

37

Die Parameter

```
63  "syslog": {  
64      "active": false,  
65      "server": "192.168.0.100",  
66      "port": 514  
67  },  
68  "bme": {  
69      "active": false  
70  },  
71  "ota": {  
72      "username": "",  
73      "password": ""  
74  },  
75  "other": {  
76      "rememberStationTime": 30,  
77      "sendBatteryVoltage": false,  
78      "externalVoltageMeasurement": false,  
79      "externalVoltagePin": 34,  
80      "lowPowerMode": false,  
81      "lowVoltageCutOff": 0  
82  }  
83 }
```

Aktiviert die Übertragung zu einem Syslogserver
IP-Adresse des Syslogservers
Port des Syslogservers, Standard

„true“ ermöglicht die Anzeige der Wx-Daten
(Temperatur, Luftfeuchtigkeit und Druck)
auf dem OLED-Bildschirm und ermöglicht außerdem die
Übertragung der Wx-Daten an den APRS-IS-Server und
die Anzeige als Telemetry.

Der Standardwert ist „false“. Die Modi iGate und
Digipeater können Wx-Daten senden.

Der Anschluss des BME280 an das iGate erfolgt:
SDA to IO21
SCL to IO22
GND to GND (any)
VCC to 3.3V

38

Die Parameter



```
71  "ota": {  
72    "username": "",  
73    "password": "",  
74  },  
75  "other": {  
76    "rememberStationTime": 30,  
77    "sendBatteryVoltage": false,  
78    "externalVoltageMeasurement": false,  
79    "externalVoltagePin": 34,  
80    "lowPowerMode": false,  
81    "lowVoltageCutOff": 0,  
82  },  
83 }
```

geilste Funktion ever

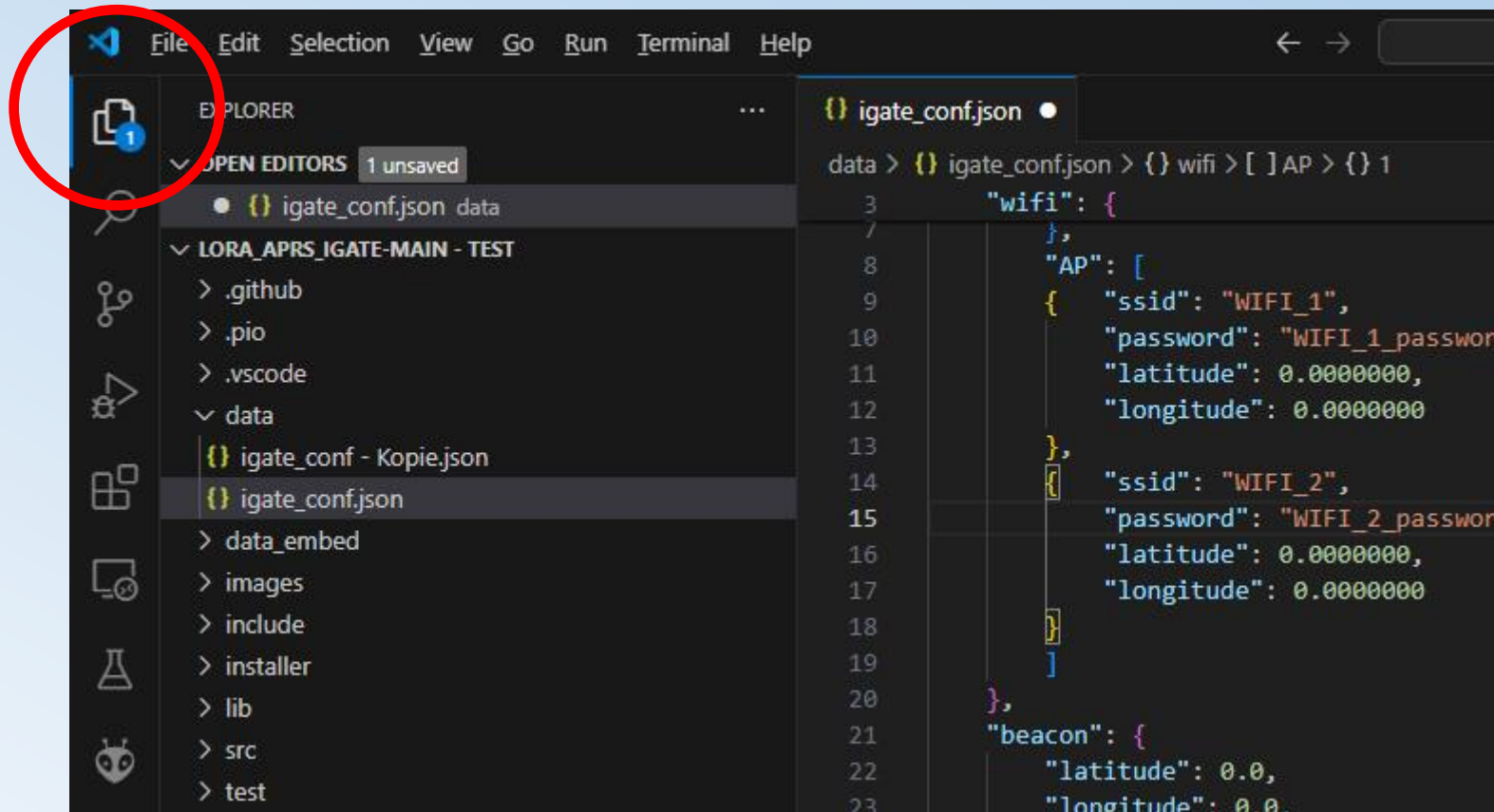
Zeit in Minuten, gehörte Station im Buffer

Überwachung der Batterie am Batterieanschluss

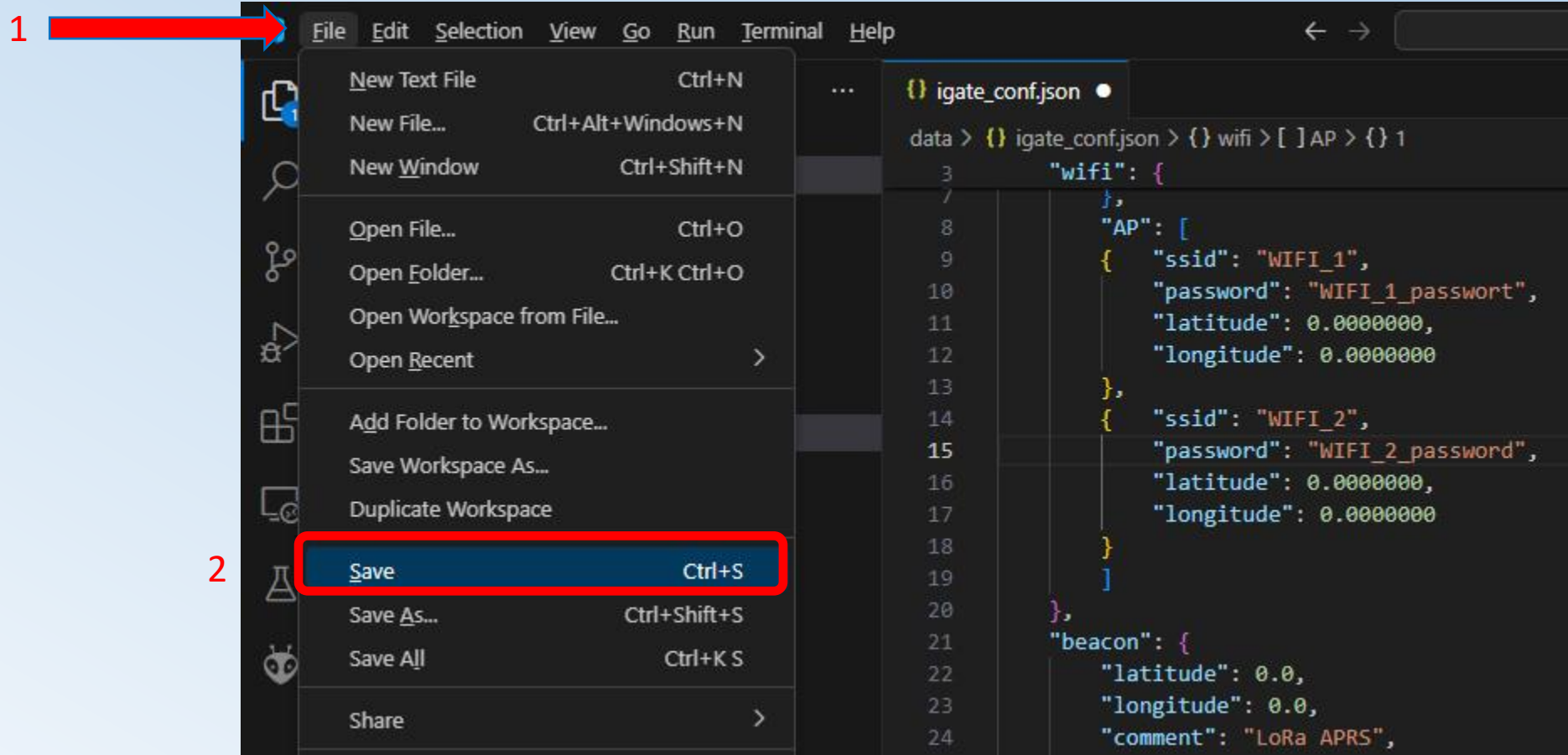
Überwachung externe 12V Batterie mit Spannungsteiler
am PIN 34

nicht näher beschrieben

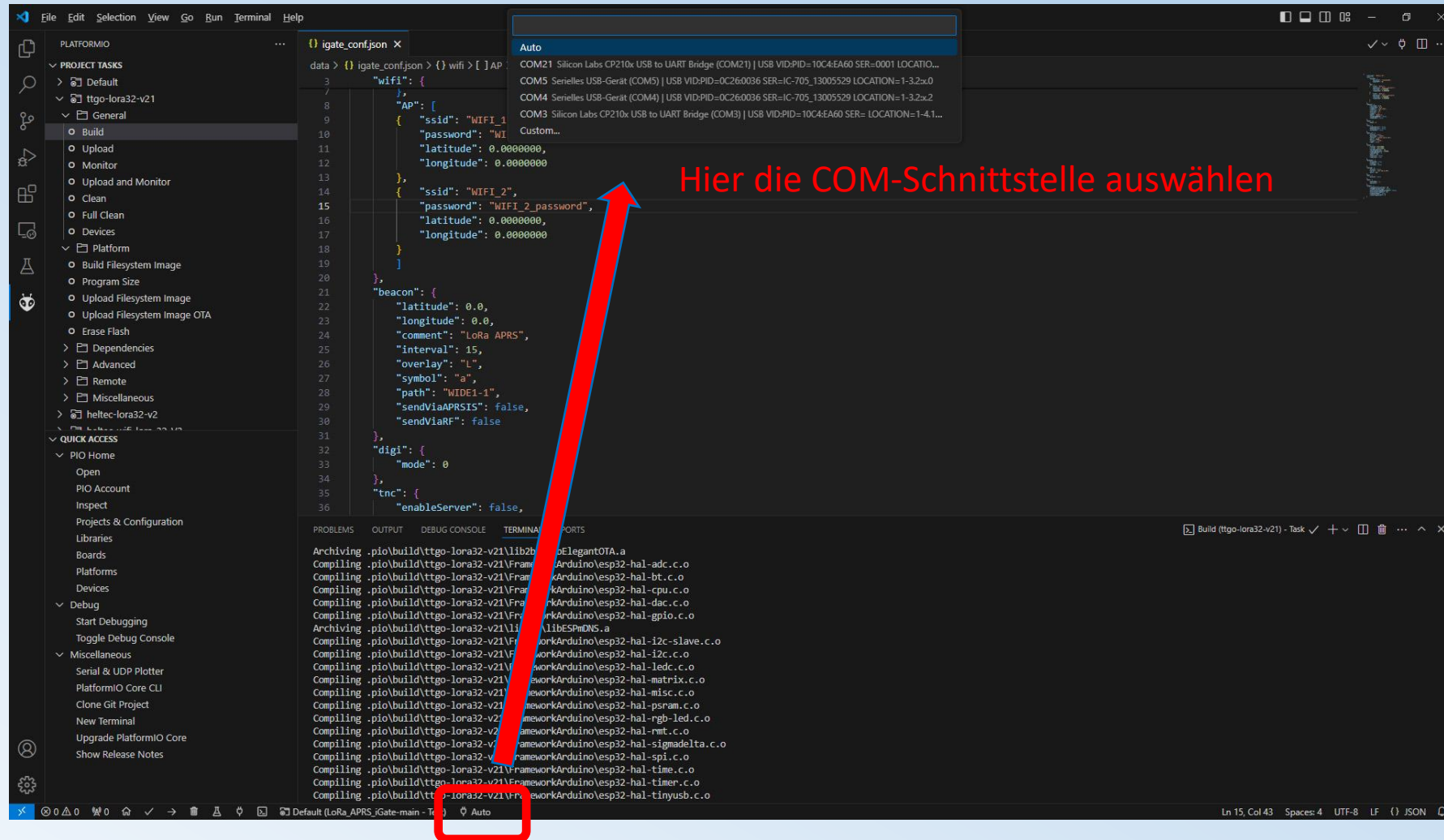
Upload in das I-Gate



Upload in das I-Gate



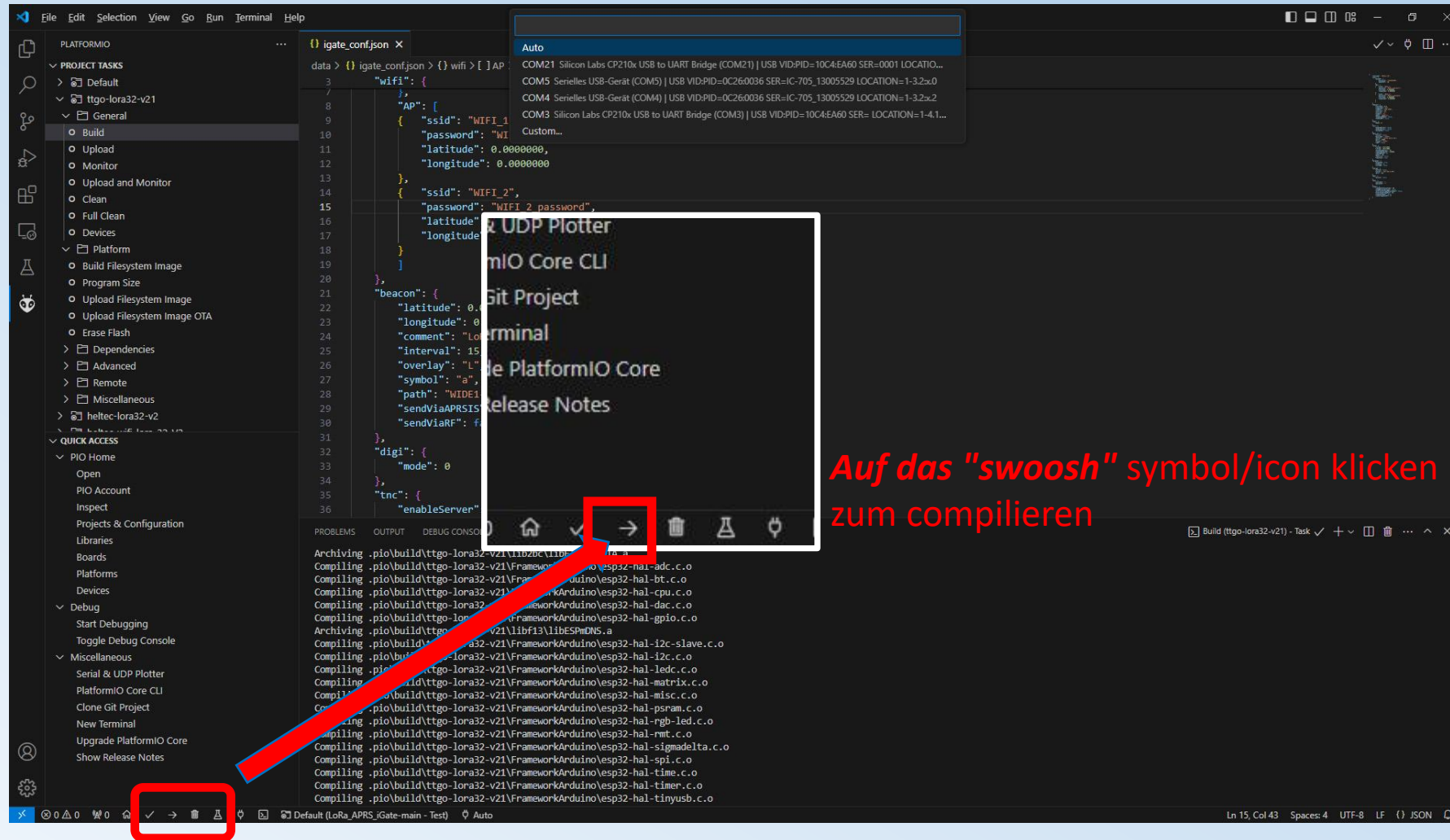
Upload in das I-Gate



42

Upload in das I-Gate

Hier die COM-Schnittstelle auswählen



Auf das "swoosh" symbol/icon klicken zum compilieren

43

Upload in das I-Gate

```
21     "beacon": {
22         "latitude": 0.0,
23         "longitude": 0.0,
24         "comment": "LoRa APRS",
25         "interval": 15,
26         "overlay": "L",
27         "symbol": "a",
28         "path": "WIDE1-1",
29         "sendViaAPRSIS": false,
30         "sendViaRF": false
31     },
32     "digi": {
33         "mode": 0
34     },
35     "tnc": {
36         "enableServer": false,
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

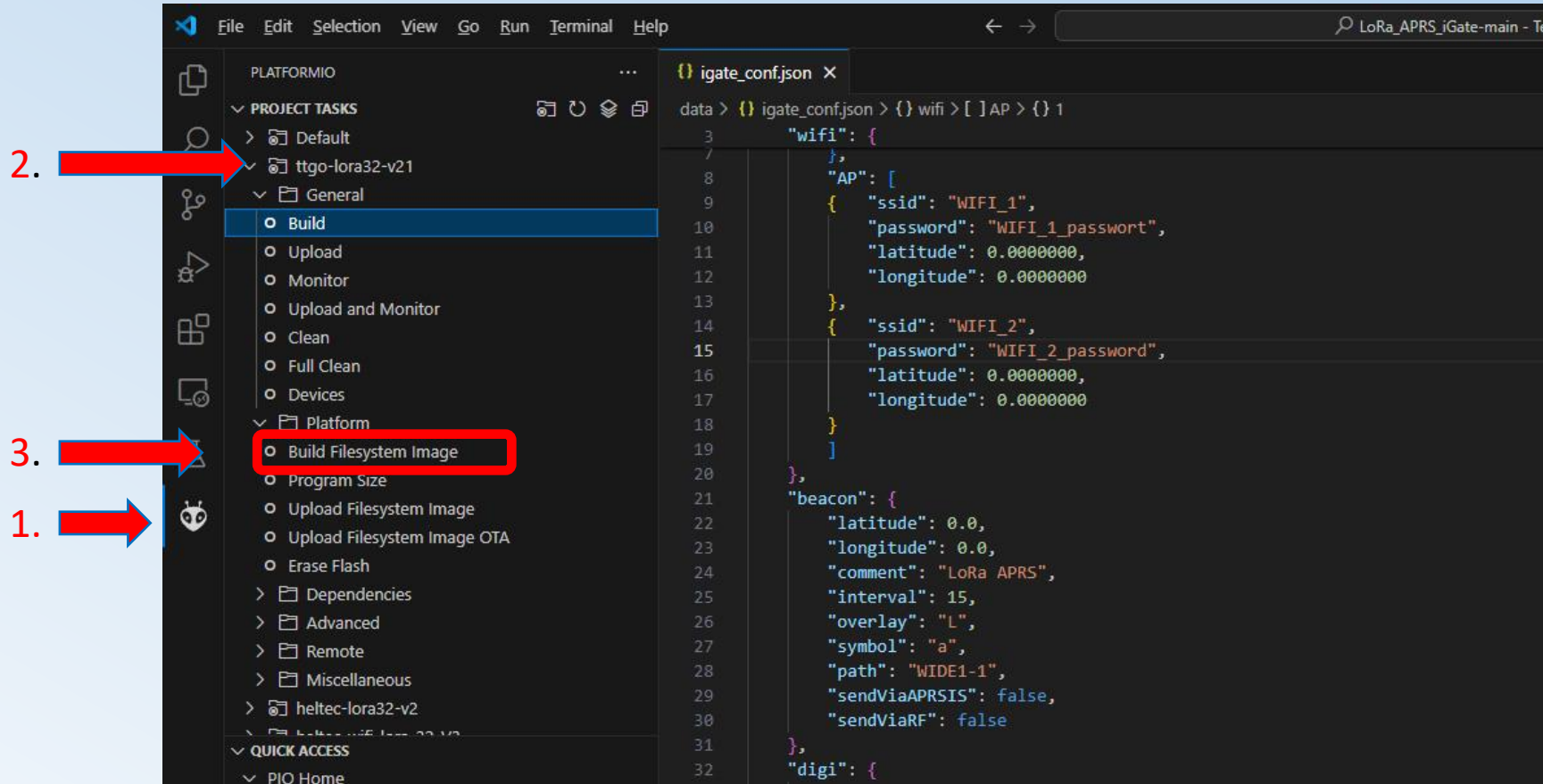
Archiving .pio\build\ttgo-lora32-v21\libFrameworkArduino.a
Linking .pio\build\ttgo-lora32-v21\firmware.elf
Retrieving maximum program size .pio\build\ttgo-lora32-v21\firmware.elf
Checking size .pio\build\ttgo-lora32-v21\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [==] 15.7% (used 51356 bytes from 327680 bytes)
Flash: [=====] 96.3% (used 1262225 bytes from 1310720 bytes)
Building .pio\build\ttgo-lora32-v21\firmware.bin
esptool.py v4.5.1
Creating esp32 image...
merged 25 ELF sections
Successfully created esp32 image.
===== [SUCCESS] Took 50.07 seconds =====

Environment	Status	Duration
ttgo-lora32-v21	SUCCESS	00:00:50.066

===== 1 succeeded in 00:00:50.066 =====

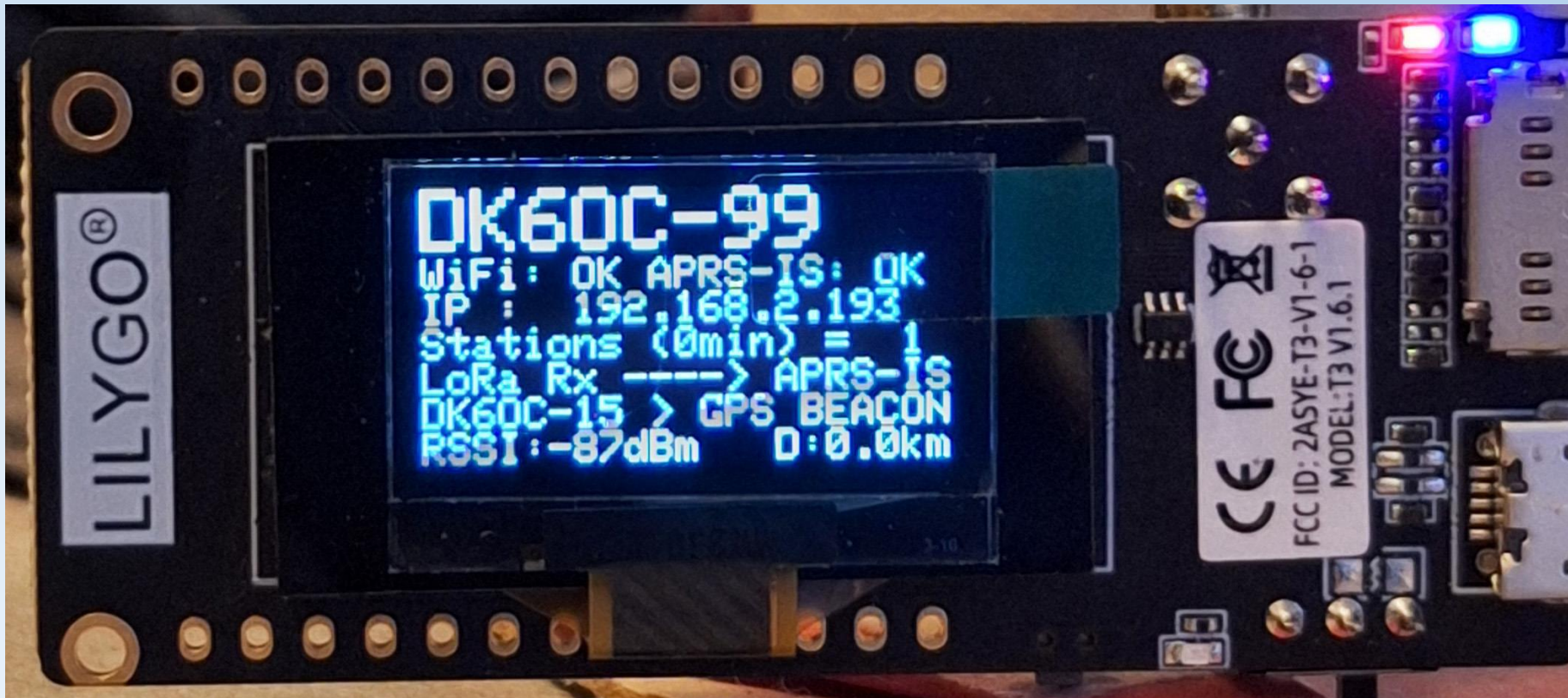
Terminal will be reused by tasks, press any key to close it.

Upload in das I-Gate



45

Upload in das I-Gate



auto AP

auto AP = Automatischer Access Point
das I-Gate stellt für die voreingestellte Zeit (powerOFF =10min)
einen Zugang in Form eines Access Points zur Verfügung.
Im Display wird innerhalb dieser Zeit die IP-Adresse des I-Gates
angezeigt: **192.168.4.1**

Wenn man sich mit dem AP verbindet und nun im Browser die
IP-Adresse **192.168.4.1** eingibt, gelangt man auf HTML-Seite des
I-Gates und kann hier alle Einstellungen vornehmen.

auto AP

192.168.4.1

CA2RXU's LoRa iGate Configuration

OTABackupRestoreSave

 **Station**

Add your ham callsign and SSID. Optionally, you can leave a comment describing your station.

Callsign and SSID

NOCALL-10

Beacon Comment

LoRa APRS

Beacon path

WIDE1-1

Symbol

Red diamond with L



Latitude

0

Longitude

0

auto AP

 **Auto AP**
Add your password and power off timeout to auto access point. Auto AP will start if there is no WiFi connection. Timeout will count from startup or last client disconnected.

Password

Power off timeout (in minutes)

Set to 0 if you don't want this option.

 **WiFi Access**
Add all Wi-Fi Networks intended to be used, and their respective coordinates to georeference the iGate location.

SSID
IKS-2.4

Passphrase



[Add network](#)

WICHTIG!

damit das I-Gate sich mit dem eigenen W-Lan Netz verbinden kann

auto AP

Der Vorteil hierbei ist:

das I-Gate ist im Netzwerk erreichbar und muss nicht mehr an den Rechner mittels USB angeschlossen werden

Dadurch ist auch über VPN eine Erreichbarkeit gegeben
(interessant für Relaisstandorte etc.)

Programmierung der I-Gate Firmware

Epilog:

Die **Grund-Programmierung** der Firmware für das I-Gate erfolgt über „VSC“ mit „platformIO“.

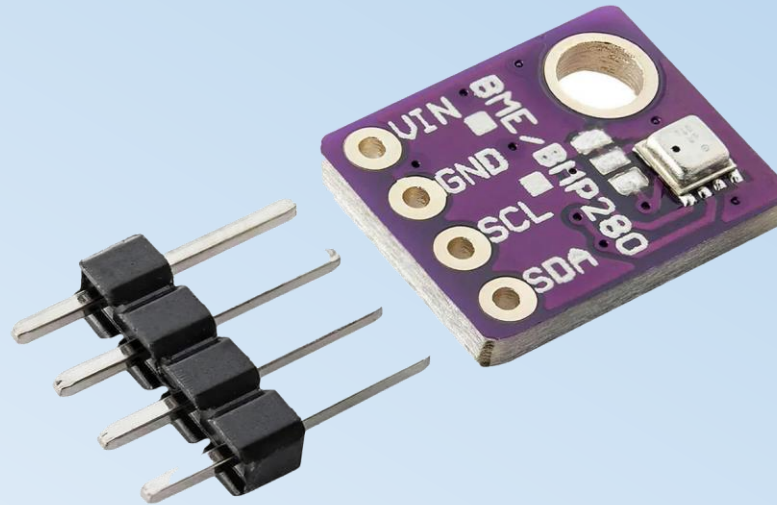
Die Parametrierung kann ebenfalls über „VSC“ erfolgen, jedoch ist die Parametrierung über das Web-Interface bedeutend komfortabler.

Spätere Anpassungen der Parameter oder auch Firmware-updates können aus der Ferne erfolgen, dazu muss das I-Gate nicht mehr per USB mit dem Rechner gekoppelt werden.

Komponenten

BME-280

Barometrischer Sensor für Temperatur, Luftfeuchtigkeit und Luftdruck



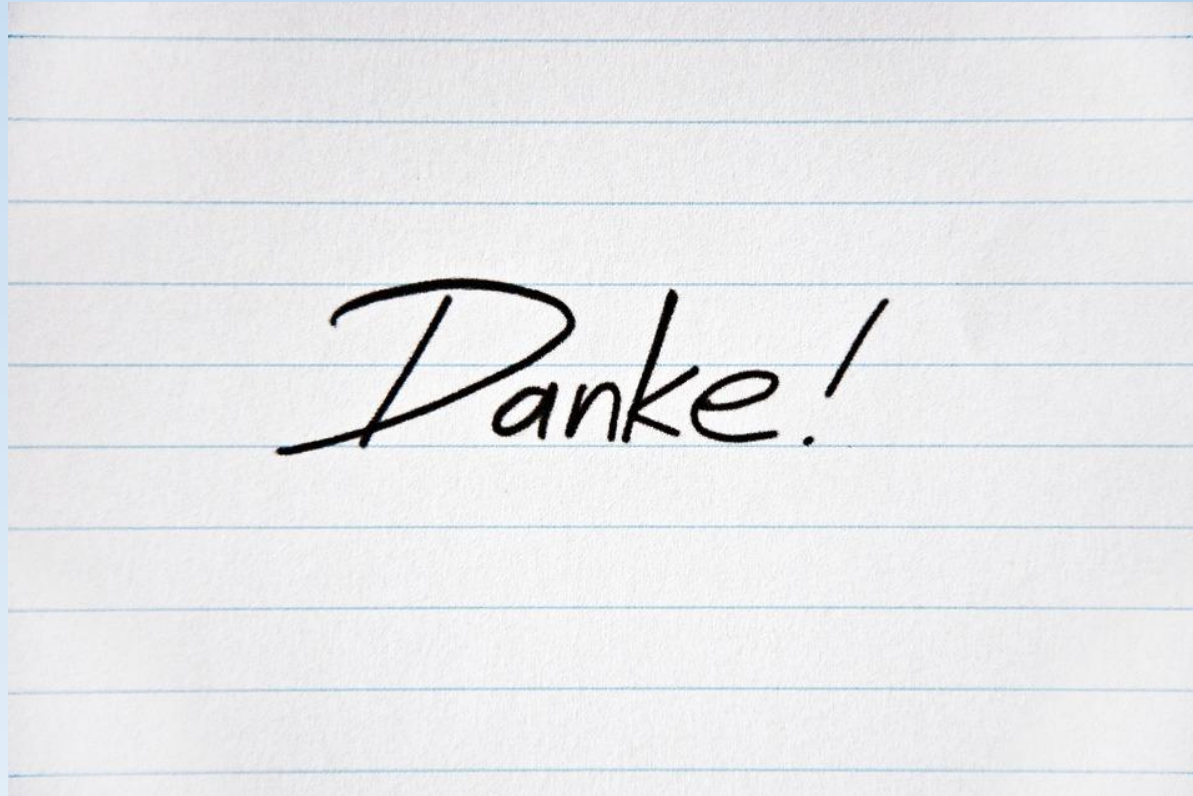
www.az-delivery.de

Habt Ihr noch Fragen?



Pixabay License
Freie kommerzielle Nutzung
Kein Bildnachweis nötig

Vielen Dank . . . vy 73 de



Pixabay License
Freie kommerzielle Nutzung
Kein Bildnachweis nötig