
Inhaltsverzeichnis

MeshCom/MeshCom Start

[Versionsgeschichte interaktiv durchsuchen](#)
[Visuell Wikitext](#)

Version vom 16. Februar 2022, 04:16 Uhr
(Quelltext anzeigen)

[Oe1kbc](#) ([Diskussion](#) | [Beiträge](#))

Markierung: **Visuelle Bearbeitung**

[← Zum vorherigen Versionsunterschied](#)

Version vom 12. Mai 2023, 07:11 Uhr (Quelltext anzeigen)

[Oe1kbc](#) ([Diskussion](#) | [Beiträge](#))

Markierung: **Visuelle Bearbeitung**

[Zum nächsten Versionsunterschied →](#)

Zeile 40:

Genau hier setzt das Projekt '''<span style="font-size:14.0pt;

line-height:107%">MeshCom'''
an. Der MeshCom-Server ist ein Broker-
Server welcher das '''MQTT-Protokoll'''
beherrscht. Jene Leser welche den Aufbau
dieses Protokolls genauer studieren wollen
verweise ich auf den WIKI
Artikel'''https://de.wikipedia.org/wiki/MQTT'''
bzw. auf die Beschreibung des
detaillierten Aufbau des Inhalts mit
dem Namen PROTOBUF'''https://de.
wikipedia.org/wiki/Protocol_Buffers''.

====
>Warum ein eigener MeshCom-Server als
MQTT-Broker====

*Volle
Anpassung an das etwas speziellere MQTT-
Protokoll welches die **Meshtastic**-
Firmware benutzt.Gestaltung im Header wie
Topic, Paket-ID, **Gateway-ID 32-Bit usw.**

*Frei
definierbare Logik bei der Weitergabe der
MQTT-Pakete mit Mengensteuerung<

Zeile 40:

Genau hier setzt das Projekt '''<span style="font-size:14.0pt;

line-height:107%">MeshCom'''
an. Der MeshCom-Server ist ein Broker-
Server welcher das '''MQTT-Protokoll'''
beherrscht. Jene Leser welche den Aufbau
dieses Protokolls genauer studieren wollen
verweise ich auf den WIKI
Artikel'''https://de.wikipedia.org/wiki/
/MQTT''.

====
>Warum ein eigener MeshCom-Server als
MQTT-Broker====

*Volle
Anpassung an das etwas speziellere MQTT-
Protokoll welches die **MeshCom**-Firmware
benutzt. Gestaltung im Header wie Topic,
Paket-ID

*Frei
definierbare Logik bei der Weitergabe der
MQTT-Pakete mit Mengensteuerung<

*Klares
Zuschneiden auf Zwecke der
Funkamateure

*Klares
Zuschneiden auf Zwecke der
Funkamateure

Zeile 52:

====
>Was sind die 1. Schritte=====

– Die
Programmier-Software zum laden der
Firmware in die LoRa-Module ist im ÖVSV-
WIKI Schrittweise beschrieben"[[MeshCo
m/MeshCom-ErsteSchritte|https://wiki.
oevsv.at/wiki/MeshCom/MeshCom-
ErsteSchritte]]".

– Die aktuelle
Firmware, welche schon sehr wichtige
Wünsche zum Projekt aufgenommen hat,
kann aus dem ÖVSV-WIKI geladen werden '
'[[MeshCom/MeshCom-Firmware|https://
wiki.oevsv.at/wiki/MeshCom/MeshCom
-Firmware]]'. Diese Seite stellt auch die
aktuellen Änderungen dar. Ein Auszug der
wichtigen Änderungen um ein
gemeinsames HF-Projekt zu leben sind:<

Zeile 52:

====
>Was sind die 1. Schritte=====

+ Die
Programmier-Software zum laden der
Firmware in die LoRa-Module ist im ÖVSV-
WIKI Schrittweise beschrieben

+ https://icssw.org/download-category/
download-meshcom-4-0-tools/<span
style="color: black">.

+ Die aktuelle
Firmware, welche schon sehr wichtige
Wünsche zum Projekt aufgenommen hat,
kann aus dem ÖVSV-WIKI geladen werden

+ https://icssw.org/download-category/m
eshcom-4-0-client-firmware/
+

+

****Diese Seite stellt auch die aktuellen Änderungen dar. Ein Auszug der wichtigen Änderungen um ein gemeinsames HF-Projekt zu leben sind:

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Bluetooth PIN wurde fix auf "000000"
gesetzt, das erleichtert die Verbindung
mit dem Smartphone via Bluetooth ohne
jeweils das Display ablesen zu müssen.<
/span>
```

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Bluetooth PIN wurde fix auf "000000"
gesetzt, das erleichtert die Verbindung
mit dem Smartphone via Bluetooth ohne
jeweils das Display ablesen zu müssen.<
/span>
```

Zeile 64:

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif">PSK
Encryption NONE,</span>
```

Zeile 70:

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif">PSK
Encryption NONE,</span>
```

-

```
*Channel: Very Long Range Very Slow (
BW125kHz)
```

+

```
*Channel: MediumLongRange (BW250k
Hz)
```

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Region: EU433</span>
```

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Region: EU433</span>
```

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Reconnect zum MeshCom-Server nach
einem Neustart. Wichtig damit die LoRa-
Gateways rasch reconnecten wenn der
MeshCom-Server, gerade in der Testphase
ist das öfters, neu gestartet wurde.<
/span>
```

```
*<span style="font-size:10.5pt;font-family:
'Open Sans',sans-serif"
>Reconnect zum MeshCom-Server nach
einem Neustart. Wichtig damit die LoRa-
Gateways rasch reconnecten wenn der
MeshCom-Server, gerade in der Testphase
ist das öfters, neu gestartet wurde.<
/span>
```

```
<span style="color: black">Wenn die
Firmware mit dem ESPTool für Windows
oder mittels Python esptool geladen ist
verbindet man sich mit einer Konsole (ich
verwende eine GITBash-Konsole) um die
Konfigurations-Befehle zu laden. Diese
sind:</span>
```

```
<span style="color: black">Wenn die
Firmware mit dem ESPTool für Windows
oder mittels Python esptool geladen ist
verbindet man sich mit einer Konsole (ich
verwende eine GITBash-Konsole) um die
Konfigurations-Befehle zu laden. Diese
sind:</span>
```

-

-

– `""LoRa-Nodes""`

– `*""meshtastic --set-owner OE9XXX""`

– `*""meshtastic --set region EU433""`

– `*""meshtastic --ch-index 0 --ch-set
psk none""`

– `Die
Befehle können auch i einer Zeile
eingegeben werden:`

– `*""meshtastic --set-owner OE9XXX --
set region EU433 --ch-index 0 --ch-set
psk none""`

– `Sollten
mehrere Module am PC
angeschlossen sein muss die USB-
Schnittstelle zusätzlich in jeder
Befehlszeile gesetzt werden:`

– `*""meshtastic --port com22 --set-
owner OE9XXX""`

– `Wenn ein
LoRa-Module verwendet wird welcher
keinen GPS-Empfänger verbaut hat
kann die Position bei Bedarf auch fix
eingestellt werden. Wichtig! Die
Positions-Parameter unbedingt in
einer Zeile setzen:`

– `*""meshtastic --port com22 --setlat
44.33 --setlon 15.5315 --setalt 252""`

Der Breitengrad und der Längengrad werden in Dezimalgraden ausgedrückt. Die Seehöhe wird in Ganzzahl und Metern angegeben.

Die gesetzten Parameter können mit folgendem Befehl überprüft werden:

***'''meshtastic -port com22 -info'''**

[[Datei:LoRa-Konfiguration.png|gerahmt|Ausgabe auf Befehl: '''meshtastic --info''']]

Ein erster Test ob man im MeshCom-Netz ankommt ist die Eingabe einer Text-Meldung über die selbe Konsole welche zum Konfigurieren verwendet wurde:

***'''meshtastic --sendtext 'hello world'''**

Kontrolle am Dashboard unter Menü ACTIVITY:

***via HAMNET [http://meshcom.ampr.at/>'''http://meshcom.ampr.at''']**

*via
INTERNET '''[https://srv08.oevsv.
at/mqtt
>
>https://srv08.oevsv.at/mqtt<
/span>]'''

'''LoRa-Gateways<
/span>'''

Es
werden noch zusätzlich zu obigen
LoRa-Nodes Konfigurations-Befehlen
folgende Befehle gespeichert:

*'''meshtastic --set wifi_ap_mode
false'''

*'''meshtastic --set wifi_ssid 'AP-
SSID'''

*'''meshtastic --set wifi_password 'AP-
PASSWORD'''

*'''meshtastic --set mqtt_server
44.143.8.143'''

*'''meshtastic --ch-index 0 --ch-set
uplink_enabled true'''

*'''meshtastic --ch-index 0 --ch-set
downlink_enabled true'''

Wichtig
sind folgende Vorgangsweisen:<
/span>

– *****Neustart des Gateway-Nodes nach erfolgter Konfiguration bzw. jeder Änderung*****

– *****Ein LoRa-Gateway läuft instabil wenn man es im Betrieb am PC/Laptop angeschlossen bleibt. Es sollte auch kein Debug- oder Log-Modus während eines Regelbetriebs gestartet sein.*****

– **Die Kontrolle der LoRa-Gateway-Verbindung kann über das Dashboard:**

– ***via HAMNET [http://meshcom.ampr.at/http://meshcom.ampr.at"]**

– ***via INTERNET ***[https://srv08.oevsv.at/mqtt https://srv08.oevsv.at/mqtt]*** erfolgen.**

– **
**

====Wie kann ich mittels meines LoRa-Nodes Text-Meldungen absenden bzw. empfangen.====

====Wie kann ich mittels meines LoRa-Nodes Text-Meldungen absenden bzw. empfangen.====

[[Datei:MeshCom Textmessages.
jpg|links|rahmenlos]]

[[Datei:MeshCom Textmessages.
jpg|links|rahmenlos]]

Zeile 135:

Zeile 82:

*
>""Kommando-Zeile aber nur zum
Senden von Text""

*
>""Kommando-Zeile aber nur zum
Senden von Text""

- *""**WEB-
Interface muss aber zusätzlich
installiert werden und ist im ÖVSV-
WIKI beschrieben**""

+

*"""**ANDROID
APP**"" ""**IPhone APP**""

- *"""**ANDROID
APP** <nowiki>[https://meshtastic.org
/docs/software/android/android-
installation]</nowiki>" "<sp
an style="color: black">**Hinweis: ist
nicht im Google Playstore sondern
nur im Amazon Appstore zu
erhalten.**""

+

- *""**IPhone APP** [https://meshtastic.disc
ourse.group/t/meshtastic-ios-app-first-
alpha-release/2733<nowiki>] Dieser
Download ist ein Testflight. <
</nowiki>
>**Die Anzahl der Tester ist leider vom
Entwickler limitiert und zweitweise
abgelaufen.**""

+

https://icssw.org/meshcom-app/

+

[[Datei:MeshComKarte.jpg|mini|**IPhone
APP - Kartendarstellung**]]

+

- [[Datei:MeshComKarte.
jpg|mini|**IPhone APP -
Kartendarstellung**]]

====
>Derzeit gibt es LoRa-Getways in OE und
DL:====

====
>Derzeit gibt es LoRa-Getways in OE und
DL:====

Version vom 12. Mai 2023, 07:11 Uhr

[zurück zu Kategorie:MeshCom](#)

Was benötigt man um am MeshCom Projekt teilzunehmen?

von Kurt OE1KBC - Referat für nat. & int. Projekte

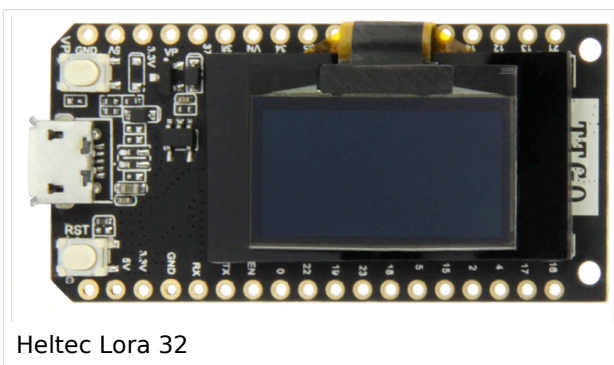
TBEAM, TLORA, HELTEC & Co.

Die aktuelle Meshtastic Firmware (1.2.53 ff) ist auf Boards, welche einen ESP32 oder nRF52 Prozessor, einen LoRa-Chip, GPS von Ublox und WIFI sowie Bluetooth Hardware am Modul anbieten, von einigen Herstellern verfügbar. Am Markt sehr günstig zu erhalten sind die Boards von Lilygo, Heltec und Wisblock. Auf Einkaufsplattform direkt in China Banggood oder via AMAZON mit folgenden Begriffen suchen.

- **Lilygo TTGO T-Beam**
- **Lilygo TTGO Lora**
- **Heltec Lora 32 (V2)**
- **Wisblock RAK4631**



TBEAM Lora mit OLED-Display



Heltec Lora 32



TBEAM im 3D-gedrucktem Gehäuse

Beim Kauf sind wichtige Hardware-Features zu beachten

- **Ganz wichtig Frequenz EU433**
- Soll ein GPS-Modul vorhanden sein? (TTGO_LORA hat das nicht)
- Soll das Modul gleich im Gehäuse geliefert werden?
- Entweder ist ein OLED-Display bereits verbaut oder wird zum selbst auflöten mitgeliefert.
- Ein Netzgerät mit 5V USB-A Buchse, wird meist nicht mitgeliefert, sollte aber fast immer im Shake vorhanden sein.

Was ist der Unterschied zwischen LoRa-Node und LoRa-Gateway

LoRa-Nodes mit Meshtastic-Firmware bilden, wenn sich die Funkmodule gegenseitig hören, eine Mesh-HF-Wolke aus wo die Kommunikation wie in einem kleinen Netzwerk funktioniert. Es können SMS-Meldungen, POSITIONS-Meldungen, NODE-Informationen und/oder MESSWERT-Daten ausgetauscht werden. Wichtig! dass man auf allen teilnehmenden NODES das gleiche Frequenzband (70cm) und die gleichen Modulationsparameter (Datenrate, Spreadingfaktor, usw.) eingestellt hat.

Was ein großer Vorteil der Mesh-Vernetzung ist, dass sich nicht alle teilnehmenden NODES tatsächlich auf dem HF-Weg „hören“ müssen. Datenpakete werden über Partner-NODES solange mittels HF-Übertragung weiter gegeben bis diese alle LoRa-Nodes erreicht haben. Wie lange ein Paket weiter gegeben wird hängt unter anderem vom HOP-Limit ab (siehe Konfiguration).

LoRa-Gateways welche ebenfalls mit Meshtastic-Firmware geflashed wurden, können auch ein Gateway zu einem sogenannten Broker-Server aufbauen. Die Übermittlung wird mit dem Protokoll **Message Queuing Telemetry Transport (MQTT)** vermittelt. Ein Broker-Server dient dazu, die Datenpakete, egal welcher Inhalts-Type, zwischen mehreren LoRa-Gateways auszutauschen.

Genau hier setzt das Projekt **MeshCom** an. Der MeshCom-Server ist ein Broker-Server welcher das **MQTT-Protokoll** beherrscht. Jene Leser welche den Aufbau dieses Protokolls genauer studieren wollen verweise ich auf den WIKI Artikel <https://de.wikipedia.org/wiki/MQTT>.

Warum ein eigener MeshCom-Server als MQTT-Broker

- Volle Anpassung an das etwas speziellere MQTT-Protokoll welches die MeshCom-Firmware benutzt. Gestaltung im Header wie Topic, Paket-ID
- Frei definierbare Logik bei der Weitergabe der MQTT-Pakete mit Mengensteuerung
- Klares Zuschneiden auf Zwecke der Funkamateure
- Schnittstellen zu anderen Message-Systemen wie APRS, DAPNET, HAMMessenger, TELEGRAM BOT
- Skalierbare Vernetzung von mehreren MeshCom-Servern mit Berücksichtigung der teilweise fragilen HAMNET-Vernetzungswege.
- Skalierbare Vernetzung von Großregionen

Was sind die 1. Schritte

Die Programmier-Software zum laden der Firmware in die LoRa-Module ist im ÖVSV-WIKI Schrittweise beschrieben

<https://icssw.org/download-category/download-meshcom-4-0-tools/>.

Die aktuelle Firmware, welche schon sehr wichtige Wünsche zum Projekt aufgenommen hat, kann aus dem ÖVSV-WIKI geladen werden

<https://icssw.org/download-category/meshcom-4-0-client-firmware/>

Diese Seite stellt auch die aktuellen Änderungen dar. Ein Auszug der wichtigen Änderungen um ein gemeinsames HF-Projekt zu leben sind:

- Bluetooth PIN wurde fix auf "000000" gesetzt, das erleichtert die Verbindung mit dem Smartphone via Bluetooth ohne jeweils das Display ablesen zu müssen.
- HOP-Limit für Nachrichten wurde auf 5 erhöht, damit Nachrichten im Mesh-Netzwerk 5 mal von anderen Nodes weitergesendet werden. Das ist sehr wichtig da an jedem **LoRa-NODE** bzw. einem **LoRa-Gateway** jeweils das HOP-Limit um 1 reduziert wird und bei „0“ wird das Paket nicht weiter verteilt.
- MeshCom logo und ÖVSV link
- Beide Sleep-Modi (light sleep, deep sleep) sind deaktiviert damit das Modul jederzeit zur Kommunikation mit dem Netzwerk als auch dem Smartphone zur Kontrolle und Abfrage bereit ist.
- Wifi Refresh für Gateway reduziert auf 5 sec. Damit werden auch ältere Router im Heimnetzwerk bedient, welche die Antwort-Tunnel meist nur 6 Sekunden „offen“ halten.
- fixe Voreinstellung der HF-Parameter von:
- PSK Encryption NONE,
- Channel: MediumLongRange (BW250kHz)
- Region: EU433
- Reconnect zum MeshCom-Server nach einem Neustart. Wichtig damit die LoRa-Gateways rasch reconnecten wenn der MeshCom-Server, gerade in der Testphase ist das öfters, neu gestartet wurde.

Wenn die Firmware mit dem ESPTool für Windows oder mittels Python esptool geladen ist verbindet man sich mit einer Konsole (ich verwende eine GITBash-Konsole) um die Konfigurations-Befehle zu laden. Diese sind:

Wie kann ich mittels meines LoRa-Nodes Text-Meldungen absenden bzw. empfangen.



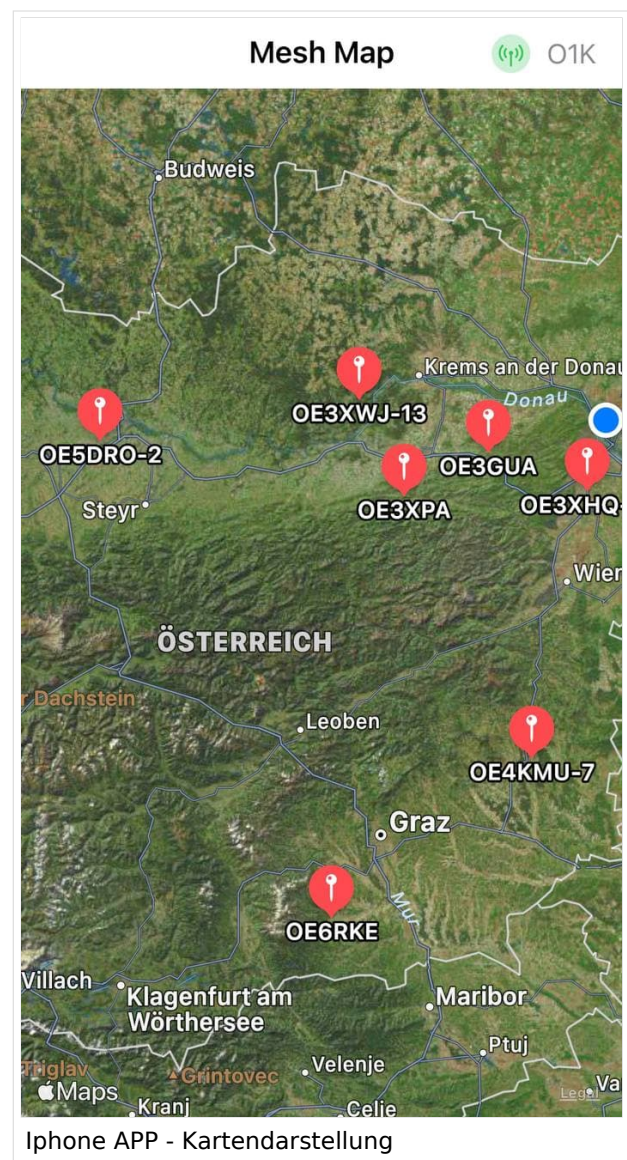
Die Positionsmeldungen werden automatisch gesendet jedoch Textmeldungen benötigen einen Client.

Es gibt folgende Clients:

Kommando-Zeile aber nur zum Senden von Text

ANDROID APP iPhone APP

<https://icssw.org/meshcom-app/>



Iphone APP - Kartendarstellung

Derzeit gibt es LoRa-Getways in OE und DL:

- OE6 Deutschlandsberg welcher bis nach Graz reicht
- OE4 Allhau bei Oberwart
- OE3 ÖVSV-HQ Wr. Neudorf
- OE3 Klosterneuburg

- OE3 Alt-Erlaa
- OE1 Wien 15
- OE3 Sieghartskirchen (Tullnerfeld)
- OE3 Jauerling bei Stein/Donau
- OE5 Linz Lichtenberg und weitere Standorte nahe Linz
- OE7 nahe Innsbruck
- DL Süd/Bayern Waging
- DL West
- Weitere kommen rasch dazu. Info via Dashboard.

73 de Kurt

OE1KBC

Nat. & Int. Projekte im ÖVSV