
Inhaltsverzeichnis

--

MeshCom/MeshCom 2.0

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 10. Juni 2022, 08:36 Uhr (Quelltext anzeigen)
Oe1kbc (Diskussion | Beiträge)
K
Markierung: Visuelle Bearbeitung
← Zum vorherigen Versionsunterschied

Aktuelle Version vom 18. März 2024, 15:42 Uhr (Quelltext anzeigen)
Oe1kbc (Diskussion | Beiträge)
Markierung: Visuelle Bearbeitung

(12 dazwischenliegende Versionen von 3 Benutzern werden nicht angezeigt)

Zeile 1:	Zeile 1:
- == MeshCom 2.0 ==	+ == MeshCom 4.0 ==
- Grundlegende Spezifikationen	
- Luftschnittstelle	+ ==== BETA-Test Dokumentation und Anleitungen ====
- AFU kompatibel der Source, Node, Gateway, Destination Kennung als Rufzeichen	+ ====https://icssw.org/meshcom-4-0-installation/<nowiki/>====
- Path-Kontrollstruktur (nur für Testzwecke)	+ ===== Grundlegende Spezifikationen =====
- Struktur der Payload in die Struktur der Meldung eingebettet	+ * "'Luftschnittstelle'"
	+ ** Mesh Netzwerk - selbst bildend und selbst heilend
	+ ** AFU kompatibel der Source, Node, Gateway, Destination Kennung als Rufzeichen
	+ ** Path-Kontrollstruktur (nur für Testzwecke)
	+ ** Struktur der Payload in die Struktur der Meldung eingebettet

- + **** Zusätzlich zur Übertragungs-Sicherung durch die Hardware sind CRC und FEC in der Struktur der Meldung einzuplanen**
- + **** Meldung und Payload komprimiert übertragen**
- + **** Node, Digipeater-only, Gateway-only, Point-to-Point (Netzerweiterungen)**
- + **** unverschlüsselt**
- + **** Adress-Header (FromCALL, ToCALL, VIA) komprimiert und mit CRC (kompatibel zu AX25v2)**
- + **** Nachrichten Priorisierung**
- + *** "'Gateway-Schnittstelle'"**
- + **** MQTT-Protokoll mit üblicher Feldstruktur aufbauen**
- + **** UDP-Übertragung**
- + **** Heartbeat zur Client/Server-ONLINE Erkennung**
- + **** Tiefe der Meldung vom und zum Gateway einstellbar (Test- und Entwicklungs-Erleichterung)**
- + **** Nach Neustart eines Gateways automatischer Übertragung von Grunddaten wie:**
- + ***** aktive NODES**
- + ***** letzte Meldungen**
- + ***** Anstoßen der Store & Forward Meldungen**
- + *** "'Modul-Schnittstellen'"**
- + **** Serial via USB**
- + **** GPIO für externe Hardware und Steuerungen**
- + **** GPS intern, extern, fix**
- + **** WiFi**

- + ***** Userschnittstelle**
- + ***** Gateway-Schnittstelle**
- + **** Bluetooth**
- + ***** APP-Schnittstelle**
- + **** ETH-Schnittstelle optional**
- + *** "'Meldungs-Grundtypen'"**
- + **** Broadcast**
- + **** Group Call**
- + **** Private Call**
- + **** Store & Forward**
- + **** Entwicklungs- und Debug-Meldungen**
- + *** "'Offene Hardware'"**
- + **** Die Verwendung der kompatibler MCU sollte eingehalten werden**
- + **** ESP32**
- + **** Fertigmodule MCU, HF, GPS gemeinsam**
- + **** wie TTGO, TLORA, HELTEC, ...**
- + **** Bevorzugterweise Aufbau Basisplatine, Steckmodule**
- + **** wie RAK WisBlock**
- + **** Vorhandene Hardware aus dem LoRa-APRS Projekt**
- + **** Semtech SX1262 LoRa-Transceiver oder kompatibel**
- + **** ETH-Modulblock mit IP-Stack für Gateways**
- + *** "'Firmware'"**
- + **** Grundstruktur für Entwicklung in der Gruppe vorbereitet**
- + **** Leicht zu erweitern, pflegen**
- + **** Klare Funktionsgliederung**

- + **** Keine direkte Hardware-Bezogenheit in der Logik-Struktur**
- + **** Logik-Struktur mit klaren Schnittstellen aufgebaut um funktionelle Erweiterungen jederzeit einzubauen ohne die getestete Basisfunktionalität zu beeinflussen**
- + *** "'Welche Service bietet MeshCom 4.0 an?'"**
- + **** Textübertragung**
- + **** Positionsübertragung (Smart Beaconsing)**
- + **** Frei definierbare Payload**
- + *** "'Feature-List'"**
- + **** Konfiguration über USB-Serial-Schnittstelle**
- + **** Rufzeichen mit APRS-konformen SSID**
- + **** Frequenzeinstellung und Anzeige**
- + **** Feldstärkeanzeige (S-Meter, RSSI, MER)**
- + **** LoRa-Modulationsparameter auch detailliert**
- + **** Fix-Position**
- + **** Batterie-Management Stufen**
- + **** Scannen nach verfügbarem MeshCom-Channel**
- + *** "'Use Cases'"**
- + **** allg. Amateurfunknachrichtendienst**
- + **** Not-Katfunk**
- + **** Infodienste**
- + ***** Wetterbericht**
- + ***** SolarFlux**
- + ***** Radioaktivität**

		+ *** Blitzortung	
		+ *** DXCluster	
		+ *** Phonie-Skeds, SOTA-Skeds	
-	Zusätzlich zur Übertragungs-Sicherung durch die Hardware sind CRC und FEC in der Struktur der Meldung einzuplanen		
-	Meldung und Payload komprimiert übertragen	+ Entwurf: Kurt OE1KBC	
-	Node, Digipeater-only, Gateway-only, Point-to-Point (Netzerweiterungen)	+ Diskussion: matrix.oevsv.at Raum https://matrix.to/#/#meshcom:matrix.oevsv.at	
-	Gateway-Schnittstelle		
-			
-	MQTT-Protokoll mit üblicher Feldstruktur aufbauen		
-			
-	UDP-Übertragung		
-			
-	Hardbeat zur Partner-ONLINE Erkennung		
-			
-	Tiefe der Meldung vom und zum Gateway einstellbar (Test- und Entwicklungs-Erleichterung)		
-			
-	Nach neustart eines Gateways automatischer Übertragung von Grunddaten wie aktive NODES, Letzter Meldungs-ID Stack, ...		
-			

- Modul-Schnittstellen
-
- Serial via USB
-
- GPIO für externe Hardware und Steuerungen
-
- GPS intern, extern, fix
-
- WiFi
-
- Userschnittstelle
-
- Gateway-Schnittstelle
-
- Bluetooth
-
- APP-Schnittstelle
-
- ETH-Schnittstelle optional
-
- Meldungs-Grundtypen
-
- Broadcast
-
- Group Call
-
- Private Call
-
- Store & Forward
-

-
- **Entwicklungs- und Debug-Meldungen**
-
- **Offene Hardware**
-
- **Die Verwendung der kompatibler MCU sollte eingehalten werden**
-
- **ESP32**
-
- **Fertigmodule MCU, HF, GPS gemeinsam**
-
- **wie TTGO, TLORA, HELTEC, ...**
-
- **Bevorzugterweise Aufbau Basisplatine, Steckmodule**
-
- **wie RAK WisBlock**
-
- **Vorhandene Hardware aus dem LoRa-APRS Projekt**
-
- **Semtech SX1262 LoRa-Transceiver oder kompatibel**
-
- **ETH-Modulblock mit IP-Stack für Gateways**
-
- **Firmware**
-
- **Grundstruktur für Entwicklung in der Gruppe vorbereitet**

-
- **Leicht zu erweitern, pflegen**
-
- **Klare Funktionsgliederung**
-
- **Keine direkte Hardware-Bezogenheit in der Logik-Struktur**
-
- **Logik-Struktur mit klaren Schnittstellen aufgebaut um funktionelle Erweiterungen jederzeit einzubauen ohne die getestete Basisfunktionalität zu beeinflussen**
-
- **Welche Service bietet MeshCom 2.0 an?**
-
- **Textübertragung**
-
- **Positionsübertragung (Smart Beaconsing)**
-
- **Frei definierbare Payload**
-
- **Feature-List**
-
- **Konfiguration über USB-Serial-Schnittstelle**
-
- **Rufzeichen**
-
- **Frequenz**
-

–	LoRa-Modulationsparameter auch detailliert
–	
–	Fix-Position
–	
–	Batterie-Management Stufen
–	
–	Use Cases
–	
–	__HIDETITLE__
	__KEIN_INHALTSVERZEICHNIS__

__KEIN_INHALTSVERZEICHNIS__

Aktuelle Version vom 18. März 2024, 15:42 Uhr

MeshCom 4\0

BETA\Test Dokumentation und Anleitungen

<https://icssw.org/meshcom-4-0-installation/>

Grundlegende Spezifikationen

- **Luftschnittstelle**
 - Mesh Netzwerk - selbst bildend und selbst heilend
 - AFU kompatibel der Source, Node, Gateway, Destination Kennung als Rufzeichen
 - Path-Kontrollstruktur (nur für Testzwecke)
 - Struktur der Payload in die Struktur der Meldung eingebettet
 - Zusätzlich zur Übertragungs-Sicherung durch die Hardware sind CRC und FEC in der Struktur der Meldung einzuplanen
 - Meldung und Payload komprimiert übertragen
 - Node, Digipeater-only, Gateway-only, Point-to-Point (Netzerweiterungen)
 - unverschlüsselt
 - Adress-Header (FromCALL, ToCALL, VIA) komprimiert und mit CRC (kompatibel zu AX25v2)
 - Nachrichten Priorisierung
- **Gateway-Schnittstelle**
 - MQTT-Protokoll mit üblicher Feldstruktur aufbauen
 - UDP-Übertragung
 - Heartbeat zur Client/Server-ONLINE Erkennung
 - Tiefe der Meldung vom und zum Gateway einstellbar (Test- und Entwicklungs-Erleichterung)

-
- Nach Neustart eines Gateways automatischer Übertragung von Grunddaten wie:
 - aktive NODES
 - letzte Meldungen
 - Anstoßen der Store & Forward Meldungen
 - **Modul-Schnittstellen**
 - Serial via USB
 - GPIO für externe Hardware und Steuerungen
 - GPS intern, extern, fix
 - WiFi
 - Userschnittstelle
 - Gateway-Schnittstelle
 - Bluetooth
 - APP-Schnittstelle
 - ETH-Schnittstelle optional
 - **Meldungs-Grundtypen**
 - Broadcast
 - Group Call
 - Private Call
 - Store & Forward
 - Entwicklungs- und Debug-Meldungen
 - **Offene Hardware**
 - Die Verwendung der kompatibler MCU sollte eingehalten werden
 - ESP32
 - Fertigmodule MCU, HF, GPS gemeinsam
 - wie TTGO, TLORA, HELTEC, ...
 - Bevorzugterweise Aufbau Basisplatine, Steckmodule
 - wie RAK WisBlock
 - Vorhandene Hardware aus dem LoRa-APRS Projekt
 - Semtech SX1262 LoRa-Transceiver oder kompatibel
 - ETH-Modulblock mit IP-Stack für Gateways
 - **Firmware**
 - Grundstruktur für Entwicklung in der Gruppe vorbereitet
 - Leicht zu erweitern, pflegen
 - Klare Funktionsgliederung
 - Keine direkte Hardware-Bezogenheit in der Logik-Struktur
 - Logik-Struktur mit klaren Schnittstellen aufgebaut um funktionelle Erweiterungen jederzeit einzubauen ohne die getestete Basisfunktionalität zu beeinflussen
 - **Welche Service bietet MeshCom 4.0 an?**
 - Textübertragung
 - Positionsübertragung (Smart Beaconing)
 - Frei definierbare Payload
 - **Feature-List**
 - Konfiguration über USB-Serial-Schnittstelle
 - Rufzeichen mit APRS-konformen SSID
 - Frequenzeinstellung und Anzeige
 - Feldstärkeanzeige (S-Meter, RSSI, MER)

-
- LoRa-Modulationsparameter auch detailliert
 - Fix-Position
 - Batterie-Management Stufen
 - Scannen nach verfügbarem MeshCom-Channel
 - **Use Cases**
 - allg. Amateurfunknachrichtendienst
 - Not-Katfunk
 - Infodienste
 - Wetterbericht
 - SolarFlux
 - Radioaktivität
 - Blitzortung
 - DXCluster
 - Phonie-Skeds, SOTA-Skeds

Entwurf: Kurt OE1KBC

Diskussion: matrix.oevsv.at Raum <https://matrix.to/#/#meshcom:matrix.oevsv.at>