

LoRa – FTD

*Long-Range Field Test Device
für Versuchs- und Messzwecke*

DL2MEE (Klaus Buchenrieder)

Version: FA_1.0-27.01.23 15:34:00

INHALTSVERZEICHNIS

Hinweise	6
Haftungsausschluss	7
Einführung	8
LoRa™ Grundlagen	9
<i>Chirp Spread Spektrum (CSS) Modulation:.....</i>	<i>9</i>
<i>LoRa™ Long Range Chirp Spektrum Modulation:.....</i>	<i>10</i>
Chirp:	10
Chips:	11
Kodierung von Datenpaketen:	12
Präambel	13
Header.....	13
Nutzdatenblock.....	13
Redundanzen und Coding Rate.....	14
Inbetriebnahme des LoRa™-FTD.....	14
<i>Treiber für den USB-Chip CP2102:.....</i>	<i>15</i>
<i>esptool:.....</i>	<i>15</i>
Installieren des esptool mit Windows:.....	15
Installieren des esptool mit anderen Betriebssystemen:	16

<i>Firmware auf das Heltec Board laden:</i>	<i>17</i>
Generische Kommandozeile:	17
MacOS:.....	17
Windows:.....	17
<i>Terminalprogramm (Konsole):.....</i>	<i>18</i>
Windows:.....	18
Mac OS:.....	19
Android:	20
IOS:	20
<i>Eingabe des Rufzeichens nach einem Firmware Update: 20</i>	
Methode A: Eingabe des Rufzeichens über die Konsole:	20
Methode B: Eingabe des Rufzeichens über eine Bluetooth App:	21
Neustart nach dem Eintrag des Rufzeichens:.....	23
Betrieb von mehr als einem LoRa™-FTD unter einem Rufzeichen:.....	24
<i>Firmware des Heltec Board auslesen und sichern:</i>	<i>24</i>
Steuerkommandos.....	26
<i>Hochfrequenzparameter</i>	<i>27</i>
[#F] Frequenz ändern.....	28
[#G] LNB-Verstärkung (Dämpfung) ändern.....	28
[#L] Sendeleistung ändern	29

<i>LoRa™-Parameter</i>	29	[RECOFF] LoRa Nachrichten protokollieren - AUS	41
[#BW] Bandbreite ändern	30	Namenskonvention SPIFFS < filename>:	41
[#CR] Kodierungsrate ändern	30	[#RM] Remove File – SPIFFS	42
[#CRC] Cyclic Redundancy Check	31	[#X] Startup Kommando festlegen	42
[#IQ] LoRa Modulation Polarity Inversion	32	[#RES] Reset	43
[#SF] Spreizfaktor ändern	33	[-] Nil	43
[#PA] Präambellänge ändern	33	[#PAR] Alle Parameter Anzeigen	43
[#SY] Synchronisationszeichen ändern	34	[#PL] LoRa Parameter Anzeigen	44
<i>Kommunikationsanweisungen</i>	34	[#RT] Rufzeichen temporär ändern	44
[.] Nachricht mit Rufzeichen senden	34	[#RP] Rufzeichen permanent ändern	44
[:] Nachricht ohne Rufzeichen senden	35	<i>Serviceanweisungen</i>	45
[!] Übungstext in CW ausgeben	35	[#*1] Namespace Inhalt ausgeben	45
Systemanweisungen	36	[#*13] Doomsday – Alles Löschen	45
[#CAT] Catenate – SPIFFS	36	[#*32] Device ID (MAC) ausgeben	45
[#H] Helptext ausgeben	36	[#*60] Inhalt der LoRa™ Chip Register ausgeben	46
[#HELP] Helptext ausgeben	37	[#*77] Inhalt des SPIFFS ausgeben	46
[#HILFE] Helptext ausgeben	37	[#*99] Namespace Inhalt löschen	46
[#HEX] Empfangene Zeichen in HEX darstellen	37	Applikationen	47
[#LS] List Space – SPIFFS Dir	38	<i>LoRa™-Bake</i>	47
[#ME] Monitor einschalten	38	[#BAKE] Starten der Bakenapplikation	48
[#MA] Monitor ausschalten	38	[#BE] Beenden der Bakenapplikation	48
[#MR] Memory Read Parameters	38	[#BI] Einstellung des Bakenintervalls	48
[#MW] Memory Write Parameters	39	[#BT] Einstellung des Bakentexts	49
[#QC] Quick Charge Mode	39	<i>LoRa™-Repeater</i>	49
[RECON] LoRa™ Nachrichten protokollieren - EIN	39		

Monitoring empfangener Pakete	50	[pos] Aktuelle Position ausgeben.....	63
Parameter Syntax.....	50	[put] Position senden	63
[#REP] Starten der Repeaterapplikation	50	[ssid] APRS SSID setzen	64
[-C] Repeaterkennung	51	APRS SSID Liste.....	64
[-R] Signal RSSI	51	[str] APRS String-Comment setzen.....	65
[-P] Paket-RSSI	52	<i>Morsetrainer</i>	66
[-S] Packet SNR.....	53	[A] CW Abkürzungen	67
[-F] Frequency Error	53	[D] CW Abkürzungen	67
[-A] Alle Messwerte senden.....	54	[F] CW Fünfergruppen.....	67
[#REE] Beenden der Repeaterapplikation.....	54	[FA] CW Fünfergruppen - angepasste Übungen	68
<i>LoRa™-Monitor Dual-Watch</i>	55	[FA ★] Fünfergruppen Hilfetext ausgeben.....	68
[#MONI] Starten der Monitor Applikation	56	[FA A] Alphanummerische Fünfergruppen	69
[FA] Frequenz FA einstellen.....	56	[FA B] Buchstaben Fünfergruppen	69
[FB] Frequenz FB einstellen.....	57	[FA Z] Ziffern Fünfergruppen.....	69
[ACA] Acronym für FA eingeben.....	57	[FA I] AFu Interpunktion Fünfergruppen.....	70
[ACB] Acronym für FB eingeben	57	[FA S] Sonderzeichen Fünfergruppen	70
<i>LoRa™-APRS Testset</i>	58	[FA X] Extended Fünfergruppen	70
APRS Parameter	58	[L] CW Lautstärke einstellen.....	71
Beispiel APRS Test Paket.....	59	[M] CW Mixed.....	71
Koordinatenbestimmung Ui-View.....	59	[P] CW Parameter ausgeben.....	72
[anz] Anzinger Forst	59	[S] CW Farnsworth Spread setzen.....	72
[def] Default APRS HF-Parameter setzen	60	[W] CW Geschwindigkeit in wpm setzen.....	73
[h] Hilfetext für die APRS Parameter.....	60	[B] CW Geschwindigkeit in BpM setzen	73
[sym] APRS Kartensymbol setzen.....	61	[] CW Kommandos Anzeigen	73
[lat ggmN.mmN S] Latitude.....	62	Hardware	74
[lon gggmm.mmE W] Longitude	63		

<i>Batteriemanagement und Anschluss des Akku:</i>	74	<i>Beispielprogramm: „Drehenkoder Test“</i>	94
<i>Probleme mit Power Banks:</i>	75	Pinout des Heltec Modul	97
<i>Brownout Detection:</i>	75	Geplante Weiterentwicklungen	98
Gehäuse, Aufbau und Erweiterungen	76	Danksagung	99
<i>Gehäuse</i>	76	Anhang: Telegrafie – Afu Abkürzungen	99
<i>Aufbau</i>	77	Abbildungsverzeichnis	104
<i>Grundverdrahtung</i>	79	Referenzen	106
<i>Erweiterung für CW</i>	82		
<i>Erweiterung mit einem Drehenkoder</i>	85		
Bauteileliste und Bezugsquellen	89		
Speichernutzung des LoRa™-FTD	90		
Memory Map des LoRa™- FTD	91		
Programmierung des Heltec Moduls	92		
<i>Nützliche Bibliotheken:</i>	92		
LoRa.h	92		
U8x8lib.h	92		
<i>Einstellungen in der Arduino IDE:</i>	93		
<i>Beispielprogramm „Blink für ESP32“</i>	94		

HINWEISE

Die im Download Bereich der Zeitschrift Funkamateure oder von mir selbst zur Verfügung gestellte Firmware für das LoRa™ Field Test Device (FTD) wurde von mir, Klaus Buchenrieder (DL2MEE), programmiert. Eine Weiterentwicklung ist geplant hängt jedoch vom verfügbaren Speicherplatz, den technischen Möglichkeiten des Moduls und meiner hierfür verfügbaren Zeit ab. Ein Anspruch darauf besteht nicht!

Ein Verkauf der Software ist nicht gestattet – die bereitgestellte Software/Firmware mit Versionsnummer beginnend mit „FA“ sind für private Verwendung durch Funkamateure kostenlos.

Es ist auch generell nicht gestattet geänderte oder erweiterte Versionen in jedweder Form zu vermarkten, diese weiterzugeben oder als Beigabe zu einem Verkaufsartikel bzw. als Verkaufsanzreiz zu verwenden.

Zur Programmierung wurden zwei externe Bibliotheken verwendet:

(1) Für die graphische und textuelle Ausgabe auf dem monochromen Display die U8g2 Bibliothek von Oli Kraus [OLIKRAUS] und (2) zum Senden und Empfangen

vonLoRa™-Paketdaten und zur Übergabe von Parametern auf die modifizierten Versionen der Arduino-LoRa Bibliothek von Sandeep Mistry [SAND2020].

Die Bibliothek von Oli Kraus (1) unterliegt der Lizenz: <https://github.com/olikraus/u8g2/blob/master/LICENSE> und die Bibliothek von Sandeep Mistry (2) unterliegt der MIT Lizenz: <https://github.com/sandeepmistry/arduino-LoRa/blob/master/LICENSE> (Stand 21. Juni 2022).

Das Gehäuse ist eine Weiterentwicklung des ursprünglichen Entwurfs von Makoto Schoppert für das Heltec WiFi LoRa™ Board, welches er seit Januar 2021 in thingiverse.com frei zum Download zur Verfügung stellt [2022Thingiverse]. Die Nutzerlizenz zum Originalentwurf findet man unter: <https://creativecommons.org/licenses/by-nc-sa/4.0/> (Stand 19. Mai 2022).

Semtech, das Semtech-Logo und LoRa™ bzw. LoRa sind eingetragene Marken oder Dienstleistungsmarken der Semtech Corporation oder ihrer Tochtergesellschaften.

Heltec, das Heltec-Logo sind eingetragene Marken der Chengdu Heltec Automation Technology Co., ltd. und deren Tochtergesellschaften.

HAFTUNGSAUSSCHLUSS

Für Schäden jeglicher Art die aus der Verwendung der bereitgestellten Unterlagen und das Field Test Device kurz auch FTD genannt, entstehen, übernimmt Klaus Buchenrieder (DL2MEE) keine Haftung und keine Verantwortung.

Die Verwendung des FTD geschieht auf eigene Verantwortung des Nutzers. Es handelt sich bei dem beschriebenen FTD Gerät um ein Funkgerät, welches ausschließlich von Funkamateuren mit einem gültigen Amateurfunkzeugnis (Amateurfunklizenz) betrieben werden darf. Es wird ausdrücklich auf die Einhaltung der entsprechenden Gesetze und Verordnungen zum Betrieb von Funkanlagen und zum Amateurfunk und die Einhaltung der Bandpläne hingewiesen.

Klaus Buchenrieder (DL2MEE) übernimmt keine Haftung für Schäden jeglicher Art, die durch die Nutzung des Geräts, das Herunterladen der Vorlagen und Beschreibungen oder durch die Nutzung von Tools und Software Dritter entstehen. Ebenfalls haftet er nicht für unmittelbare oder mittelbare Folgeschäden, Datenverlust, entgangenen Gewinn, System- oder Produktionsausfälle. Auch übernimmt der er keine Haftung für

jegliche finanzielle Schäden. Auch für Fehler in dieser Unterlage und unvollständige Beschreibungen in assoziierten Unterlagen wird keine Haftung übernommen.

Auf die Rechte Anderer an der Software insbesondere auf die Nutzung von Bibliotheken und Entwurfsvorlagen wird in diesem Schriftstück an geeigneter Stelle (siehe z.B.: auch Seite 6) hingewiesen.

Bei der Verwendung programmierter Tools und Zusatzsoftware, z.B.: Arduino IDE, Tools der Firma Heltec, von Silicon Labs und Bedienungsapplikationen, gelten die entsprechenden Lizenzen und Rechte.

EINFÜHRUNG

Das LoRa™ Field Test Device (LoRa-FTD) ist ein kompakter Portabeltransceiver der speziell für Versuche und Messzwecke mit der Modulationsart LoRa™ entwickelt wurde. Das Gerät ist mit dem hochintegrierten IoT-Entwicklungsboard ESP32 LoRa™ V.2, der Firma Heltec aufgebaut. Dieses Modul wurde ursprünglich für IoT (Internet-of-Things) Anwendungen entwickelt. Das Modul ist für Amateurfunkanwendungen sehr interessant, da es sehr preiswert ist und genügend Speicher für zukünftige Softwareentwicklungen besitzt und Hardwareerweiterungen über Ports möglich sind. In der gegenwärtigen Ausbaustufe können amLoRa™-FTD bereits alle LoRa™-Betriebsparameter, mit Ausnahme des CRC (Cyclic-Redundancy-Check), eingestellt werden. Auch ist Sende- und Empfangsbetrieb im 70 cm und eingeschränkt auch im 2 m Band möglich. Weiterhin besteht die Möglichkeit ein parametrisiertes Bakenignal zu erzeugen. Die Bedienung desLoRa™-FTD erfolgt entweder über eine serielle USB-Schnittstelle oder über ein Bluetooth Gerät wie z.B. ein Telefon mit einer Bluetooth-Seriell Client Applikation. Der Verbindungsstatus zum Bluetooth Gerät und auch empfangene Pakete können

auf dem SSD1306 kompatiblen OLED Display mit 0,96 Zoll Diagonale direkt dargestellt werden. Genauso wie die vorhandene Wifi Schnittstelle, der optional anschließbare Drehimpulsgeber mit 24 Rastungen und Taster erlauben zukünftige Erweiterungen der Funktion des Geräts. Durch die Verwendung des Heltec ESP32 LoRa™ Moduls besitzt das Gerät auch ein integriertes Batteriemanagement für 3,7 Volt LiPo/Li-Ion Akkus mit einer Kapazität von ca. 2100 mAh, die direkt über die USB-Buchse geladen werden können. Da für das Lademanagement auf dem Heltec Board der Batterielade-IC TP4054 verwendet wird, kann der Ladestrom durch Verändern eines einzigen Widerstands an eine andere Kapazität des Akkus angepasst werden. Das Heltec ESP32 LoRa™ Modul lässt sich für eigene Projekte sowohl mit der Arduino IDE (Integrated Developer Environment) als auch mit anderen IDEs programmieren. Zum Aufspielen derLoRa™-FTD Firmware verwendet man das von Heltec verfügbare *esptool*. Im Folgenden sind nach den Grundlagen, alle Schritte zur Installation und zum Betrieb desLoRa™-FTD erläutert.

Die Chirp- oder lineare Frequenzmodulation wurde bereits in den 1940er Jahren von Erich Hüttmann zur Entfernungsmessung [1] erfunden und von Sidney Darlington weiterentwickelt. Bereits 1947 wurde dieses Modulationsverfahren für ein Pulse Compression Radar verwendet. Die Vorteile der Chirp Spread Spectrum Modulation (CSS) wurden in zahlreichen Untersuchungen herausgearbeitet und die Anwendungen waren bis vor einigen Jahren fast ausschließlich in der militärischen Datenkommunikation. Heute wird CSS hauptsächlich zur drahtlosen Übertragung von Datenpaketen auf kurzen Distanzen mit kleinen Leistungen in den ISM Frequenzbereichen eingesetzt. Dabei hält man sich vorrangig an die Vorschläge und Normungen nach Standard IEEE 802.15.4a [2] für das sog. Wireless Personal Area Network und die zugehörige Mesh-Technologie.

Generell werden CSS Verfahren immer dann eingesetzt, wenn die Signalleistung unter die Rauschgrenze abfällt oder Daten nahe der Rauschgrenze übertragen werden müssen. Durch eine Bandbreitenanpassung wird eine zuverlässige Kommunikation im Bereich des Grundrauschens gesichert.

Chirp Spread Spektrum (CSS) Modulation:

In der Informationstheorie gibt das Theorem von Claude Elwood Shannon und Ralph Hartley an, wieviel Information mit einer Signalstärke S pro Zeiteinheit beim Vorhandensein von Rauschen N über einen Kommunikationskanal mit einer bestimmten Bandbreite BW übertragen werden kann. Direkt aus diesem Shannon-Hartley-Theorem folgt für einen bekannten Signal-Rauschabstand $\frac{S}{N}$ die maximale Kanalkapazität $C = BW * \log_2 \left(1 + \frac{S}{N} \right)$. Ist das Rauschen des Übertragungskanals und die Signalstärke hinreichend konstant, so hängt die Kanalkapazität daher nur noch von der Signalbandbreite ab, $\frac{C}{BW} \approx \frac{S}{N}$.

Diese Erkenntnis macht man sich bei CSS und damit auch bei LoRa™ zu Nutze indem man eine ausreichend große Bandbreite wählt und dann nur eine sehr kleine Leistung braucht um ein Datenpaket fehlerfrei zu übertragen. Durch die Wahl des „richtigen“ Chirps ergibt sich weiterhin eine gute Robustheit gegenüber Mehrweg-

¹ Die Langform dieses Kapitels ist im November 2021 in der Zeitschrift FUNKAMATEUR erschienen.

Fading-, Doppler- und In-Band-Störsignalen was für Industrial-, Scientific- und Medical-Anwendungen auf den ISM Frequenzen sehr vorteilhaft ist.

Die mit Chirp Spread Spectrum (CSS) überbrückbaren Entfernungen sind, verglichen mit den traditionellen Modulationsarten CW, AM, FM, SSB, PCM, OOK, FSK usw., viel größer und bei direkter Sichtverbindung lassen sich von jedem Funkamateurl mit 10 mW und einfachen $\frac{\lambda}{4}$ Antennen, in wenigen Metern über Grund, über 10 Kilometer überbrücken. Unser terrestrischer Rekord von C11 an der Universität der Bundeswehr München liegt derzeit noch bei 26 km zwischen einem selbstgebauten Handsender ($\frac{\lambda}{4}$ Strahler, 18 mW) und dem LoRa™-Digipeater der Klubstation DB0UFO mit einer Diamond X-50 in 24,5 m über Grund. Den Weltrekord hält derzeit Thomas Telkamp, der mit einem Wetterballon in 38 km Höhe und 25 mW EIRP, Datenpakete über etwa 832 km zum Berg Radhorst in der Tschechischen Republik übertragen konnte. Im Satellitenbetrieb ist momentan Andreas Malek, OE3DMB, der Rekordhalter. Ihm gelang es auf 2,4 GHz mit 20W und einer kleinen WLAN Antenne auf dem Magnetfuß am Autodach 71572 km mit LoRa™ über den QO-100 einen LoRa™-APRS Track zu machen.

LoRa™ Long Range Chirp Spektrum Modulation:

LoRa™ ist ein eingetragener Handelsname der Firma Semtec und steht für den Begriff „Long Range“. LoRa™ Radio Technology ist konform mit dem IEEE Standard 802.15.4 und verwendet eine modifizierte Form der Chirp-Spread-Spectrum (CSS)-Modulation. Hierbei wird die Trägerfrequenz einer Sinuskurve entweder mit aufsteigender oder mit abfallender Frequenz variiert. Die entstehenden Signale, sind den Tönen, die z.B. Fledermäuse und Delfine zur Entfernungsbestimmung und Kommunikation nutzen, sehr ähnlich. Man bezeichnet diese „Pfeiftöne“ als Chirp-Impulse oder kurz Chirps.

Chirp:

Ein LoRa™-Chirp [3] hat eine feste zeitliche Länge, während der sich nur die Frequenz zwischen einer Anfangs- (f_{low}) und Endfrequenz (f_{high}) linear ändert. Die Amplitude des Sinussignals bleibt dabei aber konstant. Wie Bild 1 zeigt, unterscheidet man, entsprechend der Frequenzänderung, einen Up- und einen Down-Chirp.

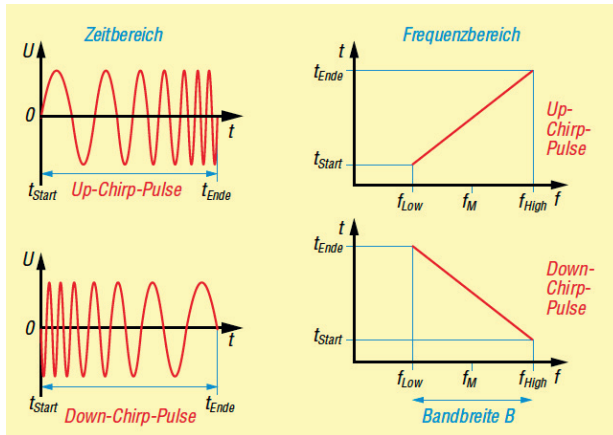


Bild 1: Up- und Down-Chirp Puls im Zeit- und Frequenzbereich [Quelle: FA 11/21 Seite 868].

Offensichtlich ist ein LoRa™ Chirp eindeutig durch die Mittenfrequenz (f_{center}), die Bandbreite (BW) und die Länge des Chirps ($t_{pulse} = t_{end} - t_{start}$) definiert.

Aus Mittenfrequenz und Bandbreite ergibt sich:

$$f_{low} = f_{center} - \frac{BW}{2}; \text{ und } f_{high} = f_{center} + \frac{BW}{2};$$

Chips:

Zur Übertragung von Datenworten unterteilt man einen Chirp in 2^{SF} Zeitscheiben die als Chips bezeichnet werden. Die Variable SF im Exponenten bezeichnet man als Spreading-Faktor. Mit dem Spreading-Faktor (SF) legt man die Anzahl der Bit fest, die mit einem Symbol kodiert werden können. Um einen Dezimalwert m zu übertragen, benötigt man entsprechend $\lceil \log_2 \cdot (m) \rceil$ Bit. Beispielsweise sind für den Dezimalwert 94 gleich 7 Bit notwendig. Bild 2 zeigt die Kodierung des Binärwerts 1011110

(\triangleq 94dezimal), als Up-Chirp mit dem Spreading-Faktor 7. Die Symboldauer entspricht 2^{SF} gleich 128 Chips. Bei der Übertragung beginnt das Symbol mit dem Up-Chirp, nicht bei der Frequenz f_{low} sondern erst, etwa in der Mitte, bei der Frequenz:

$$f_{start} = f_{low} + d \cdot \frac{f_{high} - f_{low}}{2^{SF}}; \text{ mit } d = 94 \text{ und } SF = 7.$$

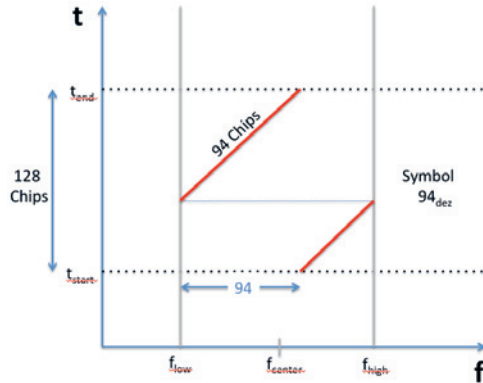


Bild 2: Up-Chirp für ein Symbol mit dem Dezimalwert 94.

Nach dem Erreichen der oberen Endfrequenz f_{high} des Up-Sweeps, folgt sogleich der Frequenzsprung zu f_{low} und folgend der Rest des Up-Sweeps mit den restlichen Chips. Der Symbolwert ist daher zweimal in einem Chirp kodiert. Beim Empfang nutzt man den dabei entstehenden Frequenzsprung, da der Symbolwert gleich der Verschiebung der Frequenz am Anfang sowie auch proportional zu der Zeit zwischen dem Frequenzsprung und dem Ende des Signals ist. Eine leichte Abweichung von der Empfangsmittenfrequenz spielt daher kaum eine Rolle.

Kodierung von Datenpaketen:

Bei der LoRa™-Datenübertragung unterscheidet man das explizite- und das implizite Paketformat bei dem der Header verkürzt und damit die Länge der Nutzdaten, die Cyclic Redundancy Checking (CRC) Information und die Kodierungsrate auf der Sende- und Empfangsseite fest voreingestellt werden müssen. Im Amateurfunk verwenden wir aus Flexibilitätsgründen ausschließlich das explizite Format, weil die Länge des Nutzdatenblocks variabel und oft nicht strikt auf 255 Zeichen im Datenpaket begrenzt ist.

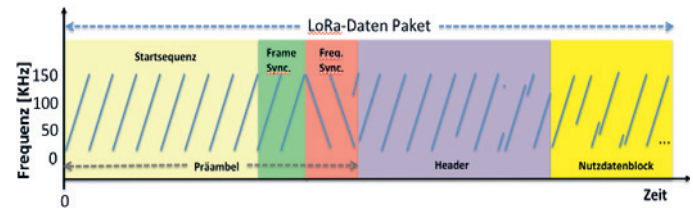


Bild 3: Exemplarisches LoRa™ Spektrogramm für eine Übertragung mit explizitem Paketformat.

Ein LoRa™-Datenpaket besteht aus drei Blöcken: der Präambel, dem Header und dem Nutzdatenblock. Bild 3 zeigt das schematische Spektrogramm für ein Datenpaket mit einem Spreading-Factor 11 und der Coding-Rate 5 [4].

Präambel

Die Präambel ist zur Zeit- und Frequenzsynchronisation des Empfängers notwendig. Sender und Empfänger müssen daher auf die gleiche Präambellänge eingestellt sein. Die Präambel besteht, im gezeigten Beispiel, aus acht nichtmodulierten Up-Chirps, gefolgt von zwei modulierten Frame-Synchronisation Up-Chirps, deren codierter Wert zur Netzwerk Identifikation verwendet werden kann. Die Identifikation verwendet man um Datenpakete zu verwerfen die nicht für den Empfänger oder für die Empfängergruppe bestimmt sind. Dies ist ein ganz ähnliches Konzept wie bei Color Codes im DMR oder bei Subtönen von FM Relaisstationen. Die direkt anschließenden beiden Down-Chirps, gefolgt von dem ein viertel langem Up- (bzw. Down-) Chirp dienen der exakten Frequenzsynchronisation und zur Bestimmung der Signalstärke am Empfangsort. Nach dem Abschluss der

Synchronisationsphase wird ein kurzer Header übertragen.

Header

Dieser enthält die Information über die Länge der Nutzlast, die verwendete Datenrate, ein Bit, welches das Vorhandensein einer zyklischen Redundanzprüfung (CRC) der Nutzlast anzeigt und eine 1-Byte-Header-Prüfsumme aufweist. Nach der Headerinformation schließt direkt die Übertragung der Nutzdaten an.

Nutzdatenblock

Der Nutzdatenblock besteht aus der kodierten Nachricht und anschließenden Sicherungszeichen. Bei der Kodierung der Nutzdaten verwendet man Forward-Error-Correction (*FEC*) um Übertragungsfehler durch kurzzeitige Störungen auszugleichen. Dazu werden auf der Senderseite dem Bitstrom redundante Bits hinzugefügt. Auf der Empfängerseite kann man damit Übertragungsfehler ohne Rückfrage beim Sender erkennen und gegebenenfalls auch selbständig korrigieren.

Redundanzen und Coding Rate

Bei LoRa™ erfolgt die Kodierung von 4-Bit-Daten mit einstellbaren Redundanzen von 5 bis 8-Bit. Die mögliche Coding-Rate (CR) ist bedingt wählbar: 4/5, 4/6, 4/7 oder 4/8. Das n/m Schema gibt an, dass für n Netto-Bits jeweils m Brutto-Bits aufgewendet werden. Durch Erhöhen der CR kann man die Datenübertragung robuster machen und Interferenzen auf dem Übertragungskanal ausgleichen. Der Preis dafür ist jedoch eine höhere Übertragungszeit für ein Datenpaket. Welche Kombination der Modem Parameter für den jeweiligen Anwendungsfall günstig sind erschließt sich nicht von selbst. Für eine schnelle Bewertung der Modem Parameter und der daraus resultierenden Time-on-Air und des Link Budgets empfiehlt sich das „LoRa Model Calculator Tool“ für den SX1272, welches man von www.semtech.com für Windows kostenlos herunterladen kann [5]. Zum Ende der Übertragung wird das Datenpaket mit 16 CRC Bits ergänzt oder ohne ein CRC abgeschlossen, siehe auch Seite 31.

INBETRIEBNAHME DES LORA™-FTD

Als LoRa™-TRX lässt sich jedes Heltec LoRa 32 V2.0 Modul verwenden. Es muss dazu nur die Firmware geladen und das Modul mit Antenne und Stromversorgung verbunden werden. Zur Konfiguration und Ein-/Ausgabe ist entweder eine Verbindung zu einem Terminal über den USB-Seriell Port oder über eine Bluetooth APP herzustellen.

In diesem Abschnitt ist beschrieben, wie Firmware für den LoRa™-FTD auf dem Heltec Board installiert und das Rufzeichen eingegeben wird. Bei dieser und allen Folgeversionen der Firmware für den Heltec muss nach einem Update der Firmware das Rufzeichen über die Konsole (Serieller Port) oder über ein gekoppeltes Bluetooth Gerät, z.B. ein Telefon, mit entsprechender App erfolgen.

Treiber für den USB-Chip CP2102:

Da am Heltec Board ein USB-Chip CP2102 von Silicon Labs eingebaut ist muss der Treiber für diesen Baustein auf dem Host-Rechner mit dem die Kommunikation zum Heltec Board durchgeführt wird, installiert sein. Man findet den VCP-Treiber und die Installationsprogramme für die gängigen Host-Computer-systeme unter der Rubrik Downloads auf der Webseite von Silicon Labs [CP210x USB to UART Bridge VCP Drivers von Silicon Labs],

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

.

Eine Anleitung für die Installation für MacOS und Windows findet man unter:

https://heltec-automation-docs.readthedocs.io/en/latest/general/establish_serial_connection.html

.

esptool:

Zum Laden der bereitgestellten Firmware in den ESP32-Chip des Heltec benötigt man nun noch das esptool von Espressif.

Installieren des esptool mit Windows:

Es ist möglich eine etwas ältere, in der Konsole direkt ausführbare Version, unter:

<https://dl.espressif.com/dl/esptool-2.3.1-windows.zip>

für Windows herunterzuladen. Gegebenenfalls gibt es zwischenzeitlich bereits eine neuere Version als 2.3.1. Steckt man nun den Heltec an, so wird das Board über USB gefunden und man kann den COM Port bestimmen. Weiter mit "esptool Prüfen" weiter unten.

Installieren des esptool mit anderen Betriebssystemen:

Für Apple, Linux und Windows Rechner kann man alles was zum Programmieren des Heltec Boards benötigt wird auch als Paket herunterladen. In diesem Paket ist auch das esptool vorhanden.

Man folgt hierbei der Anleitung unter:

https://heltec-automation-docs.readthedocs.io/en/latest/esp8266%2Barduino/quick_start.html

In der Arduino IDE klickt man nun unter <Einstellungen> die Schaltfläche für <Ausführliche Ausgabe während: Hochladen> an. Jetzt übersetzt man einen leeren *sketch* und lädt diesen hoch (Pfeil nach rechts im Kreis Symbol). Der *sketch* wird kompiliert und ein Fehler wird angezeigt, da der *sketch* ja auch kein Programm enthält.

Da das Hochladen fehlgeschlagen ist, werden im kleinen Konsolfenster der Arduino IDE Fehlermeldungen (rote Zeilen) ausgegeben. Weiter oben in diesem Fenster findet man die letzte Anweisungszeile (helle Zeile) mit dem Aufruf des esptools. Diese 'sehr' lange Kommandozeile beginnt mit dem Pfad an dessen Ende das esptool installiert ist. Man sieht hier auch den gültigen Port der

seriellen Schnittstelle, z.B.: (hier Mac daher nicht COM sondern /dev/....).

```
/Users/Klaus/Library/Arduino15/packages/esp32/tools/esptool_py/3.0.0/esptool --chip esp32 --port /dev/cu.SLAB_USBtoUART ..... --..... usw.
```

esptool Prüfen:

Nun steckt man das Heltec Board an den Rechner an und stellt sicher, dass die Power LED orangerot leuchtet. Bitte die Arduino IDE jetzt schließen falls diese noch läuft.

In ein Konsolfenster des Rechners (nicht in das Arduino IDE Fenster!) gibt man den Aufruf des Tools ein, z.B.:

```
/Users/Klaus/Library/Arduino15/packages/esp32/tools/esptool_py/3.0.0/esptool
```

und erhält als Antwort die Hilfe-Nachricht des esptools. Wird das Tool nicht gefunden, dann ist mit hoher Wahrscheinlichkeit der Pfad zum Tool falsch eingegeben.

Meldet sich das Tool, so kann die Firmware auf das Board geladen werden.

Firmware auf das Heltec Board laden:

Die Firmware befindet sich in dem mitgelieferten File, hier das Beispiel für die erste Version: **FTDFirmware**

Zur Installation der Software auf dem Heltec Board verwendet man eine Kommandozeile. Das esptool kann nicht durch anklicken geöffnet werden - es lässt sich nur über Kommandozeile steuern. Zum Laden der Firmware muss der Com-Port angegeben werden und der File im richtigen Verzeichnis stehen. Die Anweisung "write_flash 0" schreibt den Inhalt des Files ab Speicheradresse 0x0 in den ESP32 Chip.

Generische Kommandozeile:

```
<pfad>/esptool --chip esp32 --port <com-  
port> write_flash 0 <filename>
```

MacOS:

```
.... /esptool --chip esp32 --port  
/dev/cu.SLAB_USBtoUART write_flash 0  
FTDFirmware
```

Windows:

Auf einem Windows-PC mit dem Heltec, z.B. an COM4:

```
.... /esptool --chip esp32 --port com4  
write_flash 0 FTDFirmware
```

Nach dem Laden der Firmware erscheint auf dem Display des Heltec Boards oben links die Versionsnummer und in der Mitte "DISCONNECTED".

Falls diese Meldung nicht erscheint, dann ist etwas beim Laden der Firmware schief gegangen. Eventuell den Flash-Speicher des ESP32-Chips komplett löschen:

```
.... /esptool erase_flash
```

Jetzt die Firmware nochmals neu laden.

Hinweis: In Mac OS ist z.B. der Pfad z.B. für den Benutzer *Klaus* ähnlich zu:

```
/Users/Klaus/Library/Arduino15/packages/esp32/tools/esptool_py/3.0.0/es-  
ptool erase_flash
```

Terminalprogramm (Konsole):

Zur Bedienung des LoRa™-FTD wurde bei der Entwicklung unter Mac OS X ausschließlich das Konsolfenster der **Arduino IDE** (Serieller Monitor) mit den Einstellungen: „**Sowohl NL als auch CR**“, „**115200 Baud**“ und „**Autoscroll**“ genutzt.

Eine Übertragungsgeschwindigkeit von 115200 Baud ist notwendig, da die Übertragung für die serielle Schnittstelle und für das Bluetooth Interface übereinstimmen müssen und 115200 Baud für Bluetooth nicht verändert werden kann.

Grundsätzlich kann jedes Terminal Programm mit den entsprechenden Einstellungen genutzt werden. Beispielfhaft sind hier zwei gängige Programme exemplarisch angeführt.

Windows:

Unter Windows wurde zum Testen das Programm **Tera Term - VT** verwendet. Unter Setup -> Terminal sind folgende Einstellungen notwendig:

Terminal ID:	VT100
Coding (receive):	UTF-8
Coding (transmit):	UTF-8
New-line Receive:	AUTO
New-line Transmit:	CR + LF
Local echo:	„Häckchen“ <wichtig um die Eingabe zu sehen>
Locale:	german („habe keine Ahnung was das macht“)

Unter Setup -> Font -> Font... sind folgende Einstellungen zu treffen um sowohl Umlaute als auch Sonderzeichen richtig darstellen zu können:

Schriftart:	Courier <bei Courier stimmt UTF 8>
Schriftschnitt:	Standard
Schriftgrad:	9
Skript:	Westlich

Wichtig: Stecken Sie nun den LoRa™-FTD vom PC ab!
Unter Setup -> Serial Port... sehen Sie durch Klicken auf das Port-Pulldown Menu welche Ports bereits ohne den LoRa™-FTD belegt sind. Setup Fenster nun schließen!
Stecken Sie jetzt den LoRa™-FTD an und öffnen Sie erneut das Setup -> Serial Port ... Fenster. Durch den Klick auf das Port-Pulldown Menu erscheint nun zusätzlich als neuer Eintrag der COM Port an den der LoRa™-FTD angeschlossen ist. Wählen Sie nun diesen COM Port durch einen Klick aus.

Port:	<der COM Port an den der LoRa™-FTD angeschlossen ist>.
Speed:	115200
Data:	8 Bit
Parity:	none
Stop bits:	1 bit
Flow control:	none
Beide Transmit delays:	0 msec/line

Klick auf New Setting! Sichern mit Setup-> save Setup

Zum Test ob alle Einstellungen richtig sind drücken Sie bitte auf dem LoRa™-FTD die Reset Taste **R**.

Mac OS:

Zum Testen wurde unter MAC OS X der Terminal Emulator **CoolTerm** verwendet. Nach dem Download und Entpacken ist folgende Konfiguration unter „Options“ vorzunehmen:

Serial Port:

Port: /dev/cu/SLAB_USBtoUSART wählen
Baudrate: 115200
Data Bits. 8
Parity: None
Stop Bits: 1

Keine Häkchen und DTR, RTS auf ‚OFF‘ setzen.

Terminal:

Terminal Mode: Raw Mode
Enter Key Emulation: CR+LF
Custom Sequence: 00 1B
Local Echo: Häkchen (wichtig!)
Text Encoding: UTF8
Rest: keine Häkchen

Fonts:

zweimal Courier wählen

Jetzt ‚OK‘ klicken und in der CoolTerm-Menuleiste (ganz oben) mit „File -> Save as Default“ die Einstellungen sichern. Dann in der ICON-Menuliste den „Connect“ USB-Stecker anklicken.

Zum Test ob alle Einstellungen richtig sind drücken Sie bitte auf dem LoRa™-FTD die Reset Taste **R**.

Die ‚ä‘ in der Präambellänge sollten nun stimmen. Ein einfacher Test ergibt sich mit der Eingabe:

! üöäÜÖÄ\$ß

welche die Ausgabe:

üöäÜÖÄ\$ß

auf dem kleinen Display und

üöäÜÖÄ\$ß

... ..

im Terminal zeigt.

Android:

Als Eingabegeräte zum Senden und Empfangen von Nachrichten kann auch ein serieller klassischer Bluetooth-Port zu einem Endgerät mit Terminal APP verwendet werden.

Es gibt hier z.B. im Play Store viele nutzbare Apps. Bei den Entwicklungsarbeiten hat sich die Serial Bluetooth Terminal App (Stand 12.7.22: Version 1.38) von Kai Morich als sehr nützlich erwiesen (www.kai-morich.de/android).

IOS:

Für Apple Produkte steht (meines Wissens nach) im App Store keine Terminal App zur Verfügung welche mit dem derzeit implementierten klassischen Bluetooth Port verwendet werden kann.

In einer der nächsten Versionen der Firmware für das LoRa™ Field Test Device wird ein Low Energy Bluetooth Interface enthalten sein.

Eingabe des Rufzeichens nach einem Firmware Update:

Ist die Firmware geladen, dann ist das eigene Rufzeichen einzutragen. Dazu gibt es zwei Methoden.

Methode A: Eingabe des Rufzeichens über die Konsole:

1. Das Heltec Board wird über USB an einen Rechner angeschlossen auf dem ein Terminalprogramm läuft oder das Serielle Monitor Fenster in der Arduino IDE geöffnet ist. Es müssen folgende Parameter eingestellt sein: 115200 Baud, sowohl CR und NL und der korrekte COM-Port muss ausgewählt sein. Der Treiber für das Heltec Gerät ist ja bereits vom Laden der Firmware her vorhanden.
2. Nach dem Drücken der Reset Taste am Heltec Board erscheint in der Konsole die Parameterliste. In den ersten Zeilen können noch einige falsche Zeichen stehen – das wird nach dem Eintragen des Rufzeichens verschwinden – also kein Problem an dieser Stelle. Erscheint die Parameterliste nicht, dann stimmt etwas mit den Einstellungen der seriellen Schnittstelle

oder des COM-Ports nicht. Hier kann ich nicht helfen, da ich das System welches benutzt wird nicht kenne und auch nicht kennenlernen möchte.

3. Wenn die Parameterliste angezeigt wird, dann steht in der letzten Zeile: [R] Rufzeichen = NO-CALL
4. Nun geben Sie bitte das eigene Rufzeichen mit dem Befehl:

#rp DLOABC

ein. Keine Leerzeichen verwenden und prüfen dies mit dem Befehl:

#p1 und die Parameterliste mit dem neuen Rufzeichen wird angezeigt. Bei Nutzung von Bluetooth wird das Gerät mit dem eigenen Rufzeichen in der Device Liste angezeigt.

Methode B: Eingabe des Rufzeichens über eine Bluetooth App:

1. Die Anzeige des Heltec Boards zeigt „DISCONNECTED“.

Den Heltec von der Stromversorgung trennen und wieder einschalten. Der Heltec startet neu und zeigt nach 3 Sekunden wieder das gleiche Bild im Display.

Sollte das Display dunkel bleiben, dann den Reset Taster **R** auf dem Heltec kurz drücken und dem Gerät Zeit geben um neu zu starten. Das Display bleibt manchmal lange dunkel wenn Kondensatoren auf dem Board noch geladen sind und dann der „Name-space“ im Flash noch nicht angelegt werden kann. Der Namespace wird später nach der Eingabe des eigenen Rufzeichens permanent angelegt und ab diesem Moment startet das Heltec Board wieder ganz normal.

2. Nun bitte die Bluetooth App am Telefon öffnen.
3. Suche in der App unter Verfügbare Geräte (Devices) das Gerät mit dem Namen "NO-CALL".

Die folgenden beiden Schritte sind je nach verwendeter Bluetooth APP unterschiedlich durchzuführen. Da ich Eure App nicht kenne kann ich hier nicht wirklich helfen.

Die Schritte sind aber auf jeden Fall auszuführen.

4. "NO-CALL" anklicken und - verbinden -
5. "NO-CALL" anklicken und - koppeln -
6. Nach bis zu 5 Sekunden erscheint im Display des Heltec "CONNECTED" und im Display des Telefons "Connecting to NO-CALL" und es wird die Parameterliste ausgegeben.
7. Mit dem Befehl **#PL** wird in der letzten Zeile das Rufzeichen NO-CALL angezeigt.
8. Nun das **eigene Rufzeichen OHNE LEERZEICHEN** am Telefon eingeben, z.B.: **#RP DX0ABC**
9. Es erscheint auf dem Telefon "Neu Permanent: DX0ABC. Wird das eigene Rufzeichen nicht angezeigt, dann Schritt acht (8) wiederholen.
Das eingegebene Rufzeichen wird nun permanent im EEPROM des Heltec Boards gespeichert. Um dieses als Gerät mit dem Rufzeichen als Devicenamen angezeigt zu bekommen, muss man den Heltec ausschalten, neu verbinden und neu koppeln.
10. Heltec Board hierzu komplett ausschalten und den USB abstecken. Der Heltec muss einige Sekunden komplett aus sein!
12. Falls die Bluetooth App noch nicht läuft, diese bitte am Telefon wieder öffnen.
13. Das Telefon mit dem Namen (Device) "NO-CALL" suchen und - **entkoppeln** -
14. Das Gerät ist nun aus der Geräteliste verschwunden. Damit hat das Telefon die Namenszuordnung zum Heltec Board "vergessen".
15. Heltec wieder einschalten und warten bis das Display die "DISCONNECTED" Meldung zeigt. Das kann beim ersten Mal bis zu vier Sekunden dauern weil der interne Flash dazu "umgebaut" wird.
16. Suche in der App unter Verfügbare Geräte (Devices) das Gerät mit dem eigenen Rufzeichen als Gerätenamen (Device).
17. Dieses Gerät - verbinden - und - koppeln -
18. Das Gerät startet.
Der Gerätenamen ist nun das eigene Rufzeichen.

Neustart nach dem Eintrag des Rufzeichens:

Nachdem das Rufzeichen eingetragen ist, wird dieses im internen Flashspeicher des LoRa™-FTD permanent gespeichert und muss nicht jedes Mal neu eingegeben werden.

Startet man das Gerät durch Verbindung mit einer Spannungsquelle bzw. durch das Drücken der Reset Taste „R“, so erscheint auf der Anzeige des Heltec Boards z.B. die Meldung:

```
LoRa BTRX E3_013
DISCONNECTED
Verbinde jetzt
vom Handy aus.
F=439,700 s=0x73
```

Bei neueren Firmware Versionen wird eine andere Versionsnummer und zusätzliche Information angezeigt. Dies gilt auch für die anderen Beispiele in diesem Manual. Ist an den USB-Anschluss ein Terminal angeschlossen (z.B.: in der Arduino IDE bei Wahl des richtigen Com-Ports und den Einstellungen: 115200 Baud, sowohl CR und NL) so wird z.B. folgende Meldung (hier für die Firmware Version E3_13) ausgegeben:

```
Version: E3_13
Alle Rechte bei DL2MEE
Keine kommerzielle Nutzung gestattet
System gestartet
```

LoRa Einstellungen:

```
[C] Coding Rate = 4/5
[BW] Bandbreite = 125.00 kHz
[F] Frequenz = 439.7000 MHz
[G] Rel. LNB Gain = 0 dB
[L] Sendeleistung = 5 dBmW
[PA] Präambellänge = 8
[SF] Spreizfaktor = 10
[SY] Sync-Zeichen = 0x73
[R] Rufzeichen = DL2MEE
```

Ist ein Bluetooth Gerät z.B. ein Android Telefon mit einem Serial Bluetooth Terminal gekoppelt, dann erscheint diese Meldung ebenfalls auch in der APP.

Über Eingaben auf der Kommandozeile des Terminals am Rechner oder in der APP können die Grundparameter geändert oder Nachrichten gesendet werden. Die zur Verfügung stehenden Kommandos können mit der Eingabe „#H“ (auch „#h“ oder einem nicht definierten Kommando bzw. nur mit RETURN) dargestellt werden.

Siehe auch das Kapitel: Kommunikation-Anweisungen.

Betrieb von mehr als einem LoRa™-FTD unter einem Rufzeichen:

Man kann mehr als einen Heltec unter dem eigenen Rufzeichen betreiben. Das Rufzeichen in der App ist ja nur ein lesbares Kürzel für die dahinter liegende MAC-Adresse. In der App sieht man diese normalerweise unter den Devices dargestellt. Hat man nun z.B. mehrere Heltec Geräte eingerichtet und möchte diese auf dem Telefon unterscheiden, dann kann man die Geräte in der App "umbenennen".

Beispielsweise kann man ein Gerät "DL0ABC-Bake" und ein anderes "Tigerbalsam" nennen. Die Namen gelten dann aber nur in der App.

Jeder einzelne Heltec verwendet zum Senden das permanente Rufzeichen (**#rp dl0xyz**) oder, falls eingestellt, das temporäre Rufzeichen (**#rt n8hov**) des jeweiligen Geräts.

Firmware des Heltec Board auslesen und sichern:

Die im Heltec Board gespeicherte Firmware kann mit dem Kommando:

```
.... /esptool --chip esp32 --port  
/dev/cu.SLAB_USBtoUART read_flash  
0 0x200000 <filename>
```

ausgelesen und im File mit dem Namen: <filename> gespeichert werden. Das Auslesen und Speichern dauert etwa 200 Sekunden. Der erzeugte File steht dann im aktuellen Verzeichnis falls kein Pfad angegeben wird.

Zum Rückschreiben verwendet man wie bereits beschrieben, das Kommando:

```
.... /esptool --chip esp32 --port  
/dev/cu.SLAB_USBtoUART write_flash 0  
<filename>
```

Gesicherte Software lässt sich immer auf das gleiche Heltec Modul wieder zurückschreiben. Eine Kopie auf andere Heltec Module empfehle ich nicht, damit der interne permanente Speicher weiter genutzt werden kann

ohne dass man alles wieder neu eingeben muss. Bei einer Neuinstallation sollte man immer eine Originalversion vom Server verwenden denn nur dann werden alle existierenden Daten im permanenten Speicher und alle neuen Elemente (wie in Zukunft auch Texte des entstehenden Morsetrainers) auch platzsparend angelegt und bestehende Daten richtig verschoben.

Da das Gerät und das Programm nun bereits vielfältig genutzt wird habe ich auch schon unterschiedliche Sonder- und Testversionen gebaut. Die Firmware lässt sich bei allen Geräten auslesen und sichern.

Nicht für den Amateurfunk bestimmte Versionen (z.B. für ISM) sind aber an einen einzigen Chip oder an eine Serie gebunden und es erscheint nach dem Laden auf eine andere Hardware nur ein Text oder eine Versionsnummer auf der Konsole (USB Serieller Port).

STEUERKOMMANDOS

Der LoRa™-FTD wird über die Komandozeile mit einem Terminalprogramm auf einem Rechner über USB_Seriell oder einer TerminalAPP mit einem Bluetooth Gerät bedient.

Nach dem Start des TRX kann dieser sofort über den USB mit einem Seriellen Terminal gesteuert werden. Als Parameter sind: 115200 Baud, sowohl NL als auch CR einzustellen. Die Baudrate ist durch die Bluetooth Hardware fest vorgegeben und kann (leider) nicht verändert werden.

Als Terminal Programm verwendet man am einfachsten den Terminal Client der kostenlosen Arduino IDE (Integrated Developer Environment).

Eine Kopplung mit einem Bluetooth Terminal ist jederzeit möglich, je nach Firmwareversion und verwendetem Smartphone bzw. Computer kann die Kopplung etwas länger dauern (bis zu 10 Sekunden). Wird z.B. nach dem Startup der CW-Trainer über die Startup Einstellung (**#X #CW M 4**) automatisch gestartet (siehe Seite 42), so ist eine Kopplung eines Bluetooth Terminals auch während der CW Ausgabe möglich. Der Lösungstext für die CW Übung kann nach Kopplung ab dem Zeitpunkt der Kopplung mitgelesen werden. Die

Kopplung dauert aber einige Sekunden, da der Prozessor dazu mehrfach unterbrochen werden muss. Die CW Ausgabe wird dabei aber nicht gestört.

In der gegenwärtigen Ausbaustufe ist eine Bedienung bzw. Steuerung über WiFi nicht möglich.

Voraussetzung für die Eingabe von Kommandos oder zur Veränderung von Parametern ist, dass der LoRa™-FTD mit einem Terminalprogramm verbunden ist. Siehe hierzu: Inbetriebnahme des LoRa™-FTD.

Die Kommandos sind in die Befehlsgruppen:

- Hochfrequenzparameter
- LoRa™-Parameter
- Kommunikation
- Applikationen
- Systemanweisungen

eingeteilt. Bis auf die Systemanweisungen können alle implementierten Kommandos durch die Eingabe einer „leeren“ Zeile (nur „RETURN“ drücken) ausgegeben werden.

Die folgende Tabelle zeigt dies exemplarisch für die Softwareversion E3_15 (FA Version umfangreicher):

Die verfügbaren Kommandos sind:

. {Textnachricht}

```
#BAKE Bake starten
#BI   Bakeninterval ändern
#BT   Bakentext ändern
#BX   Bakenaussendung stoppen
#BW   Bandbreite ändern
#CR   Coding Rate ändern
#F    Frequenz ändern
#H    Diesen Helptext ausgeben
#G    LNB Verstärkung ändern
#L    SendeLeistung ändern
#LS   List Space - SPIFFS Dir
#MA   Monitor aus
#ME   Monitor ein
#MONI Dual-Frequency Watch
#MR   Memory Read Parameters
#MW   Memory Write Parameters
#PA   Präamble ändern
#PAR  Alle Parameter ausgeben
#PL   Lora Parameter anzeigen
#RT   Rufzeichen temporär ändern
#RP   Rufzeichen permanent ändern
#SF   Spreizfaktor ändern
#SY   Synchronzeichen ändern
```

Hochfrequenzparameter

Der SX1278 Bausteins besteht aus einem Frequenzsynthesizer, einem LoRa™ Modulator/Demodulator, einem HF-Leistungsverstärker Modul im Sendezweig (HF-Endstufe) und einem Vorverstärker/Abschwächer im Empfangszweig (LNB). Diese sind über einen Schalter (Diplexer) mit dem Antennenanschluss verbunden.

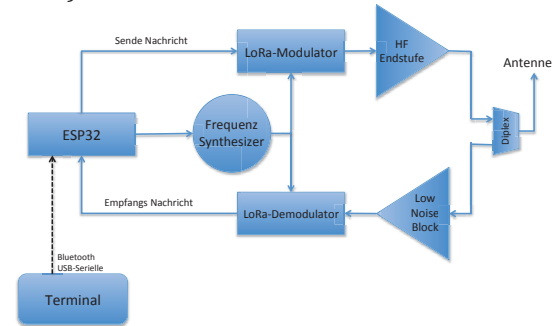


Bild 4: Blockschartbild des LoRa™ HF Pfads des Heltec Boards

Die Frequenz des Synthesizers, die Leistung der Endstufe und die „Verstärkung“ des Eingangsverstärkers Low Noise Block Gain (LNB) lassen sich individuell einstellen.

Das VSWR der Antenne muss unter 3 sein, da andernfalls die Endstufe des SX1278 zerstört wird.

[#F] Frequenz ändern

Beispiel: **#F 438,775**

Bereiche: siehe Text

LoRa™ ist eine **Gleichkanalbetriebsart** bei der die Sende- und Empfangsfrequenz identisch ist. Die Frequenz des Syntesizers lässt sich mit dem Befehl **#F** gefolgt von der Frequenz in MHz einstellen. Das Heltec ESP32 LoRa™ Modul ist für unterschiedliche ISM Frequenzbänder erhältlich. Die mit 433 MHz gekennzeichnete Variante besitzt einen Frequenzumfang von 137 – 525 MHz mit zwei Bandabschnitten in denen auch gesendet werden kann (Band 3: 137 – 175 MHz und Band 2: 410 – 525 MHz). Nominell sind die Filter auf dem Chip aber nur für das spezifizierte Band, also für das 70 cm ISM- und Amateurband, ausgelegt. Es sollte daher im 2 m Band nicht oder nur mit kleinster Leistung gesendet werden. Um Betrieb außerhalb der Amateurbänder zu verhindern ist softwaremäßig eine Einstellung des Frequenzsynthesizers auf die Bereiche **144 – 146 MHz** ($f \geq 143.999$ und $f \leq 146.001$ MHz) sowie **430 – 440 MHz** ($f \geq 430.000$ und $f \leq 440.000$ MHz) begrenzt. Da es sich hierbei um die **Mittenfrequenzen** von LoRa™ Breitbandsignalen handelt ist bei der Frequenzeinstellung jedoch **zwingend darauf zu achten, dass es zu keiner ungewünschten Aussendung außerhalb des zugelassenen Frequenzbands kommt!**

[#G] LNB-Verstärkung (Dämpfung) ändern

Beispiel: **#G 4**

Bereich: 1 ... 6

Für Versuchs- und Messzwecke kann vor dem Low Noise Block eine Dämpfung zugeschaltet werden. Die Regelung der AGC und die RSSI Messung ist von der Einstellung nicht betroffen, da eine automatische Kalibrierung nach jeder Änderung vorgenommen wird. In der Firmware des LoRa™-FTD sind als Parameter die Werte G=1 ... 6 vorgesehen. Voreingestellt ist G=1 was einer LNB-Dämpfung von 0 dB entspricht. Es ist auch der Defaultwert.

#G	dB	#G	dB
1	0	4	-24
2	-6	5	-26
3	-12	6	-48

Die Änderung wird nicht permanent im Speicher des ESP32 abgelegt.

[#L] Sendeleistung ändern

Beispiel: **#L 3**

Bereich: 2 ... 20

Gemäß Datenblatt lässt sich die Sendeleistung der HF Endstufe sich Bereich von ca. -4 dBm (ca. 400 µW) bis 20 dBm (100 mW) in 1 dB Schritten einstellen. In der Firmware des LoRa™-FTD sind als Parameter die Werte **2 ... 20** entsprechend 2 ... 20 dBm vorgesehen. Voreingestellt sind 5 dBm Ausgangsleistung (ca. 3,12 mW als Defaultwert).

dBm	mW	dBm	mW	dBm	mW	dBm	mW
2	1,6	7	5	12	15,9	17	50
3	2	8	6,3	13	20	18	63
4	2,5	9	8	14	25	19	79,4
5	3,1	10	10	15	31,6	20	100
6	4	11	12,6	16	40		

Die Änderung wird nicht permanent im Speicher des ESP32 abgelegt. Bei Verwendung eines Heltec Moduls für 433 MHz gilt obige Tabelle für den Bereich 430 ... 440 MHz. Im Bereich 144 ... 145 MHz führen die Filter im Sendezweig des SX1278 zu einer Fehlanpassung. Daher sollte in diesem Bereich nicht gesendet bzw. nur eine sehr geringe Sendeleistung gewählt werden.

LoRa™-Parameter

Die voreingestellten Betriebsparameter (Defaults) für den LoRa™ Modulator und Demodulator können mit einigen wenigen Befehlen sehr einfach verändert werden. Dadurch ist es z.B. möglich Versuche durchzuführen, Pakete mit unterschiedlichen Parametern zu senden oder mitzulesen und Betriebsparameter für Linkstrecken auszuprobieren.

Die Übertragungsparameter für Bandbreite (Bandwidth: BW), Kodierungsrate (Coding Rate: C) und Spreizfaktor (Spreading Factor: SF) bestimmen, wie empfindlich der LoRa™-Empfänger ist und wie schnell Daten übertragen werden.

Die Präambellänge (Preamblelength: PA) dient zur Synchronisation von Sender und Empfänger und das Synchronisationszeichen (SY) erlaubt somit auch eine Gruppenbildung von Teilnehmern. Das Grundprinzip wäre bei entsprechender Vereinbarung und Implementation möglicherweise ähnlich dem der Sprechgruppen in DMR.

[#BW] Bandbreite ändern

Beispiel: **#BW 3**

Bereich: 0 ... 9

Im Allgemeinen führt die Wahl einer kleineren Bandbreite zu einer langsameren Übertragungsrate (kleineren Symbolrate) aber zu einer höheren Reichweite. Die Wahl einer größeren Bandbreite führt zu einer höheren Datenrate und einer geringeren Reichweite. Die Symbolrate ergibt sich zu: $R_s = \frac{BW}{2SF}$.

Mit z.B. einem Spreizfaktor SF=12 und einer Kodierungsrate von C=4/5 ergeben sich mit den einstellbaren, normierten Bandbreiten BW für 2m und für 70cm folgende Symbolraten R_s :

#BT	2m-Band in kHz	70cm-Band in kHz	SF	C	R_s in Bit/sek
0	7,8	7,8	12	4/5	18
1	10,4	10,4	12	4/5	24
2	15,6	15,6	12	4/5	37
3	20,8	20,8	12	4/5	49
4	31,2	31,2	12	4/5	73
5	41,7	41,7	12	4/5	98
6	62,5	62,5	12	4/5	146
7	125	125	12	4/5	293
8	-----	250	12	4/5	586
9	-----	500	12	4/5	1172

250 bzw. 500 kHz sind im 2m Band nicht einstellbar.

[#CR] Kodierungsrate ändern

Beispiel: **#CR 5**

Bereich: 5 ... 8

Die Kodierungsrate ist das Verhältnis zwischen den Nutzbits der Daten und den hinzugefügten Bits zur Fehlerkorrektur (durch Cyclic Error Coding).

#CR	Kodierungsrate	Mehraufwand bei der Übertragung (Faktor)
5	4/5	1,25
6	4/6	1,50
7	4/7	1,75
8	4/8	2,00

Bei der Wahl der LoRa™-Kodierungsrate ist es wichtig, abzuwägen, ob es notwendig ist, permanent eine hohe Kodiertrate zu verwenden, was zu einer geringeren Datenrate führt. Es kann insgesamt effizienter sein, gelegentliche Paketverluste aufgrund von Störungen zuzulassen. Im Amateurfunk wird meist eine Kodiertrate #CR = 4/5 gewählt da sich dies in zahlreichen Versuchen als zweckmäßig erwiesen hat.

Da die Kodierungsrate die physikalischen Parameter der Modulation nicht verändert, können zwei oder mehr LoRa™-Transceiver, die mit unterschiedlichen Kodierungsraten für bestimmte Linkstrecken konfiguriert sind, die Signale der jeweils anderen dekodieren. Dies kann nützlich sein, wenn sich ein Empfänger in einem Gebiet mit starken Störungen befindet, das andere Ende der Verbindung jedoch nicht. In diesem Fall können die beiden Transceiver mit asymmetrischen Kodierungsraten konfiguriert werden und so optimale Übertragungsraten erzielt werden, auch wenn die Interferenzpegel an den verschiedenen Standorten stark variieren.

(Übersetzt und ergänzt - Quelle: <https://unsigned.io/understanding-lora-parameters/>)

[#CRC] Cyclic Redundancy Check

Beispiele: **#CRC DIS**

#CRC ENA

Generell geht man davon aus, dass ein LoRa™ Paket korrekt empfangen wird und der Empfänger ebenfalls davon ausgehen kann, dass das empfangene Paket gültig ist und verwendet werden kann.

Ob ein Datenpaket während der Übertragung beschädigt wurde lässt sich z.B. mit einer Prüfsumme ermitteln,

welche vor dem Versenden und nach dem Empfang getrennt berechnet wird. Bei LoRa™ verwendet man hier das sog. Cyclic Redundancy Checking Verfahren, welches in den Heltec Chips hardwaremäßig implementiert ist. Stimmen die CRCs überein, geht man davon aus, dass das empfangene Paket in Ordnung ist. Beschädigte Pakete mit einem fehlgeschlagenen CRC Check werden verworfen. Für eine Prüfung muss man sicherstellen, dass der interne CRC des LoRa™-Geräts tatsächlich aktiviert ist. Wenn das CRC Checking nicht aktiviert ist, werden auch beschädigte Pakete akzeptiert. Das kann bei Betrieb an der Rauschgrenze sogar erwünscht sein. Beim Repeater- oder Gateway-Betrieb ist CRC Checking oft sogar vorgeschrieben.

#CRC DIS CRC Generierung und Test vollständig abschalten (disable CRC). Normaler QSO Betrieb bzw. Verbindungen von Endknoten zu Endknoten.

#CRC ENA CRC Generierung und Test für beides, das Senden (TX) und das Empfangen (RX), einschalten (enable CRC). Typischerweise für Repeater und Gateways.

[#IQ] LoRa Modulation Polarity Inversion

Beispiele: **#IQ DIS**
#IQ ENA
#IQ ERX
#IQ ETX

Die sogenannte IQ-Umkehrung (IQ-Invert) dient dazu, Interferenzen zwischen der Up-Link und der Down-Link Kommunikation bei Umsetzer-, Repeater-, Satelliten- und Gateway-Verbindungen abzuschwächen. Wird die IQ-Invertierung eingeschaltet (enabled) so werden die Chirps mit umgekehrtem Frequenzverlauf gesendet und auch so wieder demoduliert. Dadurch kann man sicherstellen, dass sich gesendete und empfangene Nachrichten erkennbar unterscheiden. Normalerweise lesen Gateways Nachrichten mit ausgeschaltetem Invert-IQ und senden mit eingeschaltetem Invert-IQ. Knoten lesen Nachrichten mit eingeschaltetem Invert-IQ und senden mit ausgeschaltetem Invert-IQ. Auf diese Weise kann ein Gateway nur Nachrichten von Knoten und niemals Nachrichten von anderen Gateways demodulieren, und ein Knoten niemals Nachrichten von anderen Knoten. Man vermeidet damit, dass Nachrichten zwischen zwei Knoten unendlich oft hin und her übertragen werden. Die Chips der Semtech SX127X und der SX130X Serie unterstützen die LoRa Modulation Polarity Inversion (IQ-Umkehrung) auf der Hardware Ebene und sind gemäß meiner bisherigen Erfahrung untereinander kompatibel.

Von einigen OMs wurde mir jedoch berichtet, dass anscheinend nicht alle Implementationen der Hardware oder der zur Programmierung verwendeten Bibliotheken zu konsistenten Ergebnissen in der Anwendung führen.

Daher kann man am FTD mit:

- #IQ DIS** die IQ-Umkehrung vollständig abschalten (disable IQ). Normaler QSO Betrieb bzw. Verbindungen von Endknoten zu Endknoten.
- #IQ ENA** die IQ-Umkehrung für beides, das Senden (TX) und das Empfangen (RX), einschalten (enable IQ).
- #IQ ERX** die IQ-Umkehrung nur für den Empfang einschalten (enable IQ for RX).
- #IQ ETX** die IQ-Umkehrung nur zum Senden einschalten (enable IQ for TX).

Möchte man z.B. LoRa™-Satelliten beobachten, so ist **#IQ ERX** sehr nützlich. Zum Empfang des NORSAT 2 schlug Thomas Telkamp im Jahr 2018 folgende Parameter vor:

```
Thomas Telkamp Jan '18
These are the LoRa settings:
Center frequency: 161.862500Mhz
Bandwidth: 125kHz
Sync word: 0x12
Spreading: SF11
Low Data Rate Optimize: on
Header: on
Coding Rate: 4/8 (this is in the header, no need to configure)
IQ inverted (this is default for rx, don't change anything)
Set LNA Gain to max (LNA Boost is not supported for the Lf port).
```

[#SF] Spreizfaktor ändern

Beispiel: **#SF 7**

Bereich: 6 ... 12

Der Spreizfaktor steuert die Chirp-Rate und damit die Geschwindigkeit der Datenübertragung wobei niedrigere Spreizungsfaktoren schnellere Chirps und damit eine höhere Datenübertragungsrate ergeben. Bei jeder Erhöhung des Spreizfaktors halbiert sich die Chirp-Sweep-Rate und damit auch die Datenübertragungsrate. Höhere Spreizfaktoren sorgen für eine höhere Empfängerempfindlichkeit. Normalerweise werden bei schwachen Signalen höhere Spreizfaktoren verwendet.

#SF	Chips / Symbol	Demodulator SNR in dB	Empfänger Empfindlichkeit in dBm für BW = 125 kHz
6	64	-5	-123
7	128	-7,5	-126
8	256	-10	-129
9	512	-12,5	-132
10	1024	-15	-134,7
11	2048	-17,5	-137
12	4096	-20	-138

[#PA] Präambellänge ändern

Beispiel: **#PA 12**

Bereich: 6 ... 65535

Die Präambel ist zur Synchronisation des Empfangsdemodulators notwendig. Die Grundeinstellung (Default) ist eine Sequenz von 12 Symbolen (vollständige Up-Chirps). Für die meisten (ISM) Anwendungen ist dies ausreichend und im Amateurfunk wird PA = 12 auf der Sende- und Empfangsseite verwendet. Sollte die Präambellänge eines Senders nicht bekannt sein, dann kann Empfangsseitig zur Ermittlung der korrekten Präambellänge, PA = 65535 eingestellt werden.

Gesendet werden darf damit aber nicht!

Längere Präambeleinstellungen werden z.B. bei ISM-Anwendungen zum „aufwecken“ von Knoten, z.B. in einem Mesh, verwendet. Hierbei verlässt ein Empfangsknoten den Energiesparmodus, um den Kanal kurzzeitig auf eine Präambel zu prüfen. Wird eine Präambel empfangen, so bleibt der Knoten für eine definierte Zeit im Empfangsmodus um die eigentlichen Daten des Hostknotens zu empfangen. Andernfalls geht der Knoten sofort wieder für eine definierte Zeit in den Energiesparmodus zurück. Typischerweise ist die Sendedauer etwas länger als die Schlafphase.

[#SY] Synchronisationszeichen ändern

Beispiel: **#SY 55**

Bereich: 0x00 ... 0xFF

Nach der Präambel folgt das sog. Synchronisationszeichen, ein Byte also 256 Möglichkeiten, das zur Bildung von Empfangsgruppen (Ruf-Gruppen), für Direktrufe und zur Netzwerk Identifikation verwendet werden kann. Wird ein Datenpaket empfangen, bei dem das Synchronisationszeichen nicht mit dem des Empfängers übereinstimmt, wird dieses Datenpaket verworfen – also nicht dargestellt.

Anm.: Die gegenwärtige Firmware lässt keine Erweiterung zu, so dass alle Datenpakete mitgelesen bzw. dargestellt werden können und ist auf ein einziges Synchronisationszeichen begrenzt.

Im 70 cm Band verwendet man derzeit z.B. die Synchronisationszeichen:

- 0x34 im ISM Band 433/434 MHz
- 0x55 für Punkt-zu-Punkt (Direktverbindung)
- 0x73 für einfachen Repeaterbetrieb .

Kommunikationsanweisungen

Als Eingabegeräte zum Senden und Empfangen von Nachrichten dienen der USB-Port des Heltec Boards zum seriellen COM Port eines Rechners oder der Bluetooth-Port zu einem Endgerät mit Terminal APP.

Nachrichten die gesendet werden sollen gibt man in der Kommandozeile ein. Empfangene Nachrichten werden im Terminal am COM-Port oder in der Terminal APP dargestellt. Durch Einschalten des Monitors (#ME) können Empfangspakete auch auf dem kleinen Display des Heltec Boards dargestellt werden.

[.] Nachricht mit Rufzeichen senden

Beispiel: **. Testme** Bereich: max. 255 Zeichen

Der Text nach dem Punkt wird ohne das separierende Leerzeichen mit angehängtem „de“ und dem eigenen Rufzeichen gesendet.

Gesendet wird: **.Testme de DL2MEE**

Das Rufzeichen kann mit dem Befehl #RT temporär und mit #RP permanent geändert werden.

[:] Nachricht ohne Rufzeichen senden

Beispiel: **:Testme**

Bereich: max. 255 Zeichen

Der eingegebene Text wird so wie eingegebenen ohne weitere Zeichen gesendet. Ein Rufzeichen wird nicht angehängt. Damit kann z.B. ein APRS-String für ein LoRa APRS i-Gate oder ein Datenpaket für LoraHAM direkt gesendet werden. (Die drei Steuerzeichen am Anfang der Nachricht an ein i-Gate sind als Buchstaben zu kodieren). Auch ist es damit möglich einen Teststring an ein Messgerät oder zu Fernsteuerzwecken auszusenden.

[!] Übungstext in CW ausgeben

Beispiel: **! Test foo**

Bereich: max. 255 Zeichen

Der Übungstext nach dem Ausrufezeichen wird ohne das separierende Leerzeichen in Morsetelegraphie ausgegeben. Auf der kleinen Anzeige erscheint der eingegebene Übungstext während der Ausgabe. Auf der Konsolenausgabe werden zusätzlich auch die Punkte (dit) und Striche (dah) dargestellt. Die Parameter der Gehörschrift

lassen sich mit den Parametern der CW-Applikation ändern (siehe Seite 72f). Für die Ausgabe ist das Grundgerät mit einer Zusatzschaltung zum Anschluss des Kopfhörers zu erweitern (siehe Seite 82).

SYSTEMANWEISUNGEN

Alle Systemanweisungen beginnen mit dem „#“ Zeichen in der Kommandozeile.

[#CAT] Catenate – SPIFFS

Beispiel: **#CAT <filename>**

Bereich: filename max. 6 Zeichen beginnend mit einem Buchstaben (kein Schrägstrich)

Mit **#CAT** wird der Inhalt eines im SPIFFS Speichers (Serial Peripheral Interface Flash File System) des ESP32 abgelegten Files ausgegeben. Es wird keine Formatierung vorgenommen. Damit können auch fehlerhaft geschriebene Daten oder Fileinhalte erkannt werden. Namenskonvention für die Angabe von Filenamen siehe **#MW** auf Seite 38.

Der Name „cat“ leitet sich vom englischen Verb concatenate bzw. dessen Synonym catenate ab.

[#H] Helptext ausgeben

Beispiel: **#H**

Eine Liste der aktuell verfügbaren Kommandos wird mit **#H** oder bei Eingabe eines ungültigen Kommandos bzw.

bei Eingabe einer leeren Kommandozeile ausgegeben. Servicekommandos und Kommandos für Funktionen welche sich noch im Beta-Teststadium befinden, werden in dieser Liste nicht dargestellt.

Beispielausgabe für die Softwareversion E3_14 (FA Version umfangreicher):

```
Die verfügbaren Kommandos sind:
. {Textnachricht}
: {Textnachricht}
! {CW-Übungstext}
#BAKE Bake starten
#BI   Bakeninterval ändern
#BT   Bakentext ändern
#BE   Bakenaussendung stoppen
#BW   Bandbreite ändern
#CR   Coding Rate ändern
#CW   Telegraphie Modus
#F    Frequenz ändern
#H    Diesen Hilfetext ausgeben
#G    LNB Verstärkung ändern
#L    SendeLeistung ändern
#LS   List Space - SPIFFS Dir
#MA   Monitor aus
#ME   Monitor ein
#MONI Dual-Frequency Watch
#MR   Memory Read Parameters
#MW   Memory Write Parameters
#PA   Präamble ändern
#PAR  Alle Parameter ausgeben
```


#PL	Lora Parameter anzeigen
#REE	Repeater stoppen
#REP	Repeater starten
#RES	Reset des Geräts
#RT	Rufzeichen temporär ändern
#RP	Rufzeichen permanent ändern
#SF	Spreizfaktor ändern
#SY	Synchronzeichen ändern
#X	Startup Kommando setzen

[#HELP] Helptext ausgeben

Beispiel: **#HELP**

Dieses zusätzliche Kommando wurde auf den Wunsch von Nutzern eingeführt und entspricht dem Kommando #H welches auf Seite 36 beschrieben ist.

[#HILFE] Helptext ausgeben

Beispiel: **#HILFE**

Dieses zusätzliche Kommando wurde auf den Wunsch von Nutzern eingeführt und entspricht dem Kommando #H welches auf Seite 36 beschrieben ist.

[#HEX] Empfangene Zeichen in HEX darstellen

Beispiel: **#HEX ENA**
#HEX DIS

Nach der Eingabe **#HEX ENA** (ENable HEXadecimal Mode) wird für jedes empfangene Zeichen nur der HexCode und nicht das Zeichen selbst ausgegeben. Zur einfacheren Weiterverarbeitung wird jedes empfangene Zeichen auf einer eigenen Zeile ausgegeben und führende 0en (Nullen) unterdrückt, z.B. wird für 0x0A der Wert 0xa ausgegeben, was mit EXCEL oder anderen Programmen direkt weiterverarbeitet werden kann.

Beispiel:

Gibt man z.B. auf dem sendenden LoRa™-FTD das Kommando: **. test de DL2MEE** ein, so wird auf dem LoRa™-FTD auf der Empfangsseite mit **#HEX ENA** folgende Liste ausgegeben:

0x74
0x65
0x73
0x74
0x20
0x64
0x65
0x20
0x44
0x4c
0x32
0x4d
0x45
0x45
0xd
0xa

Nach der Eingabe von **#HEX DIS** (DISable HEXadecimal Mode) werden alle nachfolgenden Zeichen wieder in ASCII lesbarer Form dargestellt.

[#LS] List Space – SPIFFS Dir

Beispiel: **#LS**

Zeigt den Inhalt des SPIFFS Speichers (Serial Peripheral Interface Flash File System) des ESP32 an. Dieser enthält die mit **#MW** gespeicherten Parameterfiles. Namenskonvention für Filenamen siehe **#MW** auf Seite 38.

[#ME] Monitor einschalten

Beispiel: **#ME**

Eingehende Datenpakete werden zusätzlich auf dem kleinen Display des LoRa™-FTD dargestellt.

[#MA] Monitor ausschalten

Beispiel: **#MA**

Die Funktion eingehende Datenpakete auf dem Display des LoRa™-FTD darzustellen wird deaktiviert.

[#MR] Memory Read Parameters

Beispiel: **#MR**

#MR <filename>

Bereich: filename max. 6 Zeichen beginnend mit einem Buchstaben (kein Schrägstrich)

Mit Memory Read Parameters kann man die im nicht-flüchtigen SPIFFS Speicher (Serial Peripheral Interface Flash File System) des ESP32 mit **#MW** gespeicherten Parameter wieder einlesen. Die entsprechenden Parameter werden dabei bis zu einer Änderung oder einem Reset temporär überschrieben. Namenskonvention für Filenamen siehe **#MW** auf Seite 38.

[#MW] Memory Write Parameters

Beispiel: **#MW**

#MW <filename>

Bereich: filename max. 6 Zeichen beginnend mit einem Buchstaben (kein Schrägstrich)

Mit diesem Kommando werden alle derzeit gültigen Parameter in den nichtflüchtigen SPIFFS Speicher (Serial Peripheral Interface Flash File System) übertragen und in einem File abgelegt. Namenskonvention für Filenamen siehe **#MW** auf Seite 38.

[#QC] Quick Charge Mode

Beispiel: **#QC**

#qc

Um den Ladevorgang über die eingebaute USB Verbindung beschleunigen zu können kann man durch die Eingabe von **#QC** den ESP32 Prozessor in den Tiefschlaf schicken und das Terminal ausschalten. Dadurch wird nahezu kein Strom verbraucht und der Ladevorgang geht dadurch etwas schneller. Der Prozessor kann nur durch Drücken der Reset **R** des Heltec Moduls oder durch Ein-/Ausschalten des Moduls wieder gestartet werden.

[RECON] LoRa™ Nachrichten protokollieren - EIN

Beispiel: **#RECON**

Parameter: keine

Alle eingehenden LoRa™-Nachrichten werden im File **NOTE.txt** abgespeichert, siehe hierzu auch die Namenskonvention für SPIFFS-Files auf Seite 41. Der File **NOTE** wird beim Aufruf des Kommandos **#RECON** automatisch angelegt sofern dieser File nicht bereits existiert. Eine Liste der existierenden Files kann mit **#ls** (siehe Seite 37) ausgegeben werden.

```
Steuerkommando: LS
SPIFFS Directory:
/f.txt
/NORBI.txt
/NOTE.txt
```

Der File **NOTE** ist mit dem Kommando **#rm NOTE** (siehe Seite 42) löscherbar und kann mit **#cat NOTE** (siehe Seite 36) ausgegeben werden.

Bei jeder Eingabe von **#RECON** werden in den File **NOTE** die zu diesem Zeitpunkt gültigen LoRa™-Parameter als Text gespeichert. Ändert man diese, so wird diese Änderung nicht automatisch im File gespeichert. Erst eine weitere Eingabe von **#RECON** führt zur erneuten Speicherung der Parameter.

```
-----  
Aktuelle LoRa Einstellungen:  
Coding Rate   = 4/5  
Bandbreite    = 250.00 kHz  
Frequenz      = 436.7015 MHz  
Rel. LNB Gain = 0 dB  
Spreizfaktor  = 10  
Sync-Zeichen  = 0x12  
Präambellänge = 8  
-----
```

Alle LoRa™-Nachrichten werden sukzessive an die bestehenden Zeilen angehängt (konkateniert mit `append()`).

```
-----  
DL2MEE Test Paket 1   :: PRSSI = -74 SNR = 9 FERR = 16119  
DL2MEE Test Paket 2   :: PRSSI = -75 SNR = 7 FERR = 16120  
DL2MEE Test Paket 3   :: PRSSI = -74 SNR = 9 FERR = 16117  
DL2MEE Test Paket 4   :: PRSSI = -76 SNR = 8 FERR = 16115  
-----
```

Da das Filesystem SPIFFS mit zunehmender Speichernutzung immer langsamer wird, kann es anscheinend vorkommen, dass Pakete nicht mehr gespeichert werden können und „verloren“ gehen. Dieses Problem ist bei meinen Tests bisher glücklicherweise nie aufgetreten - berichtet werden die Probleme ab 1/3 der Belegung des gesamten Flashspeichers des ESP32. Es empfiehlt sich daher den **NOTE** File nach Beendigung des Versuchs einfach wieder zu löschen.

Wie bereits gezeigt, werden zur späteren Begutachtung der empfangenen Datenpakete nach jeder Nachricht der Packet RSSI, der SNR und der Frequenzoffset angehängt. Zur Trennung sind drei Leerzeichen gefolgt von drei Doppelpunkten eingefügt. Ein Zeitstempel für die Empfangszeit der Nachrichten ist leider bisher noch nicht verfügbar, da das Gerät noch keine Systemuhr besitzt. Eine Erweiterung mit einem Uhrenmodul, wie z.B.: Tiny RTC-Modul DS1307, ist angedacht aber bisher leider aus Zeitgründen noch nicht realisiert.

[RECOFF] LoRa Nachrichten protokollieren - AUS

Beispiel: **#RECOFF**

Parameter: **keine**

Eingehenden LoRa™-Nachrichten werden nicht mehr im File **NOTE** protokolliert.

Nach der Eingabe von **#RECON** werden LoRa™-Pakete wieder im File **NOTE** angehängt.

Wird das Protokoll der LoRa™ Nachrichten nicht mehr auf dem LoRa™-TRX benötigt, so empfiehlt es sich dieses ggf. über eine Verbindung des Geräts über USB mit **#cat** **NOTE** auszugeben, zu sichern und anschließend mit **#rm** **NOTE** (siehe Seite 42) vom LoRa™-TRX zu löschen.

Namenskonvention SPIFFS <filename>:

Filenamen beginnen auf keinen Fall mit einem Schrägstrich und enthalten keine Sonderzeichen! Systemfiles werden mit Kleinbuchstaben angelegt und können vom Nutzer des Geräts nicht manipuliert werden. Nutzerfiles werden mit Großbuchstaben bezeichnet.

Der Schrägstrich vor dem Filenamen und die Filenamen-erweiterung .txt nach dem Filenamen wird vom Betriebssystem des LoRa™-TRX eigenständig hinzugefügt.

Wird kein <filename> für einen Parameterset angegeben so wird der Parameterset im sytemeigenen File **/f.txt** gespeichert. Dieser File wird automatisch generiert.

Ein <filename> darf nur 6 Zeichen besitzen und muss mit einem Buchstaben beginnen. Werden mehr als sechs Zeichen für den Filenamen benutzt, so werden diese ignoriert, also abgeschnitten. Es wird immer die Extension „.txt“ angehängt. Andere Extensions werden durch „.txt“ ersetzt.

Mit dem Servicekommando **#*77** (siehe Seite 46) kann der SPIFFS Bereich des ESP32 Prozessors zu Service und

Programmierzwecken auf der Konsole ausgegeben aber nicht verändert werden.

[#RM] Remove File – SPIFFS

Beispiel: **#RM <filename>**

Bereich: filename max. 6 Zeichen beginnend mit einem Buchstaben (kein Schrägstrich)

Löscht einen File mit dem Namen **<filename>** aus dem SPIFFS Speichers (Serial Peripheral Interface Flash File System) des ESP32. Namenskonvention für Filenamen siehe **#MW** auf Seite 38.

Systemfiles, wie z.B.: /f.txt haben Filenamen die mit Kleinbuchstaben beginnen. Diese können nicht mit dem Kommando **#RM** gelöscht werden. Eine Löschung, z.B.: vor der Weitergabe des Geräts an einen anderen OM oder einer Firmwaresicherung (siehe Seite 24) kann nur durch ultimatives Löschen aller Inhalte im Flashspeicher durch Formatierung und gleichzeitigem Rücksetzen des EEPROMs erreicht werden, siehe unter Kommando Domsday **#*13** auf Seite 45.

[#X] Startup Kommando festlegen

Beispiele: **#X <Startup Kommandos>**

#X #MR

#X #MR #CW M 3 #-

#X #BAKE

#X #-

Beim Einschalten oder nach dem Drücken der Reset Taste **R** des Heltec Moduls wird das Startup Kommando genau einmal ausgeführt. Mit dem Kommando **#X** kann man das Kommando bzw. die Kommandos festlegen welche direkt nach dem Reset auszuführen sind. Diese werden als „Startup Sequenz“ bezeichnet und im „permanenten“ Flash Speicher des ESP32 abgelegt. Die Startup Sequenz darf keinen Reset **#RES** enthalten! Die Startup Sequenz **#-** (siehe Seite 43) wird ignoriert und es wird nichts ausgeführt.

Möchte man z.B. die letzten Einstellungen des Geräts, also den Status bewahren, so sichert man diesen vor dem Ausschalten mit **#MW** (siehe Seite 38) und trägt als Startup Sequenz **#MR** (siehe Seite 38) ein. Als Kommando gibt man hierzu **#X #MR** ein.

Manche Funktionen lassen sich durch Drücken der Taste **P** unter- bzw. abbrechen.

Beispiel: **#X #CW F 1**

Beim Startup werden für **(1)** eine Minute **(F)** Fünfergruppen in **(#CW)** Telegraphie ausgegeben. Dabei werden die zuletzt verwendeten Einstellungen der Telegraphieübung genutzt.

Beispiel: **#X #-**

Beim Startup soll kein Kommando ausgeführt werden. Es können dann nur Kommandos auf der Konsole oder über Bluetooth direkt eingegeben werden.

Beispiel: **#X #REP**

Das Gerät wird als Repeater gestartet. Weitere Parameter können noch angefügt werden → siehe Applikation **#REP**.

[#RES] Reset

Beispiel: **#RES**

Bei Eingabe von wird ein Reset des Geräts mit nachfolgendem Start der mit dem Startup Kommando festgelegten Sequenz durchgeführt.

[#-] Nil

Beispiel: **#-**

Die Eingabe führt keine Funktion aus und verändert keine Variable(n) → Anwendung siehe **#X**.

[#PAR] Alle Parameter Anzeigen

Beispiel: **#PAR**

Diese Kontrollfunktion zeigt alle aktuell gültigen Parameter und die interne Darstellung an. Das interne Format wurde gewählt um dem **#cat** und **#*77** Format zu entsprechen.

[#PL] LoRa Parameter Anzeigen

Beispiel: **#PL**

Diese Kontrollfunktion zeigt die aktuell gültigen LoRa™ Einstellungen.

Ausgabe für z.B. die Softwareversion E3_15 (FA Versionen umfangreicher):

LoRa Einstellungen:

[CR]	Coding Rate	= 4/5
[BW]	Bandbreite	= 125.00 kHz
[F]	Frequenz	= 439.7000 MHz
[G]	Rel. LNB Gain	= 0 dB
[L]	Sendeleistung	= 5 dBm
[PA]	Präambellänge	= 8
[SF]	Spreizfaktor	= 10
[SY]	Sync-Zeichen	= 0x73
[R]	Rufzeichen	= DL2MEE

[#RT] Rufzeichen temporär ändern

Beispiel: **#RT N8HOV**

Bereich: Rufzeichen ohne Leerzeichen

Mit diesem Kommando wird das Rufzeichen für Kommunikations-, Nachrichten- und Bakenaussendungen nur bis zum Reset oder Neustart des LoRa™-FTD temporär geändert. Das neu eingegebene Rufzeichen wird im internen, flüchtigen RAM-Speicher abgelegt.

[#RP] Rufzeichen permanent ändern

Beispiel: **#RT DL0LW**

Bereich: Rufzeichen ohne Leerzeichen

Diese Funktion ändert das Rufzeichen für Kommunikations-, Nachrichten- und Bakenaussendungen des LoRa™-FTD bis zur nächsten permanenten Rufzeichenänderung mit #RP. Das neu eingegebene Rufzeichen wird im internen Flash-Speicher abgelegt. Die Eintragungen auf den Bluetooth Gerät(en) bleibt hiervon unberührt, da das Rufzeichen auf den Bluetooth Geräten nur ein Platzhalter für die eigentliche MAC-Adresse ist. Bei einer Änderung muss die Koppelung und Verbindung der Bluetooth Geräte neu erfolgen.

Serviceanweisungen

Serviceanweisungen erlauben es Einstellungen in der LoRa™ Hardware vorzunehmen, Konstanten in der Firmware zu verändern und neue Kommandos im Beta Stadium der Entwicklung zu testen bzw. Fehler darin zu finden. Diese Kommandos sind **nicht für den alltäglichen Gebrauch gedacht** und können dazu führen, dass die Firmware ganz oder in Teilen neu aufgespielt werden muss. Bei auftretenden Fehlern drückt man zuerst die „P“ und dann die „R“ (Reset) Taste des Heltec Boards um in einen stabilen Zustand zurückzukehren.

[#*1] Namespace Inhalt ausgeben

Beispiel: **#*1**

Keine Parameter

Das temporär und das permanent hinterlegte Rufzeichen werden ausgegeben. Ist kein Rufzeichen hinterlegt oder eine Speicherung fehlgeschlagen wird als Default „NO-CALL“ angezeigt und sollte mit #RP umgehend korrigiert werden.

[#*13] Doomsday – Alles Löschen

Beispiel: **#*13**

Keine Parameter

Alle in den Speichern des Geräts vorhandenen Daten und Einstellungen werden ultimativ gelöscht !

Führen Sie diese Funktion nur dann aus wenn Sie z.B. das Gerät an einen anderen OM weitergeben möchten und dazu alle Parameter im Gerät löschen möchten. Nach der Eingabe von **#*13** folgt man den Anweisungen im Konsolfenster. Man wird aufgefordert die P Taste für einige Sekunden zu halten.

Nicht gelöscht werden die MAC-Adresse und die Chip-ID.

[#*32] Device ID (MAC) ausgeben

Beispiel: **#*32**

Keine Parameter

Die Device ID (MAC Adresse) wird nur auf der Konsole über USB ausgegeben!

Bei Heltec Prozessoren ist die Device ID mit der vom Hersteller zugewiesenen eindeutigen MAC Adresse identisch. MAC-Adresse steht für Media Access Control Address und

ist eine eindeutige Hardware-Kennung, die jedes Gerät in einem Netzwerk identifiziert. Eine MAC-Adresse besteht aus sechs Gruppen von zwei hexadezimalen Ziffern. Hier sind diese nicht wie sonst üblich durch Doppelpunkte getrennt.

Beispiel:

Device = 487449A8CC84

[#*60] Inhalt der LoRa™ Chip Register ausgeben

Beispiel: **#*60**

Keine Parameter

Mit diesem Kommando werden die Register des LoRa™ Chips SX127X auf dem Heltec Board ausgegeben. Zur Interpretation der Registerinhalte und der darin enthaltenen Steuerbits ist das entsprechende Manual der Firma Semtec [5] zu verwenden.

Mit dem Befehl **#*60** lassen sich Einstellungen überprüfen, z.B.: nach dem Kommando **#IQ DIS** zeigt das Register **0x33: 0x27** (Beispielsweise hier Bit 6: InvertIQRX 0->Normal; 1->Inverted) und nach dem Setzen des IQ-Bits mit **#IQ ENA** den Inhalt **0x33: 0x66**. Die Ausgabe erfolgt nur auf der Konsole (seriell).

[#*77] Inhalt des SPIFFS ausgeben

Beispiel: **#*77**

Keine Parameter

Mit Eingabe des Systemkommandos **#*77** wird der Inhalt des (S)erial (P)eripheral (I)nterface (F)lash (F)ile (S)ystems, der ein einfaches Dateisystem halten kann, auf der Konsole ausgegeben. In diesem Dateisystem steht die Datei `/f.txt` in welcher, z.B. mit dem Kommando **#MW**, der Status und alle eingestellten Parameter abgelegt werden können. Für Service- und Programierzwecke wird `/f.txt` ausgegeben. Ein Inhalt kann nur dann gezeigt werden wenn vorher mit **#MW** (siehe Seite 38) der Status gespeichert wurde.

[#*99] Namespace Inhalt löschen

Beispiel: **#*99**

Keine Parameter

Das permanente Rufzeichen im Flash-Speicher des LoRa™-FTD wird gelöscht und durch „NO-CALL“ ersetzt. Es muss nun mit **#RP** neu eingegeben werden. Zu beachten sind auch hier die neuen Einstellungen auf dem Bluetooth Gerät(en).

APPLIKATIONEN

Der LoRa™-FTD enthält bisher eine Bake, einen Repeater, einen APRS Tester und einen Morsetrainer als Applikationen. Weitere Applikationen, die bereits entwickelt aber noch nicht auf den ESP-32 portiert sind, werden folgen. Hierzu gehöre z.B. eine Dual-Watch Applikation mit graphischer Darstellung der Feldstärken auf dem kleinen Display zur gleichzeitigen Beobachtung zweier Frequenzen und weitere Applikationen für messtechnische Aufgaben, wie z.B. Bitfehlerratenmessung, Kollisionserkennung, Paketprotokollierung mit Annotation, Linkbegutachtung und automatische Ermittlung von Linkparametern.

LoRa™-Bake

Bei Nutzung der Bake an einer Antenne sind zwingend die Vorschriften für automatisch arbeitende Stationen genau zu beachten und eine Genehmigung einzuholen!

Die hier implementierte Bake dient ausschließlich zur Erzeugung eines wiederkehrenden Datenpakets für Messzwecke. Hiermit kann z.B. im Labor ein definiertes Signal erzeugt und ein normales Empfangsgerät oder ein LoRa™-Datenempfänger eingestellt bzw. auf Funktion überprüft werden. Ebenfalls können z.B. Linkstrecken oder Antennenanordnungen überprüft und optimiert werden – die gesetzlichen Bestimmungen sind zu beachten. Hierzu sendet die Bake in einstellbaren Zeitabständen wiederkehrend ein Datenpaket mit folgender Struktur:

<Rufzeichen> *** <Bakentext> *** <Wert des Bakenzählers>

Beim Start der Applikation wird der Bakenzähler auf Null gesetzt und zur Überprüfung die Grundparameter (Frequenz, Rufzeichen, Bakentext und

Wiederholungsintervall) auf Konsole und Bluetooth Gerät ausgegeben. Nach jeder Übertragung wird der Wert des Bakenzählers (Nummer des Bakentelegramms) ebenfalls auf Konsole und Bluetooth Gerät ausgegeben und der Bakenzähler nach jeder Aussendung um Eins erhöht. Damit lassen sich Messwerte den empfangenen Paketen leicht zuordnen und Paketverluste festgestellt werden.

Anmerkung: In der gegenwärtigen Implementierung kann die Ausgangsleistung nur fest eingestellt werden (z.B. #L 5). Eine automatische Stufung wie bei Baken im Kurzwellenbereich ist noch nicht implementiert.

[#BAKE] Starten der Bakenapplikation

Beispiel: **#BAKE**

Keine Parameter

Mit dem Start der Applikation Bake werden die vorab eingestellten LoRa™-, Hochfrequenz- und Bakenparameter übernommen und es beginnt die zyklische Aussendung von nummerierten Datenpaketen.

[#BE] Beenden der Bakenapplikation

Beispiel: **#BE**

Keine Parameter

Die Bakenapplikation wird durch die Eingabe eines ungültigen Kommandos bzw. durch die Eingabe von **#BE** erst nach Ablauf des Bakenintervalls und Übertragung des letzten Datenpakets beendet. Reset **R** wirkt sofort.

[#BI] Einstellung des Bakenintervalls

Beispiel: **#BI 12**

Bereich: 5 ... 86401

Als Intervallzeit, die Zeit zwischen zwei aufeinander folgenden Aussendungen von Datenpaketen der Bake, sind 7 Sekunden voreingestellt (Defaultwert). Diese Zeit kann von 5 Sekunden bis 86401 Sekunden (24 Stunden) eingestellt werden. Da die Intervallzeit auf dem ESP-32 Board von der Taktfrequenz des ESP-32 abgeleitet wird, ist diese nicht sehr genau und die Einstellung muss ggf. durch Versuche ermittelt und entsprechend eingestellt werden.

[#BT] Einstellung des Bakentexts

Beispiel: **#BT Neuer Bakentext**

Bereich: 1 ... 128 (darstellbare) Zeichen

Der voreingestellte Bakentext:

LoRa™-Versuchsbake

lässt sich durch jede, nicht leere Folge von bis zu 128 alphanumerischen Zeichen ersetzen.

Anmerkung: Da die Zeichen einzeln an den Datenpuffer des LoRa™-Bausteins übergeben werden ist der Bakentext streng genommen nicht längenbegrenzt und kann 128 Zeichen überschreiten. Die Synchronisation mit dem Empfänger geht jedoch bei längeren Paketen „verloren“ und kann ggf. durch die Anpassung der Präambellänge verbessert werden. Da dies eine Möglichkeit für Versuche bietet, wird die Länge des Bakentexts nicht geprüft und kein Fehler angezeigt.

LoRa™-Repeater

Bei Nutzung des Repeaters an einer Antenne sind zwingend die Vorschriften für automatisch arbeitende Stationen genau zu beachten und eine Genehmigung dafür einzuholen!

Im Gegensatz zu einer Relaisstation ist bei einem Repeater die Empfangs- und Sendefrequenz identisch. Repeater arbeiten im **Gleichkanalbetrieb**. Ein empfangenes Paket wird mit den gleichen Betriebsparametern und auf der gleichen Frequenz, evtl. angereichert mit Messdaten, wieder zurückgesendet. Oft bezeichnet man derartige, automatisch arbeitende Stationen auch als „Papagei“. Mit Repeatern lassen sich im Labor lokal oder aber auch über kurze Distanzen LoRa™-Geräte bzw. Software testen und Pakete weiterleiten.

Für den Repeaterbetrieb werden die aktuell gültigen LoRa™- und HF-Grundeinstellungen übernommen. Man muss daher vor dem Aufruf der #REpeater Funktion alle Einstellungen für den Sendebetrieb vornehmen und am Besten mit der „“ Funktion testen.

Wird der LoRa™-FTD von der Spannungsquelle getrennt oder abgeschaltet, so wird der Repeater beim Wiedereinschalten nicht automatisch gestartet!

Der Repeater lässt sich nicht fernsteuern.

Monitoring empfangener Pakete

Alle Pakete werden mit den dazu gehörenden Empfangsparametern über die serielle Schnittstelle des Heltec Boards (115200 Baud, sowohl NL als auch CR) am Repeater LoRa™-FTD ausgegeben. Wird z.B. ein Paket mit dem Inhalt: „test de DL2MEE“ an den Repeater gesendet, so kann man bei Verwendung der Arduino-IDE auch den Zeitstempel anzeigen und damit das Paket millisekundengenau nachverfolgen oder protokollieren.

```
12:15:46.404 -> Das wurde Empfangen und wird weitergeleitet:
12:15:46.404 -> test de DL2MEE
12:15:46.404 ->
12:15:46.404 -> RSSI:                -130 dBm
12:15:46.404 -> Packet_RSSI:         -112 dBm
12:15:46.404 -> Packet_SNR:          7 dB
12:15:46.438 -> Frequency Error: -15857 Hz
12:15:46.438 -> -----
```

Parameter Syntax

Die Kommandozeile beginnt mit einem # Zeichen und kann danach folgend, in Groß- oder auch in Kleinbuchstaben, auch gemischt, eingegeben werden (case insensitive). Leerzeichen sind nicht zwingend erforderlich, erhöhen aber die Lesbarkeit. Die Parameter müssen mit mindestens einem Bindestrich vom Kommando **#rep**

getrennt werden. Weitere Bindestriche und Leerzeichen, auch zwischen weiteren Parameterzeichen, werden ignoriert. Parameter können in beliebiger Reihenfolge angegeben werden.

[#REP] Starten der Repeaterapplikation

Beispiel: **#REP**

Parameter: **A,C,P,R,S,F**

Mit dem Start der Applikation werden die vorab eingestellten LoRa™- und Hochfrequenzparameter übernommen. Nach einem Minuszeichen „-“ können weitere Steuerungsparameter angefügt werden.

Gibt man nur das Kommando **#REP** ein, dann wird, getrennt durch „**** routed via*“, das aktuell gültige Rufzeichen des LoRa™-FTD an das Paket angehängt und das so erweiterte Paket wieder ausgesendet.

Beispiel:

Das Paket mit dem Inhalt:

```
test de DL2MEE
```

wird vom Repeater mit

```
test de DL2MEE
```

```
*** routed via DL0UBW
```

zurückgesendet.

[-C] Repeaterkennung

Beispiele: **#REP -C**
#REP - C
#REP - <weitere Parameter> **C** <weitere Parameter>

Mit dem Steuerungsparameter -C wird die Ausgabe der Repeaterkennung unterdrückt. Das Paket wird genau so wieder ausgesendet wie es empfangen wurde. Wird der Parameter -C weggelassen, so wird die Repeaterkennung immer angehängt (default), vgl. folgende Beispiele. Der Steuerungsparameter -C beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater mit

test de DL2MEE

zurückgesendet.

[-R] Signal RSSI

Beispiele: **#REP -R**
#REP - R
#REP - <weitere Parameter> **R** <weitere Parameter>

Bei LoRa™-Paketen ist der Indikator für die empfangene Signalstärke der sog. Signal-RSSI (Received Signal Strength Indicator – RSSI). Der Signal-RSSI (kurz nur RSSI) ist ein relativer Wert, der die Signalstärke bezogen auf einen vom Hersteller definierten Wert von -164 dBm, für die Bereiche: 137-175 MHz und 410-525 MHz, anzeigt. Gemäß LoRa™ Definition sollen zur Ermittlung des RSSI die letzten zwei Up-Chirps des Frame Syncs und die direkt darauf folgenden beiden Down-Chirps am Ende der Präambel herangezogen werden. Da sich dies nur schwer realisieren lässt, wird oft die gesamte Präambel zur Ermittlung des RSSI verwendet.

Der Steuerungsparameter -R bedingt die Ausgabe des RSSI des empfangenen Signals am Repeater. Der Steuerungsparameter -R beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater z.B. mit

test de DL2MEE

*** routed via DL0UBW

HF_RSSI: -102 dBm

zurückgesendet.

Anmerkung: Der Messwert entspricht dem Registerinhalt des Semtech 127X und ist für RSSI > -100dBm nicht linear (siehe Semtech Datenblatt, S.87). Die im Labor gemessene Abweichung war kleiner als ± 1 dB pro dB, bezogen auf den idealen Wert. Für die Korrektur $SNR > 0$ gibt Semtech im Datenblatt folgende Formel an:

$$RSSI = -164 + \frac{16}{15} \cdot Packet_{RSSI};$$

[-P] Paket-RSSI

Beispiele: #REP -P

#REP - P

#REP - <weitere Parameter> P <weitere Parameter>

Mit Breitspektrum Chirp-Modulation kann man auch Pakete im Rauschen (below the noise floor) empfangen. Man berechnet daher einen mittleren RSSI über die gesamte Empfangsdauer eines Pakets unter der Berücksichtigung (Faktor: 0,25) des gemessenen Signal-Rausch Abstands (SNR), den man als Paket-RSSI bezeichnet.

Der Steuerungsparameter -P bedingt die Ausgabe des P_RSSI des empfangenen Pakets am Repeater.

Der Steuerungsparameter -P beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater z.B. mit

test de DL2MEE

*** routed via DL0UBW

P_RSSI: -108 dBm

zurückgesendet.

Anmerkung: Der Messwert entspricht dem Registerinhalt des Semtech 127X und ist für RSSI > -100dBm nicht linear (siehe Semtech Datenblatt, S.87).

[-S] Packet SNR

Beispiele: **#REP -S**

#REP - S

#REP - <weitere Parameter> **S** <weitere Parameter>

Das Signal-Rausch-Verhältnis SNR (Signal-to-Noise Ratio) ist definiert als das Verhältnis der mittleren Leistung des Nutzsignals zur mittleren Rauschleistung des Störsignals bzw. des Grundrauschens.

Der Steuerungsparameter **-S** bedingt die Ausgabe des RSSI des empfangenen Signals am Repeater.

Der Steuerungsparameter **-S** beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater z.B. mit

test de DL2MEE

*** routed via DL0UBW

P_SNR: 11 dB

zurückgesendet.

[-F] Frequency Error

Beispiele: **#REP -F**

#REP - F

#REP - <weitere Parameter> **F** <weitere Parameter>

Der Steuerungsparameter **-F** bedingt die Ausgabe der Frequenzabweichung in Hertz zwischen der Mittenfrequenz des empfangenen Signals und des Quarzoszillators des LoRa™-Moduls auf dem Heltec Board. Der LoRa™-Baustein kann Abweichungen der Mittenfrequenz bis ca. $\pm 25\%$ der Bandbreite BW tolerieren und die Abweichung in diesem Bereich messen. Das Ergebnis dieser Messung wird ausgegeben.

Der Steuerungsparameter **-F** beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater z.B. mit

test de DL2MEE

*** routed via DL0UBW

F_ERR: -17073 Hz

zurückgesendet.

[-A] Alle Messwerte senden

Beispiele: **#REP -A**

#REP - A

#REP - <weitere Parameter> **A** <weitere Parameter>

Der Steuerungsparameter **-A** bedingt die Ausgabe der Repeaterkennung und aller Messwerte.

Der Steuerungsparameter **-A** beeinflusst die Ausgabe über die serielle Schnittstelle am Repeater nicht.

Anmerkung: Bei **#REP -ac** bzw. **#REP -ca** wird der Steuerungsparameter **-c** ignoriert.

Beispiel:

Das Paket mit dem Inhalt:

test de DL2MEE

wird vom Repeater z.B. mit

test de DL2MEE

*** routed via DL0UBW

P_RSSI: -108 dBm HF_RSSI: -107 dBm P_SNR: 7 dB

F_ERR: -16738 Hz

zurückgesendet.

[#REE] Beenden der Repeaterapplikation

Beispiel: **#REE**

Keine Parameter

Die Repeaterapplikation wird durch die Eingabe eines ungültigen Kommandos bzw. durch die Eingabe von **#REE** beendet. Eine noch laufende oder initiierte Paketaussendung wird jedoch nicht mehr abgebrochen.

LoRa™-Monitor Dual-Watch

Mit der Dual-Watch Anwendung kann man die Aktivität auf zwei Frequenzen im Bereich von 137 MHz bis 525 MHz beobachten und die sog. Kanal RSSI Werte ausgeben.

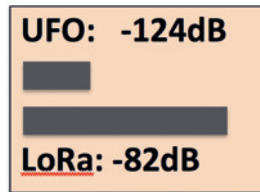


Bild 5: Balkenanzeige im Dual-Watch Mode

Die Balkenanzeige dient dabei nur zur Visualisierung der ermittelten Kanal-RSSI Werte und die Skalierung wird fortwährend so angepasst, dass die Balken zunächst linear anwachsen und dann der Balken für das stärkere Signal die Skalierung für das schwächere Signal bedingt. Es ist daher nur eine informative Anzeige und für quantitative Messungen nicht geeignet. Diese Art der Anzeige wurde aber bewusst so gewählt, weil dann beide Balken in Gänze sichtbar sind und man dann sofort erkennt welches Signal gerade das Dominante ist. Damit ist es z.B. möglich eine Störquelle auf einem Nachbarkanal zu finden, zu bewerten bzw. festzustellen warum zu

bestimmten Zeiten Paketverluste auftreten. Mit dem Dual-Watch Monitor war es beispielsweise möglich am Standort des DBOUFO die Beeinflussung des LoRa™-Repeaters durch die Antenne des benachbarten DMR-Relais zu ermitteln und die Antennenstandorte der beiden Geräte zu optimieren. Hierbei war keine genaue Anzeige sondern eine relative Darstellung mit gut erkennbaren Unterschieden sehr hilfreich.

Das Display teilt sich in die Information für die Frequenz [FA] mit dem oberen Balken und [FB] mit dem unteren Balken. Über bzw. direkt unter dem jeweiligen Balken steht links das Acronym und rechts der zahlenmäßig ermittelte RSSI Wert des LoRa™-Empfängerbausteins.

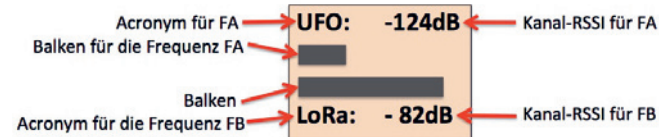


Bild 6: Anzeigemodule des Dual-Watch im Display

Mit dem jeweilige Acronym [ACA] bzw. [ACB] kann man die Balken semantisch besser zuordnen.

Während einer Beobachtung werden die gemessenen Kanal-RSSI Werte zusammen mit den Frequenzen und Acronymen in der Konsole ausgegeben. Verwendet man für die Ausgabe z.B. die Konsole der Arduino IDE, so

kann man neben ‚Autoscroll‘ auch noch den ‚Zeitstempel anzeigen‘.

```
12:35:14.587 -> 439.70000 MHz Fre1:: -123 db ; 433.95000 MHz Fre2:: -124 db
12:35:14.761 -> 439.70000 MHz Fre1:: -122 db ; 433.95000 MHz Fre2:: -87 db
12:35:14.934 -> 439.70000 MHz Fre1:: -123 db ; 433.95000 MHz Fre2:: -87 db
```

Kopiert man die Ausgabe in einen File mit der extension **.csv** , so lassen sich die Zeilen mit Leerzeichen separiert, z.B. auch in EXCEL, weiterverarbeiten.

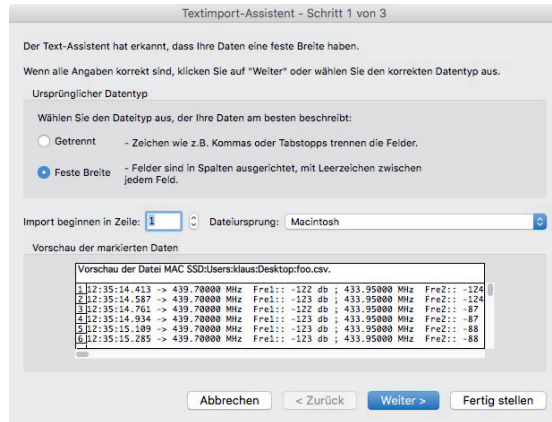


Bild 7: Einstellungen zum Import des .csv Files in EXCEL.

[#MONI] Starten der Monitor Applikation
Beispiel: **#MONI**

Parameter: **FA,FB,ACA,ACB**

Mit dem Start der Applikation werden die vorab eingestellten Frequenz- und Acronymparameter übernommen und das Balkendiagramm auf dem kleinen Display desLoRa™-TRX dargestellt sowie die ermittelten Werte über die serielle Schnittstelle ausgegeben.

Die Monitor Applikation lässt sich jederzeit mit der Taste **P** beenden.

[FA] Frequenz FA einstellen

Beispiel: **#MO FA 438,775**

Bereich: [siehe Text](#)

Die Frequenz des Synthesizers für den oberen Balken A lässt sich mit dem Befehl **#MO FA** gefolgt von der Frequenz in MHz einstellen.

Die Einstellung der Frequenz 438,775 MHz ermöglicht die „Überwachung“ dieser Frequenz und visualisierung der Feldstärke auf dieser Frequenz. Die eingestellte Bandbreite muss größer 40 kHz [#BW 5 ... 9] sein, ist für beide Balken A und B gleich und geht (irgendwie) in die Messung ein. Eine Bandbreite von 125 kHz [#BW 7] lieferte bisher die besten Ergebnisse.

Das Heltec ESP32 LoRa™ Modul ist für unterschiedliche ISM Frequenzbänder erhältlich und die mit 433 MHz gekennzeichnete Variante besitzt für den Empfang einen Frequenzumfang von 137 – 525 MHz. Der gesamte Bereich ist für den Monitor Dual-Watch Betrieb einstellbar. Die ermittelten Kanal-RSSI werte sind aber nur für zwei Bandabschnitte (Band 3: 137 – 175 MHz und Band 2: 410 – 525 MHz) auf 1 dB genau. Eine Version mit gemessenen Korrekturtabellen für Frequenz- und Bandbreiteneinstellungen und entsprechender automatischer Interpolation ist noch nicht ganz fertig und wird in einer zukünftigen Version verfügbar sein.

[FB] Frequenz FB einstellen

Beispiel: **#MO FB 145,775**

Bereich: [siehe Text](#)

Siehe obige Erläuterung für [FA].

[ACA] Acronym für FA eingeben

Beispiel: **#MO ACA UFO:**

Bereich: [4 ASCII Zeichen maximal](#)

Das Acronym ist ein lesbarer Bezeichner für die gewählte Frequenz über dem RSSI Balken links und in der Liste ausgegeben. Die Defaultbelegung von ACA ist „Fre1:“. Nur bei der Initialisierung sind fünf Zeichen für das Acronym möglich.

[ACB] Acronym für FB eingeben

Beispiel: **#MO ACB LoRa:**

Bereich: [4 ASCII Zeichen maximal](#)

Das Acronym für die gewählte Frequenz FB wird unter dem RSSI Balken links und in der Liste ausgegeben. Die Defaultbelegung von ACB ist „Fre2:“. Nur bei der Initialisierung sind fünf Zeichen für das Acronym möglich.

LoRa™-APRS Testset

Mit dem APRS Testset können GPS-Positionsdaten an ein APRS i-Gate gesendet und in der Karte auf <https://aprs.fi/> dargestellt werden. Hierzu ist ein APRS Paket von einem „Tracker“ an ein APRS i-Gate zu senden. Als Default Position wurden zufällig für den GPS Punkt die Koordinaten: 48° 06.81N, 11° 53,66 E gewählt. Dies entspricht JN58WC, also genau der Mitte des Anzinger-Englharter Forsts, der hiermit eine sicherlich unerwartete Aufmerksamkeit erhalten wird.
<https://aprs.fi/#!mt=road-map&z=12&lat=48.1189&lng=11.9074>

APRS Parameter

Für LoRa™-APRS werden in DL derzeit folgende Parameter eingesetzt (Stand Oktober 2021):

TX-Frequenz: 433.775 MHz (von Tracker zu i-Gate)
RX-Frequenz: 433.900 MHz (von i-Gate zu Tracker - bei RxGates nicht aktiv).

Spreading factor:	[SF]	12
Bandwidth:	[BW]	125 kHz
Coding rate:	[C]	4/5
Sync word:	[SY]	0x12
LoRa™ Modus:		explicit

In DL wird das OE-LoRa Format für den Up-Link zu einem APRS i-Gate eingesetzt. Die ersten drei Bytes im Header des LoRa™-APRS-Datenpakets geben den Typ: "<" (0x3C), das Ziel: 255 (0xFF) und die Quelle: 1 (0x01) an. Der Rest des Datenpaketes wird in reinem ASCII gemäß der APRS Protocol Reference siehe: <http://www.aprs.org/doc/APRS101.PDF> [8], übertragen. Die CRC-Berechnung/Auswertung ist im Amateurfunk meist ausgeschaltet bzw. wird vom LoRa™ Baustein in Hardware ausgeführt. Es gelten auch in dieser Betriebsart die Protokollbeschreibung sowie Empfehlungen von Bob, WB4APR (<http://www.aprs.org>).

Beispiel APRS Test Paket

```
<?DL2MEE-7>APLT00-1:14806.84N/01153.78E[LoRa-FTD
```

Dies bedeutet aufgelöst:

<	(0x3C)	OE-Header:
?	255	(0xFF) OE-Header: Ziel
0	1	(0x01) OE-Header: Quelle hier Tracker
DL2MEE		Rufzeichen des Senders
-7		SSID für Portabeleinsatz
>		Trennzeichen
AP		Destination Address
LT		LoRa™-Tracker
00		Software Version hier 0
-1		Control Field/ID
:!		Einleitung Lat/Long Position Format
4806.84N		48° 06.84' North
/		Trennzeichen
01153.78E		011° 53.78' East (fünfstellige Angabe!!!)
[Darstellung laufende Person in der Karte
LoRa-FTD		Paketnr: 5

Info die in der Karte gezeigt wird. Hier kann ein beliebiger Text stehen, kann auch leer gelassen werden.

Koordinatenbestimmung Ui-View

Einen Standort kann man ganz einfach in der zoombaren Landkarte unter <https://aprs.fi/> nachschauen. Im Kartenbild wird die Position des Mauszeigers im oberen Teil ausgegeben. Um es direkt übernehmen zu können, sollte auf aprs.fi das Ui-View Format: "Grad°MM.MM (APRS)" eingestellt sein (Auswahl mit: rechte, blaue Menueleiste -> Einstellungen -> Einheiten und Zeit). Auf normalen Karten werden die geografischen Koordinaten normalerweise mit Grad, Minuten und Sekunden angegeben. Dann muss man umrechnen weil bei Ui-View Grad, Minuten und 1/100 Minuten angegeben werden.

[anz] Anzinger Forst

Beispiel: **#aprs anz**
#APRS ANZ

Keine Parameter

Dieser Steuerungsparameter setzt die Koordinaten defaultmäßig auf die Mitte des Anzing-Egelhartinger Forsts. Die Wahl der Position bei 48° 06.81N; 011° 53.66E also JN58WB erfolgte rein zufällig und gestattet eine schnelle Überprüfung in aprs.fi unter <https://aprs.fi/#!lat=48.08270&lng=11.63306>.

[def] Default APRS HF-Parameter setzen

Beispiel: **#aprs def**
#APRS DEF

Keine Parameter

Dieser Steuerungsparameter setzt die für LoRa™-APRS notwendigen Grund- bzw. Defaultparameter:

Frequenz Tracker zu i-Gate	433.775 MHz
Spreading Factor	12
Coding Rate	4/5
Bandbreite	125 KHz
Synchronzeichen	0x12
CRC	Aus

Alle Hochfrequenzparameter werden übernommen bzw. können separat eingestellt werden. Nach der Eingabe von **#APRS def** werden alle relevanten LoRa™- und HF-Parameter zur Überprüfung nochmals ausgegeben. Diese können dann auch einzeln noch geändert werden.

[h] Hilfetext für die APRS Parameter

Beispiel: **#aprs h**
#APRS H

Keine Parameter

Eine beispielhafte Ausgabe des Hilfetexts für die Testapplikation APRS sieht wie folgt aus:

Beispiel für die Softwareversion E3_17 (FA Versionen umfangreicher):

```
APRS-Parameter:
[anz] Position im Anzinger Forst
[def] Default HF-Parameter setzen
[sym] Kartensymbol setzen
[lat gmmm.mmN|S] Lattitude
[lon gggmm.mmE|W] Longitude
[pos] Aktuelle Position ausgeben
[put] Position senden
[ssid] APRS SSID setzen
[str] APRS Message setzen
```


[sym] APRS Kartensymbol setzen

Beispiel: **#aprs sym** <weitere Parameter>

#APRS SYM <weitere Parameter>

Parameter: ein ASCII-Zeichen entsprechend der APRS Symbolspezifikation (Basis Set).

Im Basis Set sind 92 Bildsymbole definiert, Bild 8. Die folgende Abbildung des Basis Sets ist ein überarbeiteter Auszug aus der Originaltabelle von W6KWF. Das gewählte Paketformat für **put** gestattet nur ein Zeichen als Symbol Code, daher sind keine Overlays (siehe Format auf Seite 65) möglich.

Ein grau hinterlegtes ASCII Zeichen steht für das Symbol welches dann in der Karte dargestellt wird. Das Zeichen ; entspricht z.B. einem grünen Zelt. Mit

#APRS sym ;

wählt man z.B. das Symbol (;) für das Zelt aus.

Beispiel: Darstellung eines grünen Zelts an der Position: 4806.62N/01153.00E. Es soll zusätzlich noch der Kommentar „Hallo 1234“ im Info Feld erscheinen.



Bild 8: Bildsymbole des APRS-Basissets

Um das i-Gate zu erreichen wird exemplarisch hier z.B. 18 dBmW eingestellt. Das i-Gate verwendet in diesem Beispiel die Standard LoRa™- und HF-Parameter.

```
#L 18
#aprs def
#aprs lat 4806.62n
#aprs lon 01153.00e
#aprs sym ;
#aprs str Hallo 1234
#aprs put
```

Der LoRa™-FTD sendet den String:

```
DL2MEE-7>APLT00-  
1:!4806.62N/01153.00E;Hallo 1234
```

und erzeugt damit auf aprs.fi an der gewünschten Position das „Zelt“ Symbol. Der Update der Karte kann etwas dauern, da die Verarbeitung durch die APRS Serverstruktur noch erfolgen muss. Das Roh-Paket, also die Info wie diese am Server einging, kann man unter rechts unter andere Ansichten -> Roh-Pakete darstellen.



Bild 9: Anzeige des Kartensymbols mit Metadaten

[lat ggmm.mmN/S] Latitude

Beispiel: **#aprs lat**
#APRS LAT 4806.84N

Bereich: 00.xxN ... 89.xxN bzw.: 00.xxS ... 89.xxS
mit xx: 00 ... 99

Die geographische Breite oder Breitengrad genannt, ist die im Winkelmaß in der Maßeinheit Grad angegebene nördliche oder südliche Entfernung eines Punktes der Erdoberfläche vom Äquator.

Die Eingabe hat das Format (GradMinuten.hunderstel-Minuten):

lat 4806.84N

lat gibt den gegenwärtig eingetragenen Winkel aus.

[lon gggmm.mmE|W] Longitude

Beispiel: **#aprs lon**

#APRS LON 01140.74E

Bereich: 000.xxE ... 179.xxE bzw.: 000.xxW ... 179.xxW
mit xx: 00 ... 99

Die Longitude gibt den Längengrad ist die Messung östlich (East) oder westlich (West) des Nullmeridians an. Die geografische Länge wird mit Hilfe von imaginären Linien gemessen, die vertikal (nach oben und unten) um die Erde verlaufen und sich am Nord- und Südpol treffen. Jede der 360 Grad Linien entspricht einem Bogengrad der geografischen Länge. Wichtig ist hierbei die fünfstellige Eingabe vor dem Trennungspunkt von 000° bis 180° mit vorangestellter Null bei kleinen Winkeln gefolgt von E (Ost) oder W (West).

Die Eingabe hat das Format (GradMinuten.hunderstel-Minuten):

lon 01140.74E

lon gibt den gegenwärtig eingetragenen Winkel aus.

[pos] Aktuelle Position ausgeben

Beispiel: **#aprs pos**

#APRS POS

Keine Parameter

Mit dem Steuerungsparameter pos wird die Latitude und Longitude der aktuell eingegebenen Position zur Überprüfung ausgegeben.

[put] Position senden

Beispiel: **#aprs put**

#APRS PUT

Keine Parameter

Die Übertragung der Positionsinformation an ein LoRa i-Gate erfolgt mit dieser Anweisung. Zur Überprüfung wird das gesendete APRS-Paket ohne die drei Header Bytes, ausgegeben.

Hinweis: Bei jedem Ein/Ausschalten gehen evtl. eingestellte Parameter verloren und ein -put führt zu keinem „Ergebnis“ in der Karte.

Oft hilft dann die Eingabe der Sequenz:

#L 18	Leistung 18 dBm
#aprs def	APRS-LoRa HF-Parameter
#aprs anz	Anzinger Forst Parameter
#aprs put	Position an i-Gate senden

[ssid] APRS SSID setzen

Beispiel: **#aprs ssid** <weitere Parameter>

#APRS SSID <weitere Parameter>

Parameter: 1 ... 15

SSIDs werden zur Unterscheidung verschiedener APRS Anwendungen genutzt. Entsprechend dem AX25 Protokoll sind die SSIDs auf 16 (0 - 15) begrenzt. Die -0 wird in der Regel nicht angezeigt und daher nicht verwendet. Wird die SSID weggelassen so handelt es sich um die primäre Feststation die an der in der Rufzeichenliste der Bundesnetzagentur eingetragenen Adresse betrieben wird.

Die Empfehlungen zur Auswahl der APRS SSID findet man unter: <http://aprs.org/aprs11/SSIDs.txt>.

Unter dieser Adresse findet man folgende Liste.

APRS SSID Liste

SSID	Beschreibung
-0	Ortsfeste Primärstation. Für Rufzeichen ohne SSID wird die SSID Null angenommen.
-1,2,3,4	Generische Zusatzstation, Digi, Mobilgerät, Wetterservice, usw.
-5	Andere Netze (Dstar, iPhones, Androids, Blackberrys, Tablets, usw.)
-6	Sonderaktivitäten, Satellitenbetrieb, Camping oder 6 Meter, usw.
-7	Handfunkgeräte oder andere tragbare Geräte und LoRa (portabel).
-8	Boote, Segelboote, Wohnmobile oder zweites Hauptmobil
-9	Primäres Mobilgerät
-10	Internet, I-Gates, Echo-Link, Winlink, AVRS, APRN, usw.
-11	Ballons, Flugzeuge, Raumfahrzeuge, usw.
-12	APRStt, DTMF, RFID, Geräte, One-Way-Tracker, LoRa™-Tracker, usw.
-13	Wetterstationen
-14	Trucker oder generell Vollzeitfahrer
-15	Allgemeine Zusatzstation, Digi, Mobil, wx, usw.

Quelle: <http://aprs.org/aprs11/SSIDs.txt>

Darstellung in APRS.fi

Wird ein gültiges APRS Paket von einem LoRa i-Gate empfangen so kann dieses in <https://aprs.fi/> dargestellt werden. Enthält das Paket Positionsdaten dann wird das ICON z.B. für “[“eine laufende Person an der übermittelten GPS Position in der Karte dargestellt. Die Zuordnung der ICONs und weitere Informationen findet man z.B. in den Vortragsunterlagen von Manfred Mauler, OE7AAI [9].

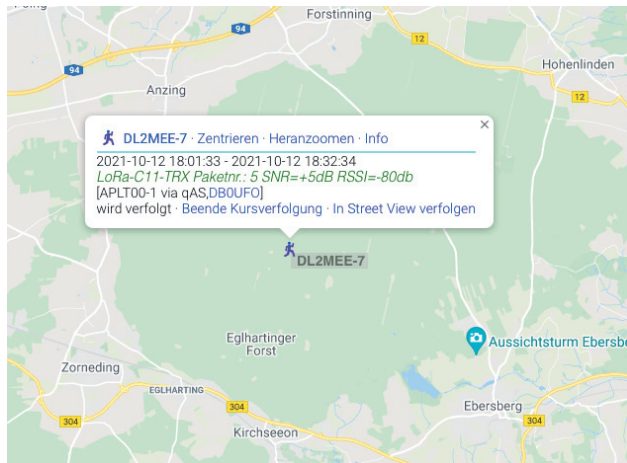


Bild 10: www.aprs.fi -Kartenausschnitt

[str] APRS String-Comment setzen

Beispiel: **#aprs str** <ASCII Zeichen>

#APRS STR <ASCII Zeichen>

Bereich: 0 ... 43 ASCII Zeichen

Das mit **#aprs put** gesendete APRS Paketanteil ist nach dem *Lat/Long Position Report Datenformat – ohne Timestamp* aufgebaut.

Lat/Long Position Report Format – no Timestamp

!	Lat	Sym Table ID	Long	Sym- bol Code	Com- ment
1	8	1	9	1	0 – 43

Quelle: APRS Protocol Reference – Version 1.0

Beispiel:

DL2MEE-7>APLT00-1:**!4806.84N/01153.78E[LoRa-FTD**

Mit diesem Kommando kann der voreingestellte Kommentar verändert werden.

Morsetrainer

Da die Morsetelegraphie eine digitale Betriebsart ist, darf hier ein Morsetrainer nicht fehlen – das FTD wäre ohne diesen nicht vollständig.

Zur Nutzung ist eine einfache Erweiterung aufzubauen, siehe Seite 82.

Die Applikation besteht aus einem Codegeber welcher direkt mit dem Kommando ! (Seite 35) aufgerufen werden kann.

Beispiel:

! Das ist ein cw-Text.

Der nach dem Ausrufezeichen angegebene Text wird mit den eingestellten Parametern als Morsezeichen in Schriftform auf der Konsole und hörbar mit einem Kopfhörer ausgegeben.

Die Ausgabe kann an den sog. Telegraphie-Blockgrenzen, also nach einer zusammengehörigen Zeichengruppe (Fünfergruppe, Afu-Abkürzung, Übungssegment oder Spruchelement) durch die Betätigung der Taste **P** des Heltec Boards unterbrochen werden. Die Unterbrechung und die damit verbundene Rückkehr in den Konsolmodus erfolgt immer erst nach Erreichen der Telegraphie-Blockgrenze!

Der Codegeber kann den vollständigen Zeichensatz bestehend aus Buchstaben und Umlautzeichen (Groß- und

Kleinschreibung gleichgesetzt), Ziffern, Sonde- und Betriebszeichen verarbeiten.

Buchstaben: aA bB cC ,ch' ,CH' dD eE fF gG hH iI jJ kK lL mM nN oO pP qQ rR sS ß tT uU vV wW xX yY zZ

Umlautzeichen: äÄ öÖ üÜ

Ziffern: 1234567890

Sonderzeichen:

/ . , : ? - ; + = _ () " ' ! @ % \$ § # &

Betriebszeichen:

HH Irrung [*HH]

KA Spruchanfang [*KA]

BK Aufforderung zum Senden (Bitte Kommen) [*BK]

BT Trennung (Beginn Trennung) [*BT]

AR Spruchende [*AR]

AS Warten [*AS]

OK Alles in Ordnung ('Oscar Keller') [*OK]

K Aufforderung zur Übermittlung (Kommen) [*K]

VV Spruchanfang [*VV]

EE Spruchbestätigung (End of Everything) [*EE]

Trennzeichen im Text:

‚Carriage Return‘ und ‚Line Feed‘ werden ignoriert, ein eingegebenes Leerzeichen (Blank) wird als angepasste Pause umgesetzt.

[A] CW Abkürzungen

Beispiel: **#CW A** <n>
#cw a <n>

Bereich: $n > 1$

Nach einem Spruchanfang (*vv) werden n-Standard Amateurfunkfunkabkürzungen ausgegeben und mit Spruchende (*ee) abgeschlossen. Die Ausgabe lässt sich mit der Taste **P** jederzeit unterbrechen. Zur Überprüfung der Übung werden die gegebenen Abkürzungen auf dem Display einzeln und als fortlaufende Liste sowohl seriell als auch über Bluetooth ausgegeben. Auf der Konsole werden zusätzlich dahs und dits in Morseschrift gezeigt.

Beispiel: **#CW A 5**

```
Übung Afu Abkürzungen:
...-   ...-   ...-   ...-
fb
...-   -...
QRS
...-   ..   ...
frd
...-   ...   -..
LUDWIG
...-   ..   ...   ...-   ..   --
test
-   .   ...   -
...-   ...-   ..
```

[D] CW Abkürzungen

Beispiel: **#CW D** <n>
#cw d <n>

Bereich: $0 \leq n \leq 200$

Um das Mitschreiben besser zu ermöglichen kann zwischen Wörter ein zusätzlicher Abstand „eingebaut“ werden. Diese Distanz wird in n Punkten angegeben. Es hat sich als zweckmäßig erwiesen diese Distanz abhängig von der Zeichenkombination etwas flexibel zu gestalten was die Lesbarkeit verbessert.

Beispiel: **#x #CW D 20 #CW W 28 #CW A 200**

[F] CW Fünfergruppen

Beispiel: **#CW F** <min>
#cw f <min>

Bereich: 1 ... 60 Minuten

Nach Spruchanfang (*vv) werden m-Minuten Fünfergruppen mit einmaliger Wiederholung, bestehend aus Buchstaben und Ziffern, ausgegeben und mit Spruchende (*ee) abgeschlossen. Die Zeitangabe ist nur ein ca. Richtwert! Die Ausgabe lässt sich mit der Taste **P**

jederzeit unterbrechen, die Wiederholung wird jedoch vollständig ausgeführt.

Auch hier werden zur Überprüfung der Übung die gegebenen Gruppen auf dem Display einzeln und als fortlaufende Liste sowohl seriell als auch über die Bluetooth-schnittstelle ausgegeben. Auf der Konsole werden zusätzlich dahs und dits in Morseschrift gezeigt.

Beispiel: **#CW F 2**

```
...- ...- ...- ...-
Übung Fünfergruppen:
OYTDA
--- -- - - - - - - - - - - - - - -
WVNOD
...- ...- ...- ...- ...- ...- ...-
« die weiteren Fünfergruppen »
FJYAD
...- ...- ...- ...- ...- ...- ...-
...- ...- ...- ...- ...- ...- ...-
```

[FA ★] CW Fünfergruppen - angepasste Übungen

Beispiel: **#CW FA** <Parameter>

#cw fa <Parameter>

Parameter: A,B,Z,I,S,X

Für die angepassten Übungen stehen neben den Buchstaben, Ziffern und Sonderzeichen entsprechend dem internationalen Zeichensatz zur Verfügung.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890
/.,:~?~+=() ' ! ; _ @ % \$ # & " \$ Ä Ü Ö

Je nach Auswahl des Gruppenparameters können unterschiedliche Zusammenstellungen gewählt werden. Alle Funktionen lassen sich durch Drücken der Taste **P** abbrechen. Die Ausgabeparameter für CW werden ohne Änderung verwendet.

[FA ★] Fünfergruppen Hilfetext ausgeben

Beispiel: **#CW FA** leer oder ungültiger Parameter

#cw fa leer oder ungültiger Parameter

Werden nach **#CW FA** keine weiteren oder inkorrekte Parameter angegeben, dann wird der Hilfetext ausgegeben.

Beispiel:

```
Für #CW FA <gültiger Parameter>
n steht für die Anzahl der
auszugebenden Fünfergruppen.
A n: Alphanummerisch A-Z 0-9
B n: Nur Buchstaben A-Z
Z n: Nur Ziffern 0-9
I n: Interpunktion /.,:~+=()
S n: Sonderz: /.,:~+=()$$ßäüö
X n: Extended: alle Zeichen.
```

[FA A] Alphanummerische Fünfergruppen

Beispiel: **#CW FA A 5**
#cw fa a 5

Die Ausgabe von n alphanummerischen Fünfergruppen erfolgt mit dem Zeichensatz:

ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Es werden für die gesamte Übung dreimal so viele Buchstaben wie Ziffern pseudozufällig aus dem Zeichensatz gewählt. Die Ausgabe in Morsezeichen beginnt mit der Spruchanfang- und endet mit der üblichen Spruchendesequenz. Die Übung lässt sich durch Drücken der Taste **P** abbrechen, die Spruchendesequenz wird dann nicht ausgegeben.

[FA B] Buchstaben Fünfergruppen

Beispiel: **#CW FA B 20**
#cw fa b 20

Für die Ausgabe von n Fünfergruppen die nur aus Buchstaben bestehen wird der Zeichensatz:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

verwendet. In der Übung werden pseudozufällig alle Buchstaben ungewichtet ausgewählt, so dass diese bei einer hohen Zahl von Gruppen gleich oft auftreten sollen. Die Ausgabe in Morsezeichen beginnt mit der Spruchanfang- und endet mit der üblichen Spruchendesequenz. Die Übung lässt sich durch Drücken der Taste **P** abbrechen.

[FA Z] Ziffern Fünfergruppen

Beispiel: **#CW FA Z 10**
#cw fa z 10

Für die Ausgabe von n Fünfergruppen die nur aus Ziffern bestehen wird der Zeichensatz:

1234567890

verwendet. In der Übung werden pseudozufällig alle Ziffernzeichen ungewichtet ausgewählt, so dass diese bei einer hohen Zahl von Gruppen gleich oft auftreten sollen. Die Ausgabe in Morsezeichen beginnt mit der

Spruchanfang- und endet mit der üblichen Spruchendesequenz. Die Übung lässt sich durch Drücken der Taste **P** abbrechen.

[FA I] AFu Interpunktion Fünfergruppen

Beispiel: **#CW FA I 15**

#cw fa i 15

Die Ausgabe von n Fünfergruppen die nur aus den Standard Interpunktionszeichen für den internationalen Amateurfunk bestehen wird durch den Parameter **I** ermöglicht. Gruppen bestehen dann nur aus den Zeichen:

/. , : ? - + = () ' ! ; _ @ % \$ # & " \$ Ä Ü Ö

Die Ausgabe in Morsezeichen beginnt mit der Spruchanfang- und endet mit der üblichen Spruchendesequenz. Die Übung lässt sich durch Drücken der Taste **P** abbrechen.

[FA S] Sonderzeichen Fünfergruppen

Beispiel: **#CW FA S 3**

#cw fa s 3

Für die, die schon alles andere geübt haben und gegen Langeweile erfolglos ankämpfen gibt es noch den erweiterten Sonderzeichenmodus. Hier bestehen die Gruppen nur aus den Zeichen:

/. , : ? - + = () ' ! ; _ @ % \$ # & " \$ Ä Ü Ö

Auch hier beginnt die Übung wieder mit einer Spruchanfangssequenz und wird mit der üblichen Spruchendesequenz abgeschlossen. Und natürlich lässt sich auch diese Übung durch Drücken der Taste **P** nach einer vollendeten Gruppe abbrechen.

[FA X] Extended Fünfergruppen

Beispiel: **#CW FA X 30**

#cw fa x 30

Zum Abschluss gibt es auch noch eine komplette Übung mit allen Buchstaben, Ziffern und Sonderzeichen. Hier werden die Zeichen wieder pseudozufällig ausgegeben wobei es bei Buchstaben und Ziffern schon einmal vorkommen kann, dass „schwierig aufzunehmende“ Sequenzen plötzlich auftreten.

ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890
/. , : ? - + = () ' ! ; _ @ % \$ # & " \$ Ä Ü Ö

Es gelten auch hier wieder die gleichen Anmerkungen wie bei den anderen Fünfergruppen Übungen.

[K] CW Paris

Beispiel: **#CW K**
#cw k

Dieses Kommando gibt 20 mal das Kalibrationswort „PARIS“ mit 20 wpm, Standardabständen und ohne Farnsworth- bzw. Dash- Verlängerungen aus. Eingestellte (permanente) CW-Parameter werden nicht verändert und bleiben erhalten.

Die Zeitkonstante des Heltec Boards kann durch das Anklicken der Kontrollbox „Zeitstempel anzeigen“ (links unten in der Arduino IDE Konsole) ermittelt werden. Die Ausgabe lässt sich mit der Taste **P** jederzeit unterbrechen. Auf der Konsole werden dahs und dits in Morseschrift gezeigt.

Beispiel: **#cw k**

```
17:14:24.459 -> Start 20 mal 'Paris' mit Standardpa-
parametern für 20 wpm ausgeben

17:14:24.459 -> .---. .- .-. .. ...
17:14:27.458 -> .---. .- .-. .. ...
17:14:30.441 -> .---. .- .-. .. ...
« die weiteren „PARIS“ Ausgaben »
17:15:21.170 -> .---. .- .-. .. ...
17:15:24.159 -> Ende: 20 mal Paris mit 20 wpm
```

Die Abweichung beträgt 300 msek was durchaus akzeptabel ist.

[L] CW Lautstärke einstellen

Beispiel: **#CW L** <n>
#cw l <n>

Bereich: 1 ... 64

Die Lautstärke des CW-Signals lässt sich mit diesem Kommando einstellen wenn keine CW-Zeichen ausgegeben werden.

[M] CW Mixed

Beispiel: **#CW M** <n>
#cw m <n>

Bereich: n > 1 Gruppen

Nach einem Spruchanfang (*vv) werden in einer unendlichen Schleife aufeinanderfolgend 3*n-Standard Amateurfunkfunkabkürzungen gefolgt von n-Fünfergruppen ausgegeben. Die Ausgabe lässt sich mit der Taste **P** jederzeit beenden. Zur Überprüfung der Übung werden die gegebenen Abkürzungen auf dem Display einzeln und als fortlaufende Liste sowohl seriell als auch über Bluetooth ausgegeben. Auf der Konsole werden zusätzlich dahs und dits in Morseschrift gezeigt.

Beispiel: **#CW M 4**

```
...- ...- ...- -.-.-
Übung Afu Abkürzungen:
irc
.. .-. -.-.
88
----. ----.
hi
.... ..
Übung Fünfergruppen:
KWOY
-. .- - - -.- -.- .- - - -.-
Übung Afu Abkürzungen:
irc
.. .-. -.-.
```

[P] CW Parameter ausgeben

Beispiel: **#CW P**
#cw p

Die Parameter bzw. Einstellungen für den Codegeber lassen sich mit diesem Kommando darstellen.

Beispiel: **#CW P**

```
CW Einstellungen:
[b] Geschwindigkeit   = 121 BpM
[w] Geschwindigkeit   = 30 wpm
[s] Farnsworth Spread = 22 dits
    Longerdash        = 0 dits
[l] Lautstärke        = 32 (max 64)
```

Nur die Parameter in [] lassen sich verändern.

[S] CW Farnsworth Spread setzen

Beispiel: **#CW S <n>**
#cw s <n>

Bereich: 1 ... 200 dits

Bei der Farnsworth-Spreading (Spreizung) werden die Zeichen mit einer höheren Geschwindigkeit gesendet und die Pause zwischen den Zeichen künstlich verlängert. Beispielsweise gibt man die Zeichen mit einer

Geschwindigkeit von 20 WPM und lässt aber genügend Zeit zwischen den Zeichen um die Geschwindigkeit auf 10 WPM zu verlangsamen. Dies erreicht man durch zusätzliche Pausen, die in dits angegeben werden. Gemäß ursprünglicher Definition entspricht ein Farnsworth-Spread von 1 dem normalen Zeichenabstand ohne zusätzlich eingefügte Pause.

Beispiel: **#CW S 22**

#CW P

(Ergebnis siehe Kasten links)

[W] CW Geschwindigkeit in wpm setzen

Beispiel: **#CW W** <wpm>

#CW W <wpm>

Bereich: 5 ... 51 wpm

Bei der ersten Initialisierung des Gerätes werden 13 wpm also 65 Zeichen pro Minute als Default Wert eingestellt. Mehr als 51 wpm dürften wohl kaum bei normalen Funkamateuren vorkommen. Ist eine Einstellung in 5er Schritten beim Üben zu groß, dann kann man die Geschwindigkeit auch in Buchstaben pro Minute einstellen, siehe **#CW B**. Der eingestellte Wert wird im permanenten Flash-Speicher abgelegt und bleibt auch beim Ausschalten des FTDs bis zur nächsten Änderung erhalten.

[B] CW Geschwindigkeit in BpM setzen

Beispiel: **#CW B** <BpM>

#cw b <BpM>

Bereich: 20 ... 255 BpM

Die Geschwindigkeit der Zeichenausgabe in Buchstaben pro Minute ist gerade beim Üben mit höheren Geschwindigkeiten vorteilhaft. Eine Änderung in Word per Minute kann auch mit dem Kommando **#CW W** erfolgen. Der eingestellte Wert wird im permanenten Flash-Speicher abgelegt und bleibt auch beim Ausschalten des FTDs bis zur nächsten Änderung erhalten.

[] CW Kommandos Anzeigen

Beispiel: **#CW**

#cw

Ohne Angabe eines Parameters werden die möglichen Kommandos angezeigt.

Für den LoRa™ Einsatz werden für verschiedene geographische Regionen separate Boards für die jeweiligen Frequenzuteilungen 433/470/868/915MHz angeboten. Beim Kauf ist daher genau darauf zu achten, dass das Gerät auch wirklich für 433 MHz Sendebetrieb ausgelegt ist und es sich mindestens um die Version V.2 handelt [Heltec2020], Bild 11. Nur diese und die Folgeversionen besitzen laut Datenblatt ein separates und geschirmtes LoRa™ Tranceiver Submodul für den SX1278. Wie bereits vom RA-02 Modul bekannt, hat auch dieses Board für LoRa™ einen uFL/IPEX Antennenanschluss [IPEX].



Bild 11: Heltec ESP32 LoRa V.2 Modul Vorder- und Rückseite (Quelle: [BildHeltecs])

Batteriemangement und Anschluss des Akku:

Wie bereits in der Einleitung erwähnt, haben alle Heltec Boards ein integriertes Batteriemangement. Zum Betrieb und zum Laden befindet sich auf der Rückseite eine kleine JST-PH 1.25 mm Buchse. An diese kann ein kleiner $\approx 3,7$ Volt LiPo/Li-Ion Akku angeschlossen und direkt über die USB-Buchse geladen werden. Da es sich hierbei um JST-PH Steckverbinder mit 1,25 mm Stiftabstand ohne Rastfährchen handelt [JST125] sind diese nicht mit Standard PH Verbindern, die einen Stiftabstand von 2,0 mm haben und an den gängigen Fernost-Akkupacks zu finden sind, kompatibel. Passende Steckverbinder findet man meist nur im Modellbau oder auf Handelsplattformen aus Fernost. Weiterhin ist auf die Polungsfarben bei vorkonfektionierten Fertiggabeln zu achten. Die rote Leitung des Kabels muss nicht immer dem Pluspol auf der Platine entsprechen. Es muss daher unbedingt die Polung auf der Leiterplatte geprüft und mit der Belegung des Akkukabels verglichen werden. Eine Verpolung führt zur Zerstörung des Heltec Moduls! Neben der Buchse ist daher ein kleines Pluszeichen aufgedruckt. Für das Lademanagement wird auf dem Heltec

Board der Batterielade-IC TP4054 mit dem SMD Code: LTH7 verwendet [TP4054]. Der Ladestrom ist mit dem Widerstand RProg fest eingestellt. Im Schaltplan des Heltec Boards ist dieser Widerstand mit 10 K Ω angegeben und mit R9 bezeichnet. Der sehr kleine SMD Widerstand verbindet den PROG-Pin des TP4054 direkt mit der Massefläche. Mit 10 K Ω und $I_{CHG} = \frac{1000 \text{ Volt}}{R_{Prog}}$; ergibt sich ein Ladestrom von 100 mA. Durch Änderung von R9 lässt sich rechnerisch ein Ladestrom ICHG von bis zu 800 mA einstellen. Realistisch ist aber nur ein Maximalwert von 180 mA, weil sonst die dünne Leiterbahn Schaden nimmt. Weil es mindestens zwei unterschiedliche Platinenlayouts für das Heltec Modul gibt und hierbei der TP4054 entweder auf der Vorder- oder Rückseite der Platine angebracht ist, kann für die Modifikation hier keine weitere Hilfestellung gegeben werden.

Probleme mit Power Banks:

Nach der Programmierung des Heltec Moduls kann dieses sofort genutzt werden. Beim Einschalten bzw. nach dem Einstecken der Power Bank bleibt beim Heltec Modul das Display oft dunkel und die Startzeile wird nicht angezeigt. Durch kurzes Drücken des Reset Tasters (RST) auf dem Modul wird das Problem behoben und die Startup Meldung erscheint nach etwa einer halben Sekunde im Display des LoRa™-FTD.

Brownout Detection:

Weiterhin ist die Spannungsüberwachung (Brownout Detection) des Boards etwas 'shaky' was sich durch plötzliches Mehrfachbooten bemerkbar macht. Fällt die Versorgungsspannung am USB unter 4.85 Volt, so löst die Brownout Detection einen Restart aus. Dies passiert beispielsweise wenn die USB Powerbank fast leer ist und geladen werden muss. Dies sollte aber nicht mit dem automatischen Abschalten einer Powerbank verwechselt werden, die immer dann auftritt, wenn die Mindeststromabnahme aus der Powerbank unterschritten wird. Da das Heltec Board für die meisten Powerbanks eine zu geringe Stromaufnahme aufweist kommt dies häufig vor.

Gehäuse

Das Gehäuse für den LoRa™-FTD ist als .stl File für den 3-D Druck im gleichen Verzeichnis wie die Firmware Datei verfügbar. Am problemlosesten ist der Druck mit PLA wobei ein sehr festes, passgenaues Gehäuse entsteht. Stabiler aber leicht nachgebend ist ein 3D-Druck mit ASA Material.

Das Design, Bild 12, ist eine Weiterentwicklung des Gehäuses für das Heltec WiFi LoRa™ Board von Makoto Schoppert, welches er seit Januar 2021 in thingiverse.com zum Download zur Verfügung stellt. Den ursprünglichen Entwurf [6] findet man unter: <https://www.thingiverse.com/thing:4729713>.

Das Design von Makoto Schoppert diente als Vorlage für das Gehäuse für den LoRa™-FTD. Es wurde mit Autodesk FUSION 360 so überarbeitet, dass neben dem Heltec Board nun auch ein handelsüblicher 3,7 Volt 1100mAh Lithium-Ionen Akku (Typ: 1C/3C) mit den Abmessungen 47 x 28 x 8 mm mit Micro JST 1.25-Stecker sowie in der Gehäuseoberkante ein Drehenkoder, ein Ein-Aus Kippschalter, ein SMA-Anschluss und seitlich eine Kopfhörerbuchse Platz finden. Im Gehäuse ist, wie in Bild 7 gezeigt, auch noch genügend Platz für z.B. eine kleine

Lochrasterplatine mit den notwendigen Pull-Up Widerständen für einen Encoder und einen Taster. Für den Drehenkoder sind zusätzliche Stützen eingebaut, so dass dieser beim Betätigen des Drucktasters nicht in das Gehäuse rutschen kann. Der Encoder kommt jedoch für die bisherigen Firmwareversionen noch nicht zum Einsatz.

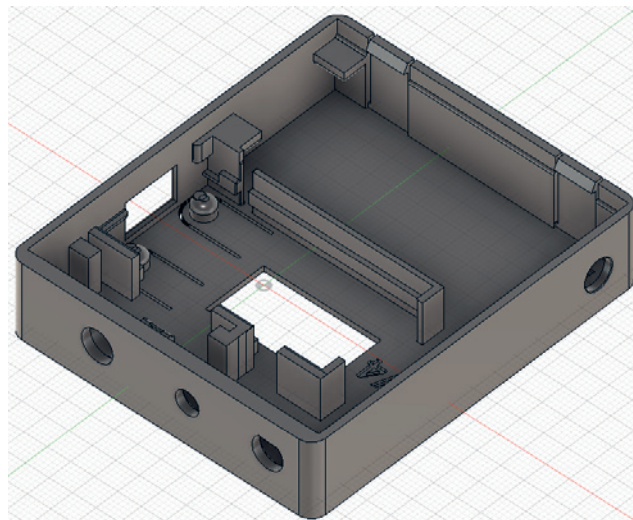


Bild 12: Das Gehäuse für den LoRa™-FTD mit Halterung für den Heltec ESP32 LoRa V.4.

Aufbau

Der Aufbau des Field Test Devices gestaltet sich sehr einfach. Da das Gerät mit oder ohne einem optionalen Drehenkoder aufgebaut werden kann gibt es für den 3-D Druck leicht unterschiedliche Gehäuse. Im Ordner mit den Files befindet sich eine aktuelle Datei mit der Beschreibung der Teile. Jedes Gehäuse besteht aus einer Oberschale und einem Bodendeckel. Hat man sich die Teile besorgt oder ausgedruckt, dann entfernt man zuerst sorgfältig die Kunststofffäden des Drucks von der Oberschale und vom Bodendeckel. Dann kann man mit dem Zusammenbau beginnen. Die beiden langen Schlitzte des Bodendeckels, in Bild 13 rot ausgeführt, sind über der viereckigen Displayaussparung anzuordnen.

Am oberen Rand des Bodendeckels, die Seite auf der sich die beiden kleinen Blöcke befinden, die dazu dienen den Drehenkoder zu halten, ist ein Klemmrand bzw. Falz ausgeführt welcher vorsichtig in die Klemmleiste der Oberschale (blau) eingesteckt wird. Dieser sollte bündig einrasten – ein Nachschneiden mit einem Rapido Messer ist nur ganz selten notwendig. Vor man etwas nachschneidet bitte genau prüfen ob dies auch wirklich notwendig ist. Bei über 40 Gehäusen war dies bisher nur ein einziges Mal bei der Fertigung mit ASA Material notwendig.

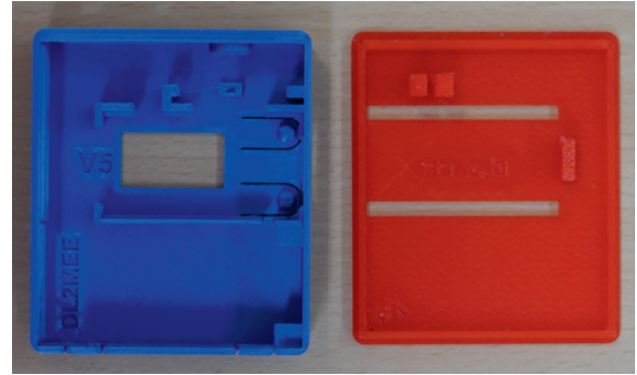


Bild 13: Oberschale (blau) neben Bodendeckel (rot).



Bild 14: Der Bodendeckel rastet zum ersten Mal ein.

Nach dem Ineinanderstecken, Bild 14, lassen sich Druckteile unter leichtem Druck zusammen “schnappen”. Danach sind beide Teile idealerweise spaltfrei verbunden und sitzen ganz fest, Bild 15.



Bild 15: Bodendeckel und Oberschale sind nun zu einem Gehäuse verbunden.

Nach Sichtprüfung muss man die Gehäuseteile nun wieder trennen. Auf der Unterseite befinden sich zwei rasende Laschen – diese dürfen auf keinen Fall verbogen werden da die Laschen sonst abbrechen.

Das erste Trennen des Gehäuses ist etwas schwierig, da die Klemmfalze anfangs noch sehr streng sitzen. Daher ist etwas Geduld und vorsichtiges Arbeiten geboten. Durch leichtes Verbiegen des Gehäuses entsteht ein kleiner Spalt auf einer der Längsseiten. In diesen Falz geht man mit dem Daumnagel hinein und zieht mit dem Daumen an der unteren Falzkante entlang bis das

Gehäuse wieder aufspringt und man den Bodendeckel leicht nach unten herausziehen kann, Bild 16. Idealerweise verwendet man hierzu keine Werkzeuge um die Gehäuseteile nicht zu beschädigen.

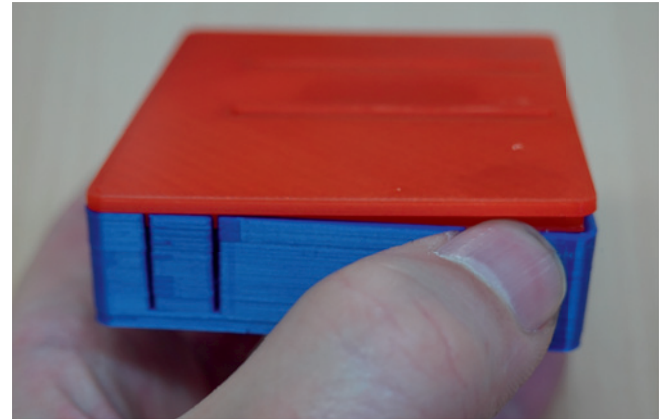


Bild 16: Trennen der Gehäuseteile – dabei nicht auf die Laschen drücken.

Das Gehäuse sollte jetzt vor dem Beginn weiterer Arbeiten mehrere Male (3) auseinander und wieder zusammensteckt werden damit sich die Kanten leicht abschleifen. Passt das Gehäuse spaltfrei zusammen, dann kann man mit dem Einbau der elektrischen Komponenten beginnen.

Grundverdrahtung

Da das Gehäuse als 3D-Druck mit einem Schmelzprozess erzeugt wurde sind die Befestigungslöcher für Antenne, Schalter, Drehenkoder und Kopfhörerbuchse oftmals nicht passgenau und müssen ggf. nachbearbeitet werden. Um Festzustellen wie viel nachbearbeitet werden muss steckt man die Bauteile zuerst nur von außen durch die Öffnungen (Bild 17) und weitet diese, falls notwendig, mit einem Rapido Messer oder einem Bohrer vorsichtig auf.

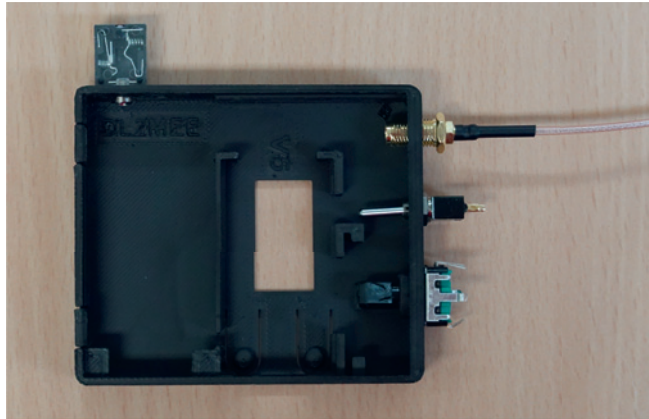


Bild 17: Alle Komponenten passen in die Bohrungen.

Die SMA-Durchführung ist in der Druckebene verdrehsicher eingebracht und die Abflachung kann daher mit dem Rapidomesser leicht Fadenebene-für-Ebene abgetragen werden. Die SMA-Durchführung sollte „streng“ passen damit sich diese beim Wechseln einer Antenne nicht leicht löst. Nun entnimmt man die Bauteile wieder und schneidet einen Streifen selbstklebende Kupferfolie von 14,8 x 90 mm zurecht. Als Folie eignet sich sog. Kupfer Schneckenband mit dem man übrigens auch sehr gut Leiterplatten reparieren kann.

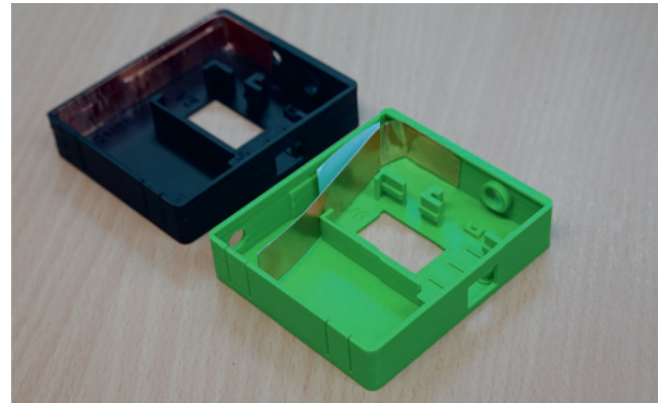


Bild 18: Einkleben des Folienstreifens.

Den vorbereiteten Streifen knickt man bei 25 mm nach hinten ab. Bis zu diesem Knick zieht man die Schutzfolie ab und klebt diesen Teil von der Antennen-/Schaltseite an der Innenwandung entlang in das Gehäuse ein, Bild 18. Eine Pinzette kann bei diesem Arbeitsschritt hilfreich sein. Dann kann man den Rest der Schutzfolie abziehen und den Streifen glattstreichen. Anschließend schneidet man die Löcher von innen frei und trennt den Überhang über die Falzkante ab (Bild 19), so dass der Bodendeckel wieder passgenau einrasten kann.



Bild 19: Korrekturschneiden des Folienstreifens.

Zum Anschluss einer Antenne an den uFL/IPX Druckknopfverbinder auf dem Heltec-Board verwendet man

ein vorkonfektioniertes ca. 10cm langes uFL/IPX auf SMA Female Kabel, Bild 20. Das Aufstecken des uFL Steckers erfordert etwas Geschick.

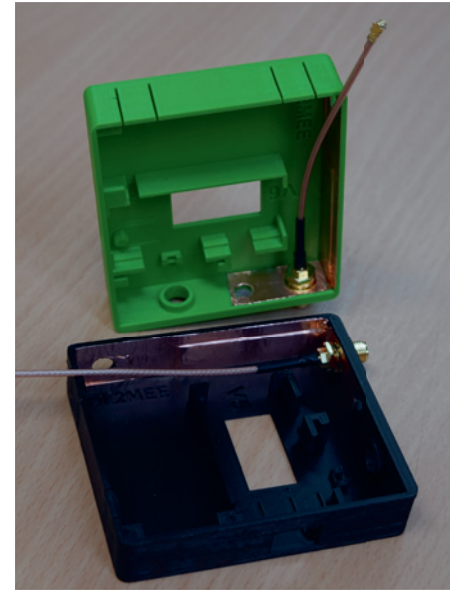


Bild 20: Eingebaute SMA-Buchse mit uFL.Pigtail

Mit dem Einsetzen der SMD-Buchse beginnt der Einbau der restlichen Komponenten. Falls man eine zweiteilige SMD-Einbaubuchse verwendet, so ist die kleine Mutter

die den Innenpin hält, sicherheitshalber leicht anzuziehen um ein späteres Lösen zu verhindern. Die Lithium Ionen Batterie wird über einen Schalter (z.B.: Reichelt MS 244) mit dem Heltec Board verbunden und kann über den USB daher leider nur im Betrieb geladen werden. Zum „Schnellladen“ siehe auch *Quick Charge* auf Seite 39.

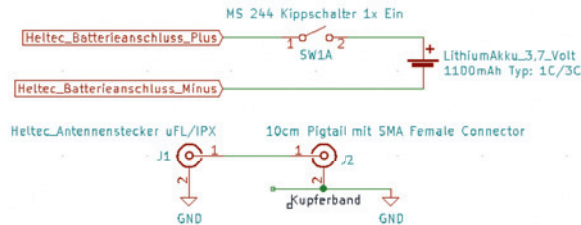


Bild 21: Grundbesaltung des FTD

Wie im Schaltplan (Bild 21) gezeigt, wird der Kippschalter nur in die bestehende Leitung zum Steckverbinder eingefügt, Bild 22. Zur Bestimmung der Drahtlänge legt man den 1C/3C 1100 mAh Akku in das Gehäuse ein und schneidet die Plus-Leitung an geeigneter Stelle auf.

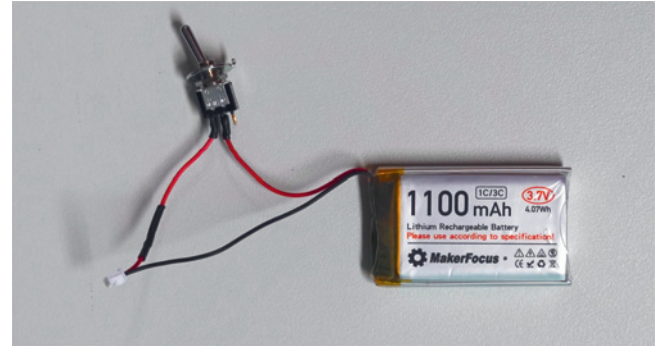


Bild 22: Batterie und Schalter

Da der Schalter eine Beilegescheibe mit Verdrehsicherung hat ist im Gehäuse eine Aussparung für den kleinen Flachnippel vorgesehen. Eventuell ist dieser mit einer Flachzange minimal nachzubiegen damit der Schalter optimal in Bohrung passt. Die Beilegescheibe mit einer kleinen „Nase“ legt auch fest, wie herum der Schalter eingebaut werden muss damit der unverlötete Pin zum Gehäuse, also nach „unten“ zeigt, Bild 23.

Vor dem Anstecken an das Heltec Board sind die Hinweise im Abschnitt zu „Batteriemanagement und Anschluss des Akkus“ (Seite 74) unbedingt zu beachten, da eine Verpolung zur Zerstörung des Heltec Boards führt!

Soll die Erweiterung mit einem Drehenkoder durchgeführt werden (siehe Seite 85), so empfiehlt es sich

diesen vor dem Einsetzen des Heltec Boards einzubauen, da ansonsten dieses später wieder ausgebaut werden muss.

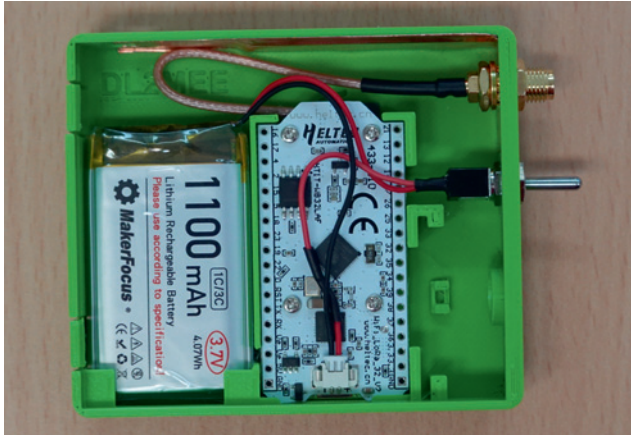


Bild 23: Die fertige Grundverdrahtung des LoRa™-FTD

Mit der Grundverdrahtung und einer angeschlossenen Antenne können alle LoRa™-Funktionen des FTDs genutzt werden.

Erweiterung für CW

Der Morsetrainer ist in der Firmware bereits enthalten und zum Betrieb benötigt man nur einen Kopfhörer der über einen Transistor an den GPIO Pin 25 angeschlossen werden kann, Bild 24.

Achtung: Der Anschluss ist nur für einen Kopfhörer ausgelegt! Der Ausgang liegt an Spannung an. Das kann bei angeschlossenen Geräten zu Beschädigungen führen!



Bild 24: Eine Version des CW-Trainers mit angeschlossenem Kopfhörer

Man kann praktisch jeden kleinen NPN Transistor verwenden, z.B.: BC107, C517, c172, c547, 546, c550 usw. Die 3.3V am Heltec Board bei Akku/Batteriebetrieb ansonsten geht auch 5V am Heltec Board.

Als Kopfhörerbuchse, auch Audio Jack genannt, wurde für dieses Gerät eine günstige 3,5mm Stereo Klinkeneinbaubuchse (z.B. Reichelt EBS 35) gewählt. Da kein Schaltkontakt notwendig ist kann auch jede andere Buchse aus der Bastelkiste verwendet werden. Im 3D-druckbaren Gehäuse ist eine Bohrung von 6,2mm Durchmesser für die Durchsteckmontage vorgesehen. Die Leitungsführung eines möglichen Aufbaus auf einer Lochrasterplatine und die dazu gehörende Verdrahtung

Lötseite:

Kugel

Kette

Kugel

Masse

Verbindungen

Anschluss
Ohr/Kopfhörer

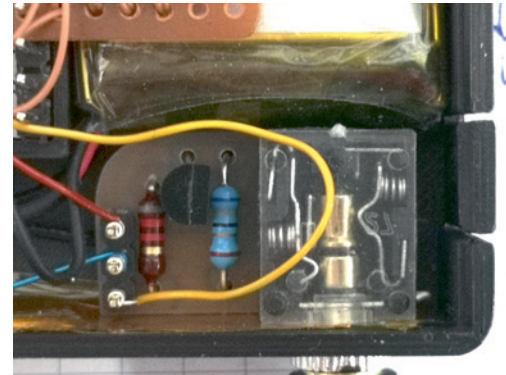
68Ω

E

+3,3V

Masse

ir



83

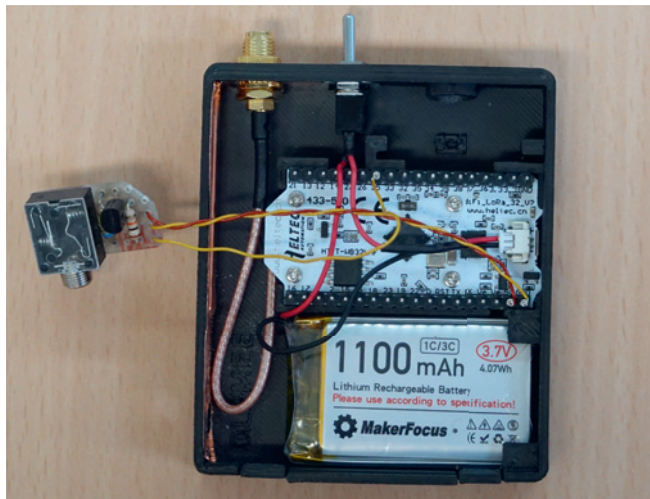


Bild 27: Ein anderer Platinenentwurf – Platine vor dem Festschrauben im Gehäuse

Nach dem Einbau der Platine ist darauf zu achten, dass der uFL Antennenstecker sich nicht versehentlich vom Heltec Modul gelöst hat und das Modul dann ohne Antenne betrieben wird.

Zum Test der Zusatzschaltung steckt man einen normalen Kopfhörer (Impedanz ca. 16 – 30 Ohm) mit 3,5 mm Klinke an und gibt z.B.:

#CW A 100

ein. Ist alles korrekt aufgebaut, dann hört man in Telephonie AFu Abkürzungen. Mit einem Oszilloskop an GPIO Pin 25 kann man die Zeichen mit einer quasi-Sinuswelle ansehen.

Achtung: Der Audio Ausgang des FTDs ist nur für einen Kopfhörer gedacht – nichts anderes anschließen! Andere Geräte gehen kaputt!

Erweiterung mit einem Drehekoder

Wie bereits beschrieben wird der Drehekoder in den bisherigen Firmwareversionen noch nicht genutzt. Die Beschreibung ist trotzdem bereits jetzt in diesem Manual enthalten, da einige Nutzer das in das Gehäuse eingebaute Heltec Board für eigene oder andere Projekte nutzen und je nach Anwendung einen Drehekoder oder einen Taster benötigen. Eine beliebte Anwendung scheint hier ein APRS i-gate der Amateurfunkgruppe in Wien zu sein – genaueres kann ich aber noch nicht darüber berichten da ich die Software selbst noch nicht übersetzt bzw. ausprobiert habe. Die Schaltung zum Anschluss des Encoders ist wieder sehr einfach.

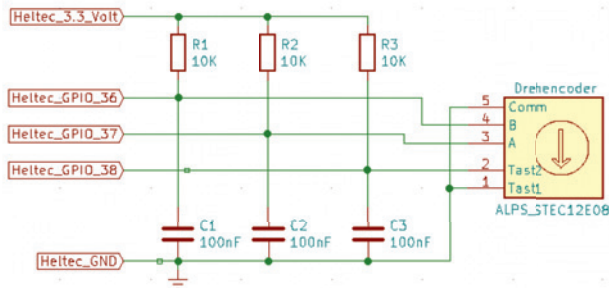


Bild 28: Schaltung zum Anschluss des Drehekoders für dasLoRa™-FTD.

Jeder Kontakt wird mit einer RC-Kombination bei der Entprellung unterstützt und gleichzeitig um die an den GPIOs fehlenden² Pull-Up Widerstände ergänzt, Bild 28. Die Widerstände R1, R2 und R3 für den Encoder und den Taster kann man z.B. wieder auf einer kleinen Lochrasterplatte unterbringen. Im dargestellten Gerät wurden die Kondensatoren C1, C2 und C3 direkt auf die Kontakte des Encoders aufgelötet, Bild 29.

² Streng genommen fehlen diese an den hier verwendeten GPIOs nur bei einigen Derivaten des CPU Chips. Um sicher zu gehen, dass Taster und Geber immer funktionieren sind hier Pull-Ups vorgesehen.

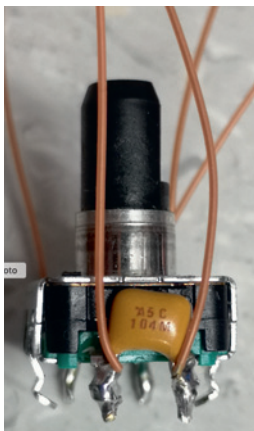


Bild 29: C3 direkt am Encoder aufgelötet.

Die kleine, frei „fliegenden“ Lochrasterplatine mit den Pullups ist in Bild 30 gut zu sehen. Vor dem Schließen des Gehäuses kann man die freien Löcher dazu nutzen die Platine, ohne elektrische Verbindung, auf den unbelegten Steckpfeften des Heltec Boards aufzustecken und dadurch gegen Bewegung zu sichern.

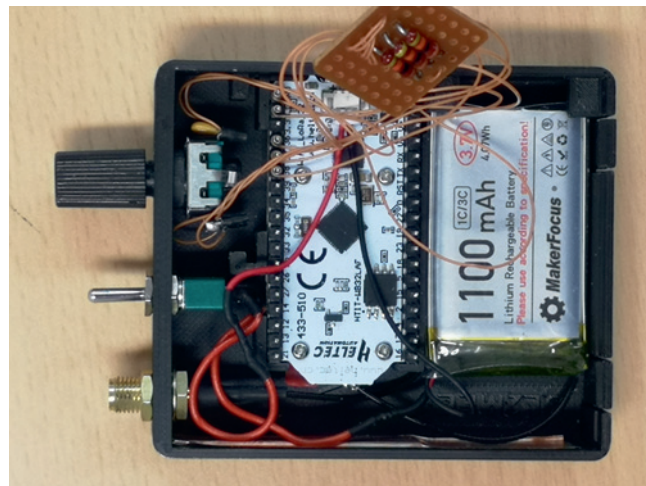


Bild 30: Der geöffnete LoRa™-FTD mit Wire-Wrap „Luftverdrahtung“ (es geht auch besser).

Eine weitere Möglichkeit den Drehencoder zu verdrahten besteht darin, eine kleine Platine mit 1206 SMD Bauteilen direkt auf den Encoder aufzusetzen und mit der Heltec Platine zu verbinden. Ein Platinen Vorschlag ist im Bild 31 und der Bestückungsplan in Bild 32 gezeigt. Die Unterlagen hierzu sind ggf. im Folder „KiCad Drehencoder Platine“ hinterlegt.

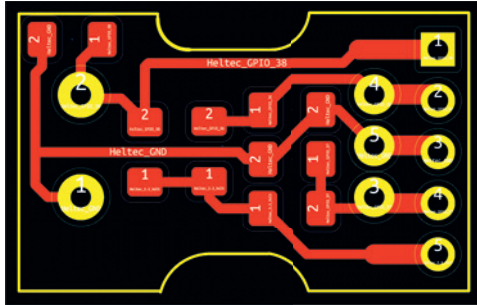


Bild 31: Platinenlayout für die Zusatzschaltung für den Drehkoder.

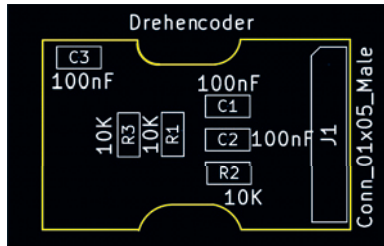


Bild 32: Bestückungsplan für die Drehkoderplatine.

Der Wert der Pull-Up Widerstände ist nicht kritisch, hier sind 12 kOhm SMD Widerstände (Bezeichnung: 123) auf die gefräste Platine aufgelötet, Bild 33. Zeichnet man eine eigene Platine so ist darauf zu achten, dass die kleinen Halteblöcke am Gehäuseboden die Platine beim Betätigen des

Schalters auch korrekt gehalten können bzw. nicht zu weit hineinragen.

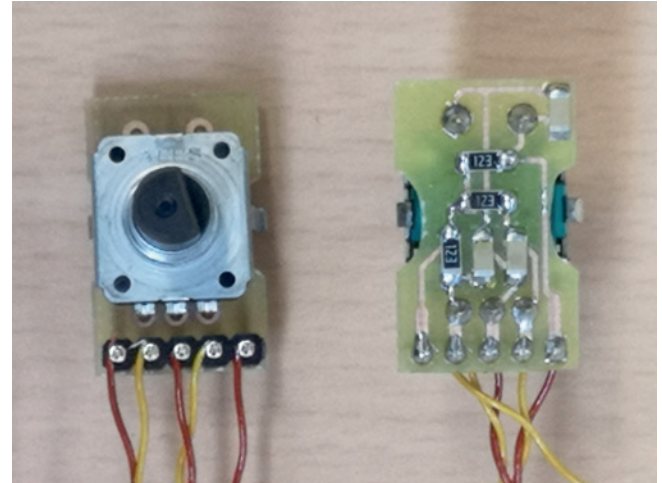


Bild 33: Platine für Pull-Ups und Stützkondensatoren.

Ist die Platine mit dem Drehgeber in das Rundloch ganz eingeschoben, so rastet eine Blechzunge in die vorgesehene Vertiefung der Gehäuseoberschale klickend ein, Bild 34. Damit wird der Drehgeber in zwei Achsen fixiert.

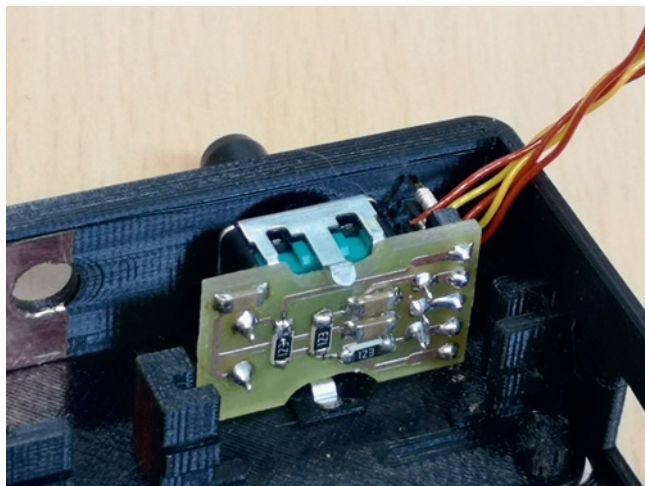


Bild 34: Der in die Gehäuseoberschale eingerastete Drehgeber mit aufgesetzter Interfaceplatine.

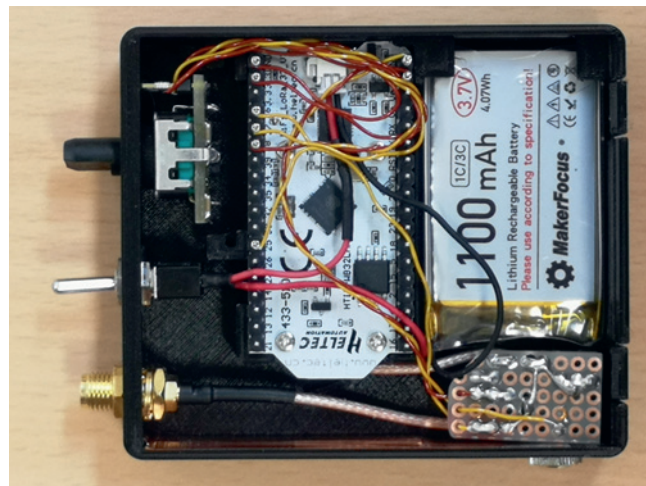


Bild 35: Das komplett aufgebaute LoRa™ Field Test Device.

Der eingebaute Drehenkoder (Bild 35) lässt sich mit dem Beispielprogramm³ auf Seite 94 überprüfen. Drückt man in dem Programm den Taster, dann werden Sternchen bzw. ein Zählerstand für den Drehgeber über die serielle Schnittstelle ausgegeben.

³ Das Programm ist natürlich vor der Nutzung mit der Arduino IDE zu übersetzen und auf das LoRa™-FTD zu laden.

Bauteileliste und Bezugsquellen

- Heltec Board z.B.: Amazon, AliExpress.com, usw.
Wifi_Lora_32_V2 für 433 MHz

Bauteile z.B.: von Reichelt Elektronik

- Drehenkoder mit Taster 24 Pulse und 24 Rastungen
Typ: ALPS **STEC 12 E 08**
- Passender Drehknopf Ø 10mm, h 15mm
Typ: **KNOPF 10-150E**
- Kippschalter 1 x Ein-Ein
Typ: **MS244** bzw.: 1S-24L244
- 3 x Kondensator **100nF**
- 3 x Widerstand **10KOhm**
- evtl. Pfostenleiste 5x1, Platine, WW-Drähte
- Pigtail Kabel z.B.: Eckstein GmbH
RF Adapter Cable SMA Female Connector to uFL/u.FL/IPX/IPEX MHF
U.FL 1.13
- Lithium Ionen Akku: 3,7 Volt Typ: 1C/3C
Amazon Suche z.B. nach:
„Akku 3,7 Volt 1C 3C 1100mAh Micro JST“
- Kupferfolienstreifen selbstklebend z.B.: Conrad
Anregung von DG1LE: ein Streifen selbstklebendes „Schneckenband“.
- Antenne für das 70cm Band, Kleinmaterial zur Verdrahtung

Die nachfolgend dargestellte Memory Map zeigt die Belegung der für den Betrieb relevanten Speicherbereiche des ESP-32. EEPROM (Electrical Erasable Programmable Read Only Memory) und Flash-EEPROM (Löschvorgang so schnell wie der Blitz, engl. Flash⁴) welche über Nutzerfunktionen verändert werden können. Auf Grund der limitierten Größe des EEPROMs des ESP32 Prozessors von 512 Bytes (332 davon in der FA Firmware nutzbar) enthält dieser Speicherbereich nur das permanente Rufzeichen, den Startup String und wenige nicht veränderbare Systemrelevante Informationen. Der Startup String wird bei jedem Systemstart bzw. Reset ausgeführt und kann nur wenige Befehle enthalten, siehe Seite 42 und 43. Der Flash Speicher wird beim Ersten Startup des FTD mit einem SPIFFS Filesystem (Serial Peripheral Interface Flash File System) und eigenen strukturierten Speicherblöcken initialisiert. Das SPIFFS Filesystem enthält nach der Initialisierung nur Systemfiles, die mit Kleinbuchstaben beginnen und nicht direkt durch den Nutzer verändert bzw. aufgelistet werden können.

⁴ Die Anekdote zur Namensfindung findet man z.B. unter:
<https://books.google.de/books?id=VOnyWUUUj04C&pg=SA6-PA4#v=onepage&q&f=false>

Zusätzlich können Nutzer Files, welche mit Großbuchstaben beginnen, angelegt und bearbeitet werden, siehe Systemanweisungen ab Seite 36.

In der folgenden Memory Map Tabelle sind die jeweiligen Befehle in der Spalte „Änderbar mit:“ angegeben. Unter dem jeweiligen Befehl findet man auch weitere Informationen zur Funktion.

Memory Map des LoRa™- FTD

Bezeichner Nutzer	Variable im Programm	Typ	Eintrag on Startup Initialisierung (#*13) bzw. Beispiel	Default bei z.B.: fehlender Eingabe	ROM	EEPROM	Flash Memory SPIFFS Systemfile /f.txt	Einstellung nach Neustart wieder sofort verfügbar	Flash Memory SPIFFS User File /A.txt	Prüfbar mit	Änderbar mit:	Bereich	Anmerkung
CHIPID	sysid	List	0D-23-AE-32-FE-EC		*			*					Nicht veränderbar
MACADDR	deviceid	uint64_t	0x04524CA8CC84		*			*	#*32				Nicht veränderbar
VERSION*	Version	String	E3_17		*		*	*	#*77, #CAT				Nicht veränderbar
COMPDATE	compile_date[]	String	Feb12 2022 19:20:18		*		*	*	#*77, #CAT				Nicht veränderbar
CALLSIGN	Callsign	String	DL2MEE	"NO-CALL"		*	*	*	#*77, #CAT, #PL	#RP			Muss nach Erststart eingeben werden
STARTUPS	startupstring	String		#:		*	*	*	#*77, #CAT	#X			Sollte nach Erststart eingeben werden
FREQUENC	frequency	Integer	439700000				*	*	#*77, #CAT, #PL	#F		144 - 146MHz und 430 - 440MHz	
TXPOWER*	TXpower	Integer	5				*	*	#*77, #CAT, #PL, #L	#L		2 ... 20dBm	Im 2m Band nur 2...5dBm
SYNCByte	sync	Byte	115				*	*	#*77, #CAT, #PL, #SY	#SY		0x00-0xFF	Manche Bausteine erlauben 0x00 nicht
SPREADFA	SpreadingFactor	Integer	10				*	*	#*77, #CAT, #PL, #SF	#SF		6 ... 12	
CODINGRA	CodingRate	Integer	5				*	*	#*77, #CAT, #PL, #CR	#CR		5 ... 8	
BANDWIDT	SignalBandwidth	Long Integer	125000				*	*	#*77, #CAT, #PL	#BW		0 ... 9	Im 2m Band nur 0 ... 7
PREAMBLE	PreambleLength	Integer	8				*	*	#*77, #CAT, #PL, #PA	#PA		6 ... 65535	
GAIN***	Gain	Integer	1				*	*	#*77, #CAT, #PL, #G	#G		1 ... 6	
IQREVERS	IQInvert	String	DIS	DIS			*	*	#*77, #CAT	#IQ ...		DIS, ENA, ERX, ETX	
CRCHECK	CRC	String	DIS	DIS			*	*	#*77, #CAT	#CRC ...		DIS, ENA	
BAKETEXT	BakenText	String	LoRa-Versuchsbake				*	*	#*77, #CAT, #BT	#BT		max. 220 Zeichen	Groß- und Kleinschreibung
BANEINTE	BakenIntervall	Integer	7				*	*	#*77, #CAT, #BI	#BI		5 ... 86401	
CWCLOCKS	CW_cwclocks	Integer	149	20		*	*	*	#*77, #CAT	#CW W		388 ... 38 (\$... 51wpm)	Änderung nur indirekt clocks=1940/W;
CWFARNSW	CW_farnsworth	Integer	5	5		*	*	*	#*77, #CAT, #CW S	#CW S		1 ... 200	
CWLAUTST	lautstaerke	Integer	32				*	*	#*77, #CAT, #CW L	#CW L		0 ... 64	
CWLONGER	longerdash	Integer	0				*	*	#*77, #CAT				Noch nicht veränderbar
CWORDDIS	cworddist	Integer	0				*	*	#*77, #CAT, #CW D	#CW D		0 ... 200	
MOFREQEA	frequencyA	Long Integer	439700000				*	*	#*77, #CAT, #MO FA	#MO FA		137MHz - 525MHz	
MOFREQEB	frequencyB	Long Integer	438400000				*	*	#*77, #CAT, #MO FB	#MO FB		137MHz - 525MHz	
MOACRONA	acronym_a	String	Fre1:				*	*	#*77, #CAT	#MO ACA		max. 4 Zeichen	
MOACRONB	acronym_b	String	Fre2:				*	*	#*77, #CAT	#MO ACB		max. 4 Zeichen	
APLATITU	Latitude	String	4806.84N	4806.84N		*	*	*	#*77, #CAT, #APRS LAT	#APRS LAT		00.xxN ... 89.xxN bzw.: 00.xxS ... 89.xxS mit xx: 00 ... 99	
APLONGIT	Longitude	String	01153.78E	01153.78E		*	*	*	#*77, #CAT, #APRS LON	#APRS LON		000.xxE ... 179.xxE bzw.: 000.xxW ... 179.xxW	
APRSID*	aprssid	String	7	"nicht Eingeben"		*	*	*	#*77, #CAT	#APRS SSID		0 ... 15	
APSYMBO	sym	String	["nicht Eingeben"		*	*	*	#*77, #CAT	#APRS SYM		nur 1 Zeichen	"Shifts" sind in LoRa noch nicht üblich.
APMESSAG	aprsmessage	String	LoRa-C11-TRX	"nicht Eingeben"		*	*	*	#*77, #CAT, #APRS STR	#APRS STR		max. 60 Zeichen	Groß- und Kleinschreibung

Beim Start bzw. Restart des FTD wird auf der Konsole in der Ersten Zeile die Fehlermeldung:

E (169) psram: PSRAM ID read error: 0xffffffff ausgegeben. Diese Fehlermeldung zeigt an, dass keine (externe) Speichererweiterung an das Heltec Board angeschlossen ist – was richtig ist. Diese Meldung ist daher zu ignorieren.

PROGRAMMIERUNG DES HELTEC MODULS

Für die Programmierung mit eigener Software kann man die Arduino IDE für das Heltec ESP32 Board einsetzen. Zur Einbindung der Heltec spezifischen Bibliotheken in die IDE folgt man am Besten der von Heltec Automation zur Verfügung gestellten ausführlichen Anleitung unter: [6]. Da das Heltec Board einen USB-UART Konverter an der Schnittstelle eingebaut hat, kann es erforderlich sein, einen entsprechenden Treiber in der Entwicklungsumgebung einzubinden [7], siehe dazu auch Seite 15.

Nützliche Bibliotheken:

Für die in der Firmware enthaltenen Funktionen wurden folgende Bibliotheken eingesetzt:

LoRa.h

Autor: Sandeep Mistry

Quelle:

<https://github.com/sandeepmistry/arduino-LoRa>

Diese Arduino-LoRa Bibliothek von Sandeep Mistry enthält alle notwendigen Grundfunktionen zum Senden und Empfangen von LoRa™-Paketen und zur Übergabe

von Parametern an den Semtech Transceiver. Das GitHub Repository enthält auch Beispiele zur Nutzung.

U8x8lib.h

Autor: Oli Kraus

Quellen:

<https://github.com/olikraus/u8g2>

<https://github.com/olikraus/u8g2/wiki/u8x8reference>

<https://github-wiki-see.page/m/olikraus/u8g2/wiki/u8x8reference>

Diese Arduino-Bibliothek ist eine monochrome Grafikbibliothek für eine Vielzahl monochromer Displays die über den Bibliotheksmanager der Arduino-IDE installiert werden kann.

Einstellungen in der Arduino IDE:

Zum Hochladen eines übersetzten Programms auf das Heltec Board sind neben den Defaults folgende Einstellungen in der Arduino IDE vorzunehmen:

Werkzeuge → Board → ESP32 Arduino → "ESP32 DEV Module"
Werkzeuge → Upload Speed → "921600"
Werkzeuge → Flash Frequency → "80MHz"
Werkzeuge → Flash Mode → "QIO"
Werkzeuge → Flash Size → "4MB (32 MB)"
Werkzeuge → Partition Scheme → "Default 4Mb with spiffs (1,2MB APP/1,5MB SPIFFS)"
Werkzeuge → Core Debug Level → "Keine"
Werkzeuge → PSRAM → "Disabled"
Werkzeuge → Port → "dev/cu/SLAB_USBtoUART"

Nach der Einstellung werden manchmal die Parameter nicht direkt übernommen und das Hochladen eines Programms ist dann noch nicht möglich. Es empfiehlt sich dann das Board kurz abzustecken und zu warten bis die Portinformation (z.B. unter Apple OSX "dev/cu/SLAB_USBtoUART") aus den Menueeinträgen der IDE verschwunden ist. Nach dem Anstecken des Boards wird erneut in der IDE ein Portscan ausgeführt und die Portinformation automatisch angezeigt. Das Hochladen von Programmen sollte nun direkt möglich

sein. Erscheint beim Kompilieren die Meldung: „Fehler in der SPI(...) Zeile“, so kann man diese Fehlermeldung ignorieren und das Programm trotzdem hochladen. Erscheint dennoch die Fehlermeldung „Beim Hochladen des *Sketches* ist ein Fehler aufgetreten“ dann trennt man das Heltec Board kurz von der USB-Schnittstelle, übersetzt erneut und lädt das Programm nochmals hoch. Sollte der Fehler jedoch „hartnäckig“ sein so kann es daran liegen, dass in einem der auszugebenden Strings noch ein Umlaut- oder ein eventuell unsichtbares Sonderzeichen steckt. Die Prüfsumme kann dann nicht richtig gebildet werden. Abhilfe schafft hier nur die Entfernung des Zeichens und eine erneute Übersetzung des Programms.

Beispielprogramm „Blink für ESP32“

Das allbekannte Blink-Programm dient zum Test ob zur Programmierung alles richtig eingestellt ist und ob die Verbindung zum Heltec ESP-32 Board funktioniert. Das Programm entstammt den Beispielprogrammen der Arduino IDE:

Datei → Beispiele → 01.Basics → Blink.

Siehe auch: <http://www.arduino.cc/en/Tutorial/Blink>. Für das Heltec Board ist im folgenden Beispielprogramm die LED Zuordnung auf PIN 25, weiße LED direkt neben der orangen Power LED, geändert.

```
const uint8_t LED_Pin = 25;
void setup() {
    // Initialisiere den Pin LED_BUILTIN als Output.
    pinMode(LED_Pin, OUTPUT); }
void loop() {
    // Endlosschleife
    digitalWrite(LED_Pin, HIGH); // LED einschalten
    delay(1000); // Eine Sekunde warten
    digitalWrite(LED_Pin, LOW); // LED ausschalten
    delay(1000); // Eine Sekunde warten
}
```

Beispielprogramm: „Drehenkoder Test“

Mit diesen Programmteilen kann man den Enkoder testen. Die Ausgabe erfolgt seriell. Zur Darstellung verwendet man am Besten den „Seriellen Monitor“ der Entwicklungsumgebung.

Der auf dem Heltec ESP32-LoRa Modul verbaute ESP32-D0WDQ6 Prozessor hat nicht an allen GPIO Pins eingebaute Pull-Up Widerstände. Angegeben sind schaltbare Pull-Ups für die GPIOs: 14, 16, 17, 18, 19, 21, 22, 23. Diese funktionieren aber bei meinen Modulen nicht zuverlässig. Daher werden hier die Pins 36, 37 und 38 verwendet. Die anderen Pins sind hierfür nicht brauchbar, da diese mit dem Display, dem LoRa™ Modul oder mit anderen I/O Devices verbunden sind, siehe hierzu auch den Abschnitt Pinout Heltec Modul auf Seite 97. Dies ist generell in den verfügbaren Unterlagen schlecht dokumentiert und evtl. auf einem anderen Board oder einer anderen Version des Heltec Boards unterschiedlich.

Das Testprogramm stammt ursprünglich von „Garry's blog“ (<https://garrysblog.com/2021/03/20/reliably-debouncing-rotary-encoders-with-arduino-and-esp32/>) auf dem Garry erklärt wie man Enkoder entprellt und mit einem Arduino nutzt. Das folgende Programmbeispiel nutzt Oleg Mazurov's Code und ist nur um wenige

Zeilen zum Test des Tastschalters erweitert. Alle zusätzlich von mir eingefügten Zeilen sind mit `//Modified` gekennzeichnet, die GPIOs wurden ohne Annotation angepasst. Mit dem Code kann ein angeschlossener Drehencoder auf Funktion geprüft werden. Die Ausgabe erfolgt auf der Konsole (z.B.: Serieller Monitor der Arduino IDE).

Programmbeispiel: Drehenkoder

```
/* Refer to the blog: https://garrysblog.com/2021/03/20/reliably-debouncing-rotary-encoders-with-arduino-and-esp32/
The solution is based on Oleg Mazurov's code for Reading rotary encoder on Arduino, here
https://chome.nerpa.tech/mcu/reading-rotary-encoder-on-arduino/ and here
https://chome.nerpa.tech/mcu/rotary-encoder-interrupt-service-routine-for-avr-micros/
Extended by Klaus Buchenrieder to allow for testing the pushbutton on the encoder.
Result shown in the "Serieller Monitor" of the Arduino IDE.

Connections
=====
Encoder | ESP32 |
-----
A        | 37      |
B        | 36      |
T        | 38      |
GND      | GND     |

*/

// Define rotary encoder pins
#define ENC_A 37
#define ENC_B 36
#define Taster 38

volatile int counter = 0;
int newlast = -1;           // Modified
int oldlast = -1;          // Modified
```

```
void setup() {

    // Set encoder pins
    pinMode(ENC_A, INPUT_PULLUP);
    pinMode(ENC_B, INPUT_PULLUP);
    pinMode(Taster, INPUT_PULLUP);           // Modified

    // Start the serial monitor to show output
    Serial.begin(115200); // Change to 9600 for Nano, 115200 for ESP32
    delay(500);           // Wait for serial to start
    Serial.println("Start");
}

void loop() {
    static int lastCounter = 0;

    read_encoder();

    // If count has changed print the new value to serial
    if(counter != lastCounter){
        Serial.println(counter);
        lastCounter = counter;
    }
    newlast = digitalRead(Taster);           // Modified
    if (newlast != oldlast)                   // Modified
    {                                           // Modified
        Serial.println("***");               // Modified
        oldlast = newlast;                   // Modified
    }                                           // Modified
}

void read_encoder() {
    // Encoder routine. Updates counter if they are valid
    // and if rotated a full indent

    static uint8_t old_AB = 3; // Lookup table index
    static int8_t encval = 0;  // Encoder value
    static const int8_t enc_states[] = {0,-1,1,0,1,0,0,-1,-1,0,0,1,0,1,-1,0}; // Lookup table
    old_AB <=< 2; // Remember previous state

    if (digitalRead(ENC_A)) old_AB |= 0x02; // Add current state of pin A
    if (digitalRead(ENC_B)) old_AB |= 0x01; // Add current state of pin B
```

```

    encval += enc_states[( old_AB & 0x0f )];

    // Update counter if encoder has rotated a full indent, that
    // is at least 4 steps
    if( encval > 3 ) {          // Four steps forward
        counter++;              // Increase counter
        encval = 0;
    }
    else if( encval < -3 ) {    // Four steps backwards
        counter--;              // Decrease counter
        encval = 0;
    }
}

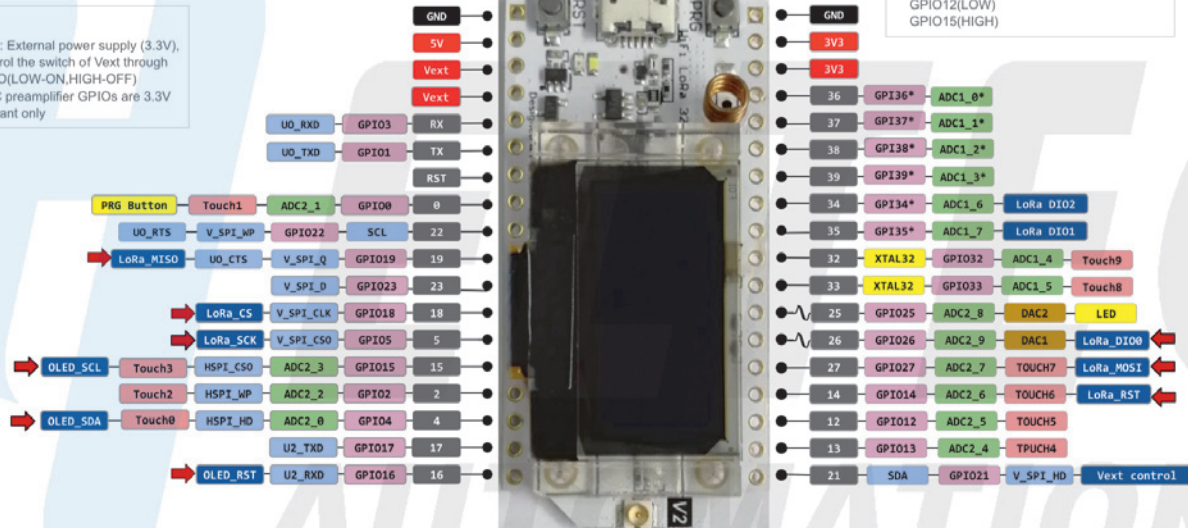
```

PINOUT DES HELTEC MODUL

WiFi LoRa 32(V2) Pinout Diagram

Notes:

- Vext: External power supply (3.3V), control the switch of Vext through GPIO(LOW-ON,HIGH-OFF)
- ADC preamplifier GPIOs are 3.3V tolerant only



Tips:

- These pins must be in the following states when downloading the program:
GPIO5(LOW)
GPIO12(LOW)
GPIO15(HIGH)

Quelle: [https://resource.heltec.cn/download/WiFi LoRa 32/WiFi LoRa 32 V2.pdf](https://resource.heltec.cn/download/WiFi%20LoRa%2032/WiFi%20LoRa%2032%20V2.pdf)

be used for other purpose unless you know what you are doing

GEPLANTE WEITERENTWICKLUNGEN

Wie bereits Eingangs beschrieben entstand das heutige FTD aus der Notwendigkeit heraus ein handliches Gerät zu haben mit dem man einfache Messungen für LoRa™ Links, Geräte und Antennen machen kann. Auf mehrfachen Wunsch der Nutzer kam dann noch mein Morsetrainer hinzu für den sich mehr und mehr OMs exklusiv interessieren, ja das Gerät sogar nur wegen des Trainers aufbauen und den LoRa™ Teil überhaupt nicht nutzen.

Seit den ersten Versionen des FTDs hat sich mein Wissen über LoRa™ und mein Interessenschwerpunkt von der reinen Entwicklung der LoRa™ Bausteine und von einfachen Transceivern hin zu komplexen Mesh Netzwerken, Lokalisierung und LoRa™ Messtechnik verschoben. Damit natürlich auch die Anforderungen an das FTD. Wie in diesem Manual an einigen wenigen Stellen zu erahnen ist, gibt es neben der Version für die Zeitschrift FUNKAMATEUR- (FA) noch limitierte experimentelle- (E) und 868MHz (ISM) Versionen des FTDs mit jeweils ganz unterschiedlichen Funktionen. Die sogenannte „ISM“ Version entwickle ich permanent für das dtec.bw Projekt ROLORAN⁵ (Resilient Operation of LoRa Networks) weiter in dem systematische Messung

durchgeführt und Analysen der Funkreichweite von LoRaWAN-Geräten in urbanen, suburbanen und ländlichen Umgebungen und die Störanfälligkeit von Übertragungen durch Störquellen untersucht werden.

Für die frei verfügbare FA Version sind folgende Erweiterungen geplant, die ich je nach Wetterlage, Lust und anderen Aktivitäten realisieren möchte (keine Reihung nach Priorität):

- Hardware Real-Time Clock für Zeitstempel beim Empfang von LoRa™ Satelliten, z.B. NORBI
- SD-Karteneinbindung für Morsetexte
- Zufällige Erzeugung von CW QSO Texten und Rufzeichen zum Üben.
- Ladbare Korrekturkurven z.B. für RSSI
- Berechnung von LoRa™ Performance Daten
- Frequency Hopping, Meshing, Store-and-Forward
- Lokalisierung über Signallaufzeiten mit mehreren FTDs
- Ermittlung bzw. Mitlesen ohne bekanntes SYN
- Nachfolgemodell mit mehr Programmspeicher und viel „cooler“ Hardware speziell für AFu.
- Vernünftige Kopfhörerverstärkerschaltung

⁵ <https://dtec.bw.de/home/forschung/unibw-m/projekt-roloran>

DANKSAGUNG

An dieser Stelle möchte ich meinen besonderen Dank der täglichen Telegraphierunde des DARC Ortsverbands München-Ost (C11) insbesondere bei Ludwig (DG1LE), Edi (DK1JU), Charles (DC5MF), Sabrina (DL9MCE), Armin (DL4ABM), Mario Silaci (SWL), Sigi Brunner (SWL) und Friedrich (DG1MKK) entgegen bringen, ohne deren Mithilfe beim Testen die ständige Verbesserung des FTDs nicht möglich gewesen wäre. Auch bedanke ich mich bei Ralf (DJ5XS) für Anregungen zur Programmierung der Monitorfunktion und Hinweise zum Drehenkoder.

Mein Dank gilt auch den Mitgliedern des dtec.bw Projekts ROLORAN⁶ (Resilient Operation of LoRa Networks) für deren freundliche Hilfe und Unterstützung bei den Messungen.

ANHANG: TELEGRAFIE – AFU ABKÜRZUNGEN

In der nachfolgenden Tabelle sind die im Morsecoder eingespeicherten AFu Abkürzungen zusammengefasst. Diese umfassen die im deutschsprachigen Amateurfunk am häufigsten verwendeten, drei im AFu nicht zulässige (xxx, QOA, QOB) Abkürzungen und die Vornamenamen der Unterstützer dieses Projekts. An dieser Stelle möchte ich mich auch noch für genau den einen „Vergeber“ entschuldigen welcher im Schnitt zufällig immer genau alle 1000 Abkürzungen wieder auftritt ;).

Als Quellen für die Abkürzungen und zur Prüfung der Richtigkeit der „Übersetzungen“ wurde vorrangig die Wikipedia.de; das Jahrbuch für den Funkamateure von Hans Schwartz, 2021; der Amateurfunklehrgang: Amateurfunk verbindet von Dirk Paulsen 2022; und die Unterlagen der Bundesnetzagentur zur Vorbereitung auf die fachliche Prüfung der Funkamateure herangezogen.

⁶ <https://dtecbw.de/home/forschung/unibw-m/projekt-roloran>

abt	about	ungefähr
ac	alternating current	Wechselstrom
adr	address	Anschrift
ads	adios (span.)	auf Wiedersehen
af	audio frequency	Tonfrequenz 3-16 kHz
afsk	audio frequency shift keying	Tonfrequenzumtastung
afu		Amateurfunk
agbp	always good brass pounding	Gruß unter alten Telegraphisten: "Gutes Messingklopfen".
agc	automatic gain control	automatische Verstärkungsregelung
agn	again	wieder, nochmals
alc	automatic level control	automatische Verstärkungsregelung
am	ante meridiem	Vormittag
ani	any	irgendein, jemand
ans	answer	Antwort
ant	antenna	Antenne
any	any	irgendein, jemand
atu	antenna tuner	Antennentuner
atv	amateur television	Amateurfunkfernsehen
avc	automatic volume control	automatische Lautstärkeregelung
award	award	Amateurfunkdiplom, Auszeichnung
awdh		auf Wiederhören
awds		auf Wiedersehen
bci	broadcast interference	Rundfunkstörungen
bcl	broadcast listener	Rundfunkhörer
bcnu	be seeing you	Hoffe Sie wieder zu treffen.
bd	bad	schlecht
bfo	beat frequency oscillator	Telegraphieüberlagerer
bk		Bitte kommen
bpm		Buchstaben pro Minute
btr	better	besser
btu	back to you	Tasteneingabe an Sie zurück
bug	semi-automatic key	halbautomatische Taste
b4	before	vorher
c	correct, yes, affirmation	korrekt, Zusage, Ja (vom. span. "si")
call	call-sign	Rufzeichen
c11		Ortsverband C11 München Ost
cfm	confirm	bestätige
cheerio	cheerio	Servus! (Grußwort)
cl	close of station	Ich schließe die Station
clد	called	gerufen
clg	calling	rufend, ich rufe ...
cloudy	cloudy	wolkig
co	crystal oscillator	Quarzoszillator
col	collate	zusammenstellen, zusammenstellen
conds	conditions	Ausbreitungsbedingungen
condx	conditions for dx	Ausbreitungsbedingungen für DX
congrats	congratulations	Herzlichen Glückwunsch, Glückwünsche
cpi	copy	aufnehmen
cq	seek you	Allgemeiner Anruf
crd	card, verification card	QSL-Karte, Stationskarte

cs	call sign	Rufzeichen
cuagn	call you again	wir treffen uns wieder auf der Frequenz
cud	could	könnte
cul	see you later	bis später, bis bald
cw	code work, continuous waves	Tastfunk, Morsetelegraphie
cwfe	CW forever	Gruß unter Telegraphisten
darc		Deutscher Amateur Radio Club
dB	decibel	Dezibel
dc	direct current	Gleichstrom
de	from	von ... (vor dem eigenen Rufzeichen)
dk		Dank(e)
dks		Danke
DL		Deutschland
dmr	digital mobile radio	Neben CW eine weitere beliebte Art des Digitalfunks
dok		Distrikts-/Ortsverbandskennzeichnung des DARC
dr	dear	liebe, lieber
dwn	down	Geh auf eine niedrigere Frequenz.
dx	distance extended	Fernverbindung über den Horizont hinaus
el	element(s)	Antennenelemente
elbug	electronic bug	elektronische Morsetaste
ere	here	hier
es	and	und
esb		Einseitenband
excus	excuse	Entschuldigung
fb	fine business	ausgezeichnet, prima, sehr gut
fer	for	für
fm	frequency modulation	Frequenzmodulation
fone	telephonic exchange	Sprechfunk
fr	for	für
frd	friend	Freund
freq	frequency	Frequenz
frm	from	von
fsk	frequency shift keying	Frequenzumtastung, Frequenzverschiebungstastung
ga	go ahead	beginnen Sie
ga	good afternoon	Guten Nachmittag
ga		Guten Abend
gb	good bye	leben Sie wohl, auf Wiedersehen
gd	good day	Guten Tag
ge	good evening	Guten Abend
gl	good luck	viel Glück
gld	glad	sehr erfreut
gm	good morning	Guten Morgen
gn	good night	Gute Nacht
gnd	ground	Masseverbindung
gp	ground plane	Ground-Plane Antenne
gt		Guten Tag
gud	good	gut
guhör	going unable to hear or receive	kein Empfang mehr möglich
ham	"Hyman-Almy-Murry Station"	Bezeichnung für einen Funkamateure seit 1908
hf	high frequency	Hochfrequenz bzw. Kurzwellen 0,3-30MHz

hi	laughing	lachen zum Ausdruck bringen
hpe	hope	hoffen
hrd	heard	gehört
hr	here	hier
ht	handy talkie	Handfunksprechgerät
hw?	how?	wie (werde ich empfangen)?
hwsat?	how about that?	wie finden Sie das?
hzi		herzlich(st)
i	I	ich
ik		ich
IARU	International Amateur Radio Union	
if	intermediate frequency	Zwischenfrequenz
info	information	Information
inpt	input	Eingangsleistung
irc	international reply coupon	Internationaler Antwortschein
knw	know	ich weis
key	key	Handtaste bzw. jeder Typ von Morsetaste
lbr		lieber
lf	low frequency	niedrigere Frequenz
lid	lousy incompetent dummy	schlechter Operator
lis	licensed, the license	lizenziert, die AFu Lizenz (Amateurfunkzeugnis)
log	log book	Logbuch
lsb	lower sideband	unteres Seitenband
lsn	listen	hören Sie bitte
ltr	letter	Brief
luf	lowest usable frequency	niedrigste nutzbare Frequenz
lw	long wire antenna	Langdrahtantenne
min	minute	Minute
mins	minutes	Minuten
mm	maritime mobile	bewegliche Seestation vorrangig auf einem Schiff
mni	many	viel, viele
mom	moment	Einen Moment bitte.
msg	message	Nachricht bzw. Funktelegramm
mtr	meter	Messgerät
muf	maximum usable Frequency	höchste für eine Kurzwellen-Funkverbindung zwischen zwei Orten verwendbare Frequenz
my	my	mein
n	no (negation)	nein, Verneinung
net	net(work)	Funknetz
nf		Niederfrequenz 3-30 kHz
nil	no information listed	absolut nichts, leer, keine Information
nix		nichts
no	no	nein
no	numero	Ordinale Nummerierung für Sequenzen
nr	near	nahe
nr	number	Nummer
nw	now	jetzt, nun
ob	old boy	alter Junge (vertrauliche Anrede)
oc	old chap	alter Knabe (vertrauliche Anrede)
ok	"Oscar Keller" (Shipping NY)	in Ordnung
om	old man	Funker(in)

op	operator	Funker(in), Operator einer Funkstation
osc	oscillator	Oszillator
ot	old-timer	langjährige(r) Funk(er)in
outpt	output	Ausgangsleistung
ov		Ortsverband
ow	old woman	Funkerin bzw. Ehefrau eines Funkers
pa	power amplifier	Endstufe
pep	peak envelope power	Hüllkurvenleistung
pm	post meridiem	Nachmittag
pse	please	Bitte
psed	pleased	erfreut
pwr	power	Leistung
rain	rain	Regen
rr	roger roger	Richtig verstanden - alles richtig angekommen
rc	rag chewing	lange Unterhaltung in CW
rcvd	received	empfangen
rcvr	receiver	Empfangsgerät
rdy	ready	bereit
ref	reference	Referenz, Bezug, bezüglich
rf	radio frequency	Radiowellen, Hochfrequenz
rfi	radio frequency interference	Oberbegriff für Funkstörungen
rig	rig	Stationsausrüstung, Funkgerät(e) incl. Antennen
rppt	report	Bericht
rpt	repeat	wiederholen
rq	request	Frage
rst	readability strength tone	Lesbarkeit Signalstärke Tonqualität
rtty	radio teletype	Funkferschreiben
rx	receiver	Empfänger
sae	self addressed envelope	adressierter Rückumschlag
sase	self addressed and stamped envelope	freigemachter adressierter Rückumschlag
shf	super high frequency	3-30 GHz
sigs	signals	Signale
sked	schedule	Verabredung mit Zeit- und Frequenzangabe
sn	soon	bald
sri	sorry	leider
ssb	single sideband	Einseitenband(Modulation) meist Sprechfunk (langsame) Standbildübertragung mittels Tonmodulation
sstv	slow scan television	Amateurfunkstation
stn	station	einige, etwas, ein wenig
sum	some	sonnig
sunny	sunny	sicher(lich)
sure	sure	Kurzwellenhörer
swl	short wave listener	Stehwellenverhältnis
swr	standing wave ratio	Röhren
tbs	tubes	Temperatur (Angabe meist in Grad Celsius)
temp	temperature	Versuch
test	test	Aufruf oder Kennzeichnung in einem Wettbewerb
test	contest	Funkverkehr
tfc	traffic	durch, geleitet über ...
thru	through	Danke
tk	thanks	

tmw	tomorrow	morgen
tnx	thanks	Danke
trub	trouble	Schwierigkeiten, Störungen, Ärger
tu	thank you	Danke Dir
tu	to you	Taste an Dich zurück
tvi	television interference	Fernsehstörungen
tx	transmitter	Sender
trcvr	transceiver	Sendeleistungsgerät
trx	transceiver	Sendeleistungsgerät
txt	text	Text, Nachricht
u	you	Du
ufb	ultra fine business	ganz ausgezeichnet ohne Einschränkung
uhf	ultra high frequency	0,3-3 GHz
unlis	unlicensed	unlizenzierte Sendestation, "Pirat"
up	up	Wähle eine höhere Frequenz, aufwärts
ur	your	Dein
urs	yours	Deine Familie, die Ihrigen
usb	upper sideband	oberes Seitenband
utc	universal time coordinated	koordinierte Weltzeit (Zulu-Time)
v	v's	Gib eine Reihe v's.
ve		verstanden
vfo	variable frequency oscillator	Einstellbarer Frequenzoszillator
vhf	very high frequency	30-300 MHz
vi		
vin		vielen
vy	very	sehr
w	Watt	Watt
wid	with	mit
wkd	worked	gearbeitet mit ..., gefunkt mit ...
wkg	working	(gerade) in Verbindung mit
wl		
wpm	words per minute	Wörter pro Minute (bezogen auf "PARIS")
watts	Watts	Watts
wud	would	würde
wx	weather	Wetter
xcus	excuse	Entschuldigung, entschuldige bitte
xcvr		
xmas	Christmas	Weihnachten
yday	yesterday	gestern
xmit	transmit	sende
xmtr	transmitter	Sender
xtal	crystal, quartz crystal	Quarz
xxx	urgency signal	Dringlichkeitszeichen -- Im Amateurfunk nicht zugelassen.
xyl	ex young lady	Ehefrau
yl	young lady	Frau
yr	year	Jahr
ys	years	Jahre
z	Z-time, Zulu-time, zero-time	koordinierte Weltzeit (Zulu-Time)
zulu	Z-time, Zulu-time, zero-time	koordinierte Weltzeit (Zulu-Time)
znite	tonight	heute Nacht

55	Viel Erfolg, viele Punkte im Contest
73	Herzliche Grüße
88	Liebe und Küsse
99	Verschwinde
59	Einwandfrei lesbar, Signalstärke sehr gut
599	Einwandfrei lesbar, Signalstärke sehr gut und Qualität des CW-Tons einwandfrei
5nn	5nn entspricht 599: Einwandfrei lesbar, Signalstärke sehr gut und Qualität des CW-Tons
QOA	Ich kann mittels Telegraphiefunk verkehren (500 kHz) -- Seefunkgruppe im Amateurfunk nicht zugelassen.
QOB	Ich kann mittels Sprechfunk verkehren (2182 kHz) -- Seefunkgruppe im Amateurfunk nicht zugelassen.
QRA	Der Name meiner Funkstelle ist ...
QRA?	Wie lautet der Name Ihrer Funkstelle? Wie ist das Rufzeichen Ihrer Station?
QRB	Die Entfernung zwischen uns beträgt ...
QRB?	In welcher Entfernung von mir befinden Sie sich?
QRD	Ich bin unterwegs nach... von...
QRD?	Wohin sind Sie unterwegs und von woher?
QRG	Meine Frequenz ist ...
QRG?	Was ist Ihre Frequenz?
QRH	Ihre Frequenz schwankt.
QRH?	Schwankt meine Frequenz?
QRI	Der Ton Ihrer Aussendung ist ... (1 bis 9 wie in RST).
QRI?	Wie ist der Ton meiner Aussendung?
QRK	Die Verständlichkeit der Zeichen ist (1 bis 5 wie in RST).
QRK?	Wie ist die Verständlichkeit meiner Zeichen bzw. der Zeichen von ...
QRL	Mein QRL ist "an der Universität der Bundeswehr", z.B. "QRL UniBw"
QRL?	Sind Sie beschäftigt?
QRM	Ich werde (1=nicht, 2=kaum, 3=mäßig, 4=stark, 5=sehr stark) gestört.
QRM?	Werden Sie durch andere Signale gestört?
QRN	Ich werde (1=nicht, 2=kaum, 3=mäßig, 4=stark, 5=sehr stark) beeinträchtigt.
QRN?	Werden Sie durch atmosphärische Störungen beeinträchtigt?
QRO	Erhöhen Sie die Sendeleistung!
QRO?	Soll ich die Sendeleistung erhöhen?
QRP	Verringern Sie die Sendeleistung!
QRP?	Soll ich die Sendeleistung verringern?
QRQ	Geben Sie schneller!
QRQ?	Soll ich schneller geben?
QRS	Geben Sie langsamer!
QRS?	Soll ich langsamer geben?
QRT	Stellen Sie die Übermittlung ein! bzw. Ich stelle die Übermittlung ein.
QRT?	Soll ich die Übermittlung einstellen?
QRU	Ich habe nichts für Sie vorliegen.
QRU?	Haben Sie etwas für mich?
QRV	Ich bin bereit Sie aufzunehmen.
QRV?	Sind Sie bereit (mich aufzunehmen)?
QRX	Ich werde Sie um ... Zulu (00:00) auf ... (kHz bzw. MHz) wieder rufen.
QRX?	Wann werden Sie mich wieder rufen?
QRZ	Sie werden von ... gerufen.
QRZ?	Von wem werde ich gerufen? (QRZ muss immer in Verbindung mit dem eigenen Rufzeichen)
QSA	Ihre Zeichen sind (1=kaum hörbar, 2=schwach, 3=ziemlich gut, 4=gut, 5=sehr gut) hörbar.
QSA?	Wie ist die Stärke meiner Zeichen?

QSB?	Schwankt die Stärke meiner Zeichen?
QSD	Ihre Zeichen sind verstümmelt.
QSD?	Sind meine Zeichen verstümmelt?
QSK	Wird im Text verwendet wenn man z.B. ein "QSK QSO" geführt hat.
QSK?	Können Sie mich zwischen Ihren Zeichen hören? Wenn ja, darf ich Sie unterbrechen?
QSL	Es wird als Synonym für "Empfangsbestätigung(skarte)" im Text verwendet.
QSL?	Können Sie den Empfang bestätigen?
QSO	Es wird als Synonym für "Verbindung" im Text verwendet.
QSO?	Können Sie mit ... verkehren?
QSP	Vermitteln Sie bitte mit ...
QSP?	Wollen Sie an ... vermitteln?
QSV	Es folgt gleich die (gewünschte) Reihe von "v"s.
QSV?	Soll ich eine Reihe "v"s senden?
QTC	Es folgen ... (Anzahl) von Telegrammen, z.B. beim Contest.
QTC?	Wie viele Telegramme haben Sie?
QSY	Gehen Sie zum Senden auf die Frequenz ...
QSY?	Soll ich zum Senden auf eine andere Frequenz übergehen?
QTH	Meine Position ist ...
QTH?	Wie ist Ihre Position?
QTR	Es ist genau ... (Uhrzeit in Zulu-Time, Format 00:00)
QTR?	Welches ist die genaue Uhrzeit?

ABBILDUNGSVERZEICHNIS

Bild 1: Up- und Down-Chirp Puls im Zeit- und Frequenzbereich [Quelle: FA 11/21 Seite 868].	11
Bild 2: Up-Chirp für ein Symbol mit dem Dezimalwert 94.....	12
Bild 3: Exemplarisches LoRa™ Spektrogramm für eine Übertragung mit explizitem Paketformat.	12
Bild 4: Blockschaltbild des LoRa™ HF Pfads des Heltec Boards	27
Bild 5: Balkenanzeige im Dual-Watch Mode	55
Bild 6: Anzeigemodule des Dual-Watch im Display	55
Bild 7: Einstellungen zum Import des .csv Files in EXCEL.....	56
Bild 8: Bildsymbole des APRS-Basissets	61
Bild 9: Anzeige des Kartensymbols mit Metadaten	62
Bild 10: www.aprs.fi -Kartenausschnitt	65
Bild 11: Heltec ESP32 LoRa V.2 Modul Vorder- und Rückseite (Quelle: [BildHeltecs])	74

Bild 12: Das Gehäuse für den LoRa™-C11-TRX mit Halterung für den Heltec ESP32 LoRa V.4.	76
Bild 13: Oberschale (blau) neben Bodendeckel (rot).....	77
Bild 14: Der Bodendeckel rastet zum ersten Mal ein.	77
Bild 15: Bodendeckel und Oberschale sind nun zu einem Gehäuse verbunden.....	78
Bild 16: Trennen der Gehäuseteile – dabei nicht auf die Laschen drücken.	78
Bild 17: Alle Komponenten passen in die Bohrungen.....	79
Bild 18: Einkleben des Folienstreifens.....	79
Bild 19: Korrekturschneiden des Folienstreifens.....	80
Bild 20: Eingebaute SMA-Buchse mit uFL-Pigtail.....	80
Bild 21: Grundbeschaltung des FTD	81
Bild 22: Batterie und Schalter	81
Bild 23: Die fertige Grundverdrahtung des LoRa™-FTD	82
Bild 24: Eine Version des CW-Trainers mit angeschlossenem Kopfhörer.....	82
Bild 25: Schaltung Kopfhöreranschluss	83

Bild 26:LoRa™-FTD mit CW-Schaltung erweitert	83
Bild 27: Anderer Platinenentwurf – Platine vor dem Festschrauben im Gehäuse	84
Bild 28: Schaltung zum Anschluss des Drehenkoders für dasLoRa™-FTD.	85
Bild 29: C3 direkt am Encoder aufgelötet.	86
Bild 30: Der geöffneteLoRa™-FTD mit Wire-Wrap „Luftverdrahtung“ (es geht auch besser).....	86
Bild 31: Platinenlayout für die Zusatzschaltung für den Drehenkoder.	87
Bild 32: Bestückungsplan für die Drehenkoderplatine.....	87
Bild 33: Platine für Pull-Ups und Stützkondensatoren.....	87
Bild 34: Der in die Gehäuseoberschale eingerastete Drehgeber mit aufgesetzter Interfaceplatine. ..	88
Bild 35: Das komplett aufgebaute LoRa™ Field Test Device.....	88

REFERENZEN

- [1] Erich Huettmann, Verfahren zur Entfernungsmessung, Deutsches Patent E 768068, 1940.
- [2] Eirini Karapistoli, Fotini-Niovi Pavlidou, Ioannis Gragopoulos und Ioannis Tsetsinas, An Overview of the IEEE 802.15.4a Standard, IEEE STANDARDS IN COMMUNICATIONS AND NETWORKING, IEEE Communications Magazine, Januar 2010, Seiten 47 – 53.
- [3] Semtech Corporation, LoRa™ Modulation Basics, Application Note AN1200.22, Rev.2 – Mai 2015.
- [4] P. Robyns, P. Quax, W. Lamotte und W. Thenaers, A Multi-Channel Software Decoder for the LoRa Modulation Scheme, In Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security, 2018, Seiten 41-51.
- [5] Semtech Corporation, LoRa™ SX1272/3/6/7/8: LoRa Modem Design Guide, Application Note: AN1200.13, Rev.1 – Juli 2013.
- [6] https://heltec-automation-docs.readthedocs.io/en/latest/esp32/quick_start.html#via-arduino-board-manager, Heltec Automation, Installationsanleitung der Heltec Bibliotheken für die Arduino Umgebung, gesehen 01.01.2021 um 09:04.
- [7] SILICON LABS, CP210x USB to UART Bridge VCP Drivers, <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>, Beschreibung des VCP Treibers und Link zur Installation, gesehen 01.01.2021 um 19:27.
- [8] Ian Wade (Editor), The APRS Working Group, Automatic Position Reporting System, APRS Protocol Reference, Protocol Version 1.0, August 2000.
- [9] Manfred Mauler, Vortrag: APRS® - Automatic Packet Reporting System – Das Telekommunikationssystem der Funkamateure, APRS Workshop, 30.06.2019.
- [SAND2020]: LoRa von Sandeep Mistry, Version 0.8.0, Die Bibliothek unterliegt der MIT Licence: <https://github.com/sandeepmistry/arduino-LoRa/blob/master/LICENSE>, Stand 21. Juni 2022.
- [OLIKRAUS]: Universal 8bit Graphics Library U8x8lib.h und u8g2 von Oli Kraus, <https://github.com/olikraus/u8g2/blob/master/cppsrc/U8x8lib.h>, Die Bibliothek unterliegt der Lizenz: <https://github.com/olikraus/u8g2/blob/master/LICENSE>, Stand 2. Januar 2021.
- [2022Thingiverse]: Makoto Schoppert, Heltec WiFi LoRa 32 (V2) Development Board Case,

19.01.2021, <https://www.thingiverse.com/thing:4729713> , Link zur Nutzerlizenz:
<https://creativecommons.org/licenses/by-nc-sa/4.0/> , gesehen 19.05.2022 um 14:52.