

Inhaltsverzeichnis

| | |
|--|-----|
| 1. QRA64 | 170 |
| 2. Benutzer:OE1VMC | 16 |
| 3. FSK441 | 30 |
| 4. FST4 | 44 |
| 5. FT4 | 58 |
| 6. FT8 | 72 |
| 7. Grundlagen Digitale Betriebsarten | 86 |
| 8. JT4 | 100 |
| 9. JT65 | 114 |
| 10. JT6M | 128 |
| 11. JT9 | 142 |
| 12. MSK144 | 156 |
| 13. WSPR | 184 |

QRA64

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr
(Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:
33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

+

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

+

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |0

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |0

| | | | |
|---|------------------------------|---|------------------------------|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| | - | | - |
| – | style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|-------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12: 33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

+

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

+

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "**Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.**

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

[<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf> "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur

[<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

Zeile 138:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

[<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf> "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

– +

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

– +

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "**Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.**

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

```
{| class="wikitable"
```

```
{| class="wikitable"
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

| | | |
|----------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|-------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

| | |
|--|--|
| <p>Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den</p> | |
| <p>VHF und UHF Bändern. QSOs in der Betriebsart QRA64A werden aktuell (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart könnte auch populär w erden auf den Lang-, Mittel- und weiteren Kurzwellenbändern.</p> | |
| <p>EME Aktivitäten im 2m Band fahren inzwischen öfters QRA64 vor allem am Wochenende, siehe [http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf ARRL News Update am 10. Jan. 2017].</p> | |
| <p>Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.</p> | <p>Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> |
| <p>Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> | |
| <p>Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).</p> | |
| | |

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

– +

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjt-doc/wsjt-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

– +

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |0

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |0

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|---|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWW], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWW-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWW], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWW-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

[OE1VMC](#) ([Diskussion](#) | [Beiträge](#))

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

[← Zum vorherigen Versionsunterschied](#)

Aktuelle Version vom 23. März 2024, 12: 33 Uhr (Quelltext anzeigen)

[OE1VMC](#) ([Diskussion](#) | [Beiträge](#))

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: [Visuelle Bearbeitung](#)

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

+

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

+

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

```
{| class="wikitable"
```

```
{| class="wikitable"
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

- [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur

- [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

Zeile 138:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

+ [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\: QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **w****erden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

+

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

+

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

Nico Palermo, [<http://www.qrz.com/db/IV3NWW>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

- [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWW-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur

- [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

Zeile 138:

Nico Palermo, [<http://www.qrz.com/db/IV3NWW>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

- + [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWW-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\; QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) $GF(64)$ mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|-------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr
(Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:
33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

QRA Codes bilden eine spezielle Unterklasse

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|----------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|---|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr
(Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:
33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

- == Digitale Betriebsarten im Detail:
QRA64 ==

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+ ==Digitale Betriebsarten im Detail:
QRA64==

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

| | |
|---|---|
| <ul style="list-style-type: none"> – ORA64 hat viele Gemeinsamkeiten mit <code>[JT65]</code>, <code>[JT9]</code> und <code>[JT4]</code>. | <p>Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]. Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.</p> |
| <ul style="list-style-type: none"> – Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. | |
| <ul style="list-style-type: none"> – Details zur Quellencodierung wurden veröffentlicht im Artikel "http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]", der in der Zeitschrift http://www.arrl.org/qex QEX während 2005 veröffentlicht wurde. | |
| <ul style="list-style-type: none"> – Das Codierungsverfahren wurde entworfen von Nico Palermo, http://www.qrz.com/db/IV3NWV IV3NWV], und implementiert in WSJT-X Version 1.7.0 durch http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr Joe Taylor], http://www.qrz.com/db/K1JT K1JT]. | <p>QRA64 hat viele Gemeinsamkeiten mit <code>[JT65]</code>, <code>[JT9]</code> und <code>[JT4]</code>. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]", der in der Zeitschrift http://www.arrl.org/qex QEX während 2005 veröffentlicht wurde.</p> |
| <ul style="list-style-type: none"> – Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. | |
| <ul style="list-style-type: none"> – QRA Codes bilden eine spezielle Unterklasse | |
| <ul style="list-style-type: none"> – der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet | |

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

```
{| class="wikitable"
```

```
{| class="wikitable"
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```


| | | |
|----------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

[OE1VMC](#) ([Diskussion](#) | [Beiträge](#))

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

[← Zum vorherigen Versionsunterschied](#)

Aktuelle Version vom 23. März 2024, 12: 33 Uhr (Quelltext anzeigen)

[OE1VMC](#) ([Diskussion](#) | [Beiträge](#))

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: [Visuelle Bearbeitung](#)

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

QRA Codes bilden eine spezielle Unterklasse

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

```
{| class="wikitable"
```

```
{| class="wikitable"
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

- == Digitale Betriebsarten im Detail:
QRA64 ==

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+ ==Digitale Betriebsarten im Detail:
QRA64==

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

– +

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjt-doc/wsjt-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

– +

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\: QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

| | |
|--|--|
| <p>– Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den</p> | |
| <p>– VHF und UHF Bändern. QSOs in der Betriebsart QRA64A werden aktuell (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart könnte auch populär w erden auf den Lang-, Mittel- und weiteren Kurzwellenbändern.</p> | |
| <p>– EME Aktivitäten im 2m Band fahren inzwischen öfters QRA64 vor allem am Wochenende, siehe [http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf ARRL News Update am 10. Jan. 2017].</p> | |
| <p>– Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.</p> | <p>+ Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> |
| <p>– Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> | |
| <p>– Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).</p> | |

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

+

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

+

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|---------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr
(Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:
33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

| | |
|--|--|
| <p>– Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den</p> | |
| <p>– VHF und UHF Bändern. QSOs in der Betriebsart QRA64A werden aktuell (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart könnte auch populär w erden auf den Lang-, Mittel- und weiteren Kurzwellenbändern.</p> | |
| <p>– EME Aktivitäten im 2m Band fahren inzwischen öfters QRA64 vor allem am Wochenende, siehe [http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf ARRL News Update am 10. Jan. 2017].</p> | |
| <p>– Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.</p> | <p>+ Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> |
| <p>– Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [http://physics.princeton.edu/pulsar/k1jt/wsjsx.html WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([http://www.qrz.com/db/K1JT K1JT]).</p> | |
| <p>– Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).</p> | |
| | |

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

–

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjt-doc/wsjt-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

+

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

–

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

–

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

–

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

+

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

–

QRA Codes bilden eine spezielle Unterklasse

–

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

–

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

```
{| class="wikitable"
```

```
{| class="wikitable"
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |X
```

```
| style="text-align:right;" |0
```

```
| style="text-align:right;" |0
```


| | | | |
|---|------------------------------|---|------------------------------|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| | - | | - |
| – | style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|-------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12: 33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

- == Digitale Betriebsarten im Detail:
QRA64 ==

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+ ==Digitale Betriebsarten im Detail:
QRA64==

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

| | |
|--|---|
| <ul style="list-style-type: none"> – ORA64 hat viele Gemeinsamkeiten mit <code>[JT65]</code>, <code>[JT9]</code> und <code>[JT4]</code>. | <p>Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]. Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.</p> |
| <ul style="list-style-type: none"> – Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. | |
| <ul style="list-style-type: none"> – Details zur Quellencodierung wurden veröffentlicht im Artikel "http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]", der in der Zeitschrift http://www.arrl.org/qex QEX während 2005 veröffentlicht wurde. | |
| <ul style="list-style-type: none"> – Das Codierungsverfahren wurde entworfen von Nico Palermo, http://www.qrz.com/db/IV3NWV IV3NWV], und implementiert in WSJT-X Version 1.7.0 durch http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor, Jr. Joe Taylor], http://www.qrz.com/db/K1JT K1JT]. | <p>QRA64 hat viele Gemeinsamkeiten mit <code>[JT65]</code>, <code>[JT9]</code> und <code>[JT4]</code>. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]", der in der Zeitschrift http://www.arrl.org/qex QEX während 2005 veröffentlicht wurde.</p> |
| <ul style="list-style-type: none"> – Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. | |
| <ul style="list-style-type: none"> – QRA Codes bilden eine spezielle Unterklasse | |
| <ul style="list-style-type: none"> – der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet | |

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|----------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

- [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur

- [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

Zeile 138:

Nico Palermo, [<http://www.qrz.com/db/IV3NWV>], stellte den neuen Übertragungsmodus QRA64 in

seinem Beitrag

+ [<http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf>] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [<http://www.eme2016.org> 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wshtx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/WSJT/>], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wshtx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|----------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).

QRA64: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 23. Februar 2018, 22:49 Uhr
(Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Link zum deutschsprachigen
Benutzerhandbuch eingefügt)

← Zum vorherigen Versionsunterschied

Aktuelle Version vom 23. März 2024, 12:
33 Uhr (Quelltext anzeigen)

OE1VMC (Diskussion | Beiträge)

(Die aktuelle Programmversion ist WSJT-X
Version 2.6.1 (Stand: 23.03.2024), siehe .
WSJT-X 2.6.1 Handbuch. Ein "candidate
release" 2.7.0-rc4 ist ebenfalls verfügbar.)

Markierung: Visuelle Bearbeitung

(18 dazwischenliegende Versionen desselben Benutzers werden nicht angezeigt)

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

-

== Digitale Betriebsarten im Detail:
QRA64 ==

-

QRA64 ist eine **neue** digitale Betriebsart
(2016 !!), die sehr geeignet ist für
niedrige Sendeleistung ("QRP-Betrieb")
und für Stationen mit Antennendefiziten.

Zeile 2:

[[Kategorie:Erde-Mond-Erde]]

+

==Digitale Betriebsarten im Detail:
QRA64==

+

QRA64 ist eine digitale Betriebsart, **die ab
WSJT-X Version 1.7.0** (2016) **verfügbar
ist und** die sehr geeignet ist für niedrige
Sendeleistung ("QRP-Betrieb") und für
Stationen mit Antennendefiziten. Diese
Betriebsart wurde ursprünglich entwickelt
für Erde-Mond-Erde (EME) Verbindungen
und für Verbindungen mit sehr geringer
Sendeleistung auf den VHF und UHF
Bändern. QSOs in der Betriebsart QRA64A
wurden ab Januar 2017 nachts auf 160m
ausprobiert. Diese Betriebsart **ist nicht** po
pulär **geworden** auf den Lang-, Mittel-
und weiteren Kurzwellenbändern. EME
Aktivitäten im 2m Band **arbeiten** inzwisch
en öfters QRA64 vor allem am
Wochenende, siehe [http://www.arrl.org
/news/view/new-digital-modes-gain-
traction-for-moonbounce-but-occasionally-
show-up-on-hf ARRL News Update am 10.
Jan. 2017].

- Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A **werden aktuell** (Januar 2017) nachts auf 160m ausprobiert. Diese Betriebsart **könnte auch** populär **werden** auf den Lang-, Mittel- und weiteren Kurzwellenbändern.
- EME Aktivitäten im 2m Band **fahren** inzwischen öfters QRA64 vor allem am Wochenende, siehe [<http://www.arrl.org/news/view/new-digital-modes-gain-traction-for-moonbounce-but-occasionally-show-up-on-hf> ARRL News Update am 10. Jan. 2017].

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC.

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html> WSJT-X] Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor] ([<http://www.qrz.com/db/K1JT> K1JT]).

- **Die aktuelle WSJT-X Programmversion ist 1.8.0 (Stand 23.02.2018, siehe [https://physics.princeton.edu/pulsar/k1jt/wsjsx-doc/wsjsx-main-1.8.0_de.pdf deutschsprachiges Benutzerhandbuch]).**

ORA64 hat viele Gemeinsamkeiten mit **[JT65]**, **[JT9]** und **[JT4]**.

Die aktuelle Programmversion ist **WSJT-X Version 2.6.1** (Stand: **23.03.2024**), siehe **[https://wsjt.sourceforge.io/wsjitx-doc/wsjitx-main-2.6.1.html WSJT-X 2.6.1 Handbuch]**. Ein "candidate release" **2.7.0-rc4** ist ebenfalls verfügbar.

Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung.

Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, **[http://www.qrz.com/db/IV3NWV IV3NWV]**, und implementiert in **WSJT-X Version 1.7.0** durch **[http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr. Joe Taylor]**, **[http://www.qrz.com/db/K1JT K1JT]**.

QRA64 hat viele Gemeinsamkeiten mit **[JT65], **[JT9]** und **[JT4]**. Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "**[http://physics.princeton.edu/pulsar/K1JT/JT65.pdf The JT65 Communications Protocol]**", der in der Zeitschrift **[http://www.arrl.org/qex QEX]** während 2005 veröffentlicht wurde.**

Das **QRA64 Kommunikationsprotokoll** basiert auf einem "Q-ary **[https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code]** für **Q=64**.

QRA Codes bilden eine spezielle Unterklasse

der sogenannten **Low-Density Parity Check (LDPC) Codes** mit Symbolen auf einem **Q-wertigen Alphabet**

| | |
|--|--|
| <p>– (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf</p> | |
| <p>– orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> | |
| <p>– QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird.</p> | <p>+ Das Codierungsverfahren wurde entworfen von Nico Palermo, [http://www.qrz.com/db/IV3NWW IV3NWW], und implementiert in WSJT-X Version 1.7.0 durch [http://en.wikipedia.org/wiki/Joseph_Hooton_Taylor,_Jr._Joe_Taylor], [http://www.qrz.com/db/K1JT K1JT]. Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [https://en.m.wikipedia.org/wiki/Repeat-accumulate_code Repeat-Accumulate Code] für Q=64. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet (Q=4, 8, 16, 32, 64,... bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).</p> |
| <p>– 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> | <p>+ </p> |
| <p>– 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole).</p> | <p>+ QRA64 basiert auf einem irregulären QRA (63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [[JT65]] verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt),</p> |
| <p>– Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> | <p>+ 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA (64,13) Code über das [https://de.m.wikipedia.org/wiki/Endlicher_Körper_Galois-Feld] GF(64) mit einem CRC-6.</p> |

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus.

Zeile 33:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Zeile 21:

Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7 x 7 [https://en.wikipedia.org/wiki/Costas_array] Costas array] beruht.

Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

{| class="wikitable"

{| class="wikitable"

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |0

| style="text-align:right;" |X

| style="text-align:right;" |X

| style="text-align:right;" |0

| style="text-align:right;" |0

| | | |
|----------------------------------|---|------------------------------|
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| - | | - |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – style="text-align:right;" X | + | style="text-align:right;" X |
| - | | - |
| – style="text-align:right;" X | + | style="text-align:right;" X |

| | | | |
|---|--|---|---|
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | - | | - |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" X | + | style="text-align:right;" X |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| – | style="text-align:right;" 0 | + | style="text-align:right;" 0 |
| | } | | } |
| | | | |
| – | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. | + | Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. |
| – | Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65. | | |

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]].

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei [[JT9]]. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

(Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt)

Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code.

Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \cdot 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Zeile 110:

| |
|--|
| +Dial Frequency |
| - |
| - style="text-align:right;" 160m |
| - style="text-align:right;" 1,836 MHz |
| - |
| - style="text-align:right;" 80m |
| - style="text-align:right;" 3,574 MHz |
| - |
| - style="text-align:right;" 60m |
| - style="text-align:right;" 5,35x MHz |
| - |
| - style="text-align:right;" 40m |
| - style="text-align:right;" 7,074 MHz |
| - |
| - style="text-align:right;" 30m |
| - style="text-align:right;" 10,136 MHz |
| - |
| - style="text-align:right;" 20m |
| - style="text-align:right;" 14,074 MHz |
| - |
| - style="text-align:right;" 17m |
| - style="text-align:right;" 18,100 MHz |

Zeile 92:

| |
|---|
| +Dial Frequency |
| - |
| + style="text-align:right;" 160m |
| + style="text-align:right;" 1,836 MHz |
| - |
| + style="text-align:right;" 80m |
| + style="text-align:right;" 3,574 MHz |
| - |
| + style="text-align:right;" 60m |
| + style="text-align:right;" 5,35x MHz |
| - |
| + style="text-align:right;" 40m |
| + style="text-align:right;" 7,074 MHz |
| - |
| + style="text-align:right;" 30m |
| + style="text-align:right;" 10,136 MHz |
| - |
| + style="text-align:right;" 20m |
| + style="text-align:right;" 14,074 MHz |
| - |
| + style="text-align:right;" 17m |
| + style="text-align:right;" 18,100 MHz |

| | |
|--|--|
| - | - |
| - style="text-align:right;" 15m | + style="text-align:right;" 15m |
| - style="text-align:right;" 21,074 MHz | + style="text-align:right;" 21,074 MHz |
| - | - |
| - style="text-align:right;" 12m | + style="text-align:right;" 12m |
| - style="text-align:right;" 24,915 MHz | + style="text-align:right;" 24,915 MHz |
| - | - |
| - style="text-align:right;" 10m | + style="text-align:right;" 10m |
| - style="text-align:right;" 28,074 MHz | + style="text-align:right;" 28,074 MHz |
| - | - |
| - style="text-align:right;" 6m | + style="text-align:right;" 6m |
| - style="text-align:right;" 50,274 MHz | + style="text-align:right;" 50,274 MHz |
| - | - |
| - style="text-align:right;" 2m | + style="text-align:right;" 2m |
| - style="text-align:right;" 144,xxx MHz | + style="text-align:right;" 144,xxx MHz |
| } | } |
| | |

Zeile 156:

| |
|--|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| - [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur |
| - [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

Zeile 138:

| |
|---|
| Nico Palermo, [http://www.qrz.com/db/IV3NWV], stellte den neuen Übertragungsmodus QRA64 in |
| seinem Beitrag |
| + [http://www.eme2016.org/wp-content/uploads/2016/08/EME-2016-IV3NWV-Presentation.pdf] "Q-ary Repeat-Accumulate Codes for Weak Signals Communications"] zur [http://www.eme2016.org 17th International EME Conference], Venedig, Italien, 19.-21. August 2016 vor. |

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<http://physics.princeton.edu/pulsar/k1jt/wsjt.html>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<http://physics.princeton.edu/pulsar/k1jt/wsjsx.html>] WSJT-X].

Weitere Informationen: [[http://en.wikipedia.org/wiki/WSJT_\(Amateur_radio_software\)](http://en.wikipedia.org/wiki/WSJT_(Amateur_radio_software))] WSJT (Wikipedia)], [<https://wsjt.sourceforge.io/>] WSJT], [<http://ac4m.us/jt65.html>] AC4M Digital Radio Site] und [<https://wsjt.sourceforge.io/wsjsx.html>] WSJT-X].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]] und [[JT6M]].

Siehe auch: [[Grundlagen Digitale Betriebsarten]], [[JT4]], [[JT9]], [[JT65]], [[WSPR]], [[MSK144]], [[FSK441]], [[FT8]], [[FT4]], [[FST4]] und [[JT6M]].

Aktuelle Version vom 23. März 2024, 12:33 Uhr

Digitale Betriebsarten im Detail\ QRA64

QRA64 ist eine digitale Betriebsart, die ab WSJT-X Version 1.7.0 (2016) verfügbar ist und die sehr geeignet ist für niedrige Sendeleistung ("QRP-Betrieb") und für Stationen mit Antennendefiziten. Diese Betriebsart wurde ursprünglich entwickelt für Erde-Mond-Erde (EME) Verbindungen und für Verbindungen mit sehr geringer Sendeleistung auf den VHF und UHF Bändern. QSOs in der Betriebsart QRA64A wurden ab Januar 2017 nachts auf 160m ausprobiert. Diese Betriebsart ist nicht populär geworden auf den Lang-, Mittel- und weiteren Kurzwellenbändern. EME Aktivitäten im 2m Band arbeiten inzwischen öfters QRA64 vor allem am Wochenende, siehe [ARRL News Update am 10. Jan. 2017](#).

Implementiert wird diese digitale Betriebsart über die Soundkarte eines PC. Zuerst eingeführt wurde QRA64 mit Hilfe der Implementierung als Open Source Software [WSJT-X](#) Version 1.7.0 durch [Joe Taylor \(K1JT\)](#).

Die aktuelle Programmversion ist WSJT-X Version 2.6.1 (Stand: 23.03.2024), siehe [.WSJT-X 2.6.1 Handbuch](#). Ein "candidate release" 2.7.0-rc4 ist ebenfalls verfügbar.

QRA64 hat viele Gemeinsamkeiten mit [JT65](#), [JT9](#) und [JT4](#). Diese digitalen Modi verwenden fast identische Nachrichtenstruktur und Quellencodierung. Details zur Quellencodierung wurden veröffentlicht im Artikel "[The JT65 Communications Protocol](#)", der in der Zeitschrift [QEX](#) während 2005 veröffentlicht wurde.

Das Codierungsverfahren wurde entworfen von Nico Palermo, [IV3NWV](#), und implementiert in WSJT-X Version 1.7.0 durch [Joe Taylor, K1JT](#). Das QRA64 Kommunikationsprotokoll basiert auf einem "Q-ary [Repeat-Accumulate Code](#) für $Q=64$. QRA Codes bilden eine spezielle Unterklasse der sogenannten Low-Density Parity Check (LDPC) Codes mit Symbolen auf einem Q-wertigen Alphabet ($Q=4, 8, 16, 32, 64, \dots$ bzw. jede Zahl Q für die ein Galoisfeld existiert). Die Q-wertigen Symbole lassen sich einfach auf orthogonale Modulation abbilden (in diesem Fall: 64-FSK).

QRA64 basiert auf einem irregulären QRA(63,12) Code mit derselben Rate und demselben Symbolalphabet wie beim Reed-Solomon Code, der in [JT65](#) verwendet wird. 12 Informationssymbole (mit je 6 bit Informationsgehalt), 51 Prüfsymbole (Länge eines Codewortes = 63 Symbole). Genau genommen handelt es sich hierbei um einen punktierten QRA(64,13) Code über das [Galois-Feld](#) GF(64) mit einem CRC-6.

Decodiert wird nach dem MAP Kriterium (Maximum A-Posteriori Probability) mit Hilfe eines iterativen Message Passing (MP) Algorithmus. Die QRA64 Codierung schlägt den bekannten Reed Solomon (63,12) Code in JT65 um weitere 1,3 dB im Signal- zu Rauschleistungsverhältnis. Der aktuelle MAP Detektor in WSJT-X 1.7.0 dekodiert QRA64 bis zu einem Signal- zu Rauschleistungsverhältnis von etwa -27 oder -28 dB.

Das Signal besteht aus 64 Tönen. QRA64 verwendet ein neues Synchronisierungsverfahren, das auf einem 7×7 [Costas array](#) beruht. Es gibt 200 verschiedene Costas Arrays der Ordnung 7. Das für FT8 verwendete Costas Array ist die Permutation (2,5,6,0,4,1,3). Die folgende Graphik veranschaulicht die zeitliche Abfolge der 7 Synchronisierungstöne: In jeder Zeile und jeder Spalte steht genau ein "X".

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 | 0 |
| 0 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | X | 0 |
| 0 | 0 | 0 | X | 0 | 0 | 0 |

Daher sieht man im QRA64 Signal keinen auffälligen Synchronisierungston bei der niedrigsten belegten Frequenz. Diese Neuerung bringt weitere 1,9 dB Vorteil gegenüber JT65.

Digitale Daten werden strukturiert in Paketen mit 72 Informationsbits, wie bei JT9. (Eine typische Nachricht besteht aus zwei Rufzeichen, die jeweils mit 28 bit dargestellt werden, einem Grid-Locator, dargestellt durch 15 bit, und einem Signalisierungsbit, dass den Nachrichtentyp festlegt) Die Informationsbits werden kodiert mit einem "Q-ary Repeat Accumulate" Code. Diese Kodierung fügt zu den 72 Informationsbits noch 306 redundante Bits hinzu. Dies ergibt einen Kodeblock bestehend aus 378 bit und eine Code Rate $r = 72/378 = 1/5,25 = 0,19$. Diese 378 bit werden gruppiert zu 63 Kanalsymbolen zu je 6 bit (also: $6 \times 63 = 378$).

Jede Aussendung dauert exakt 46,8 Sekunden, die in 126 Symbole zu je etwa 0,372 Sekunden Symboldauer aufgeteilt werden. Ein Symbol wird abgetastet mit etwa 11025 Abtastwerten pro Sekunde, was 4096 digitalen Abtastwerten pro Symbol entspricht. Die Datenrate entspricht 2.69 baud. Wirksamer Durchsatz ist etwa 0.25 Buchstaben pro Sekunde (characters per second, cps). Die Varianten QRA64A, QRA64B und QRA64C (usw.) unterscheiden sich in den Frequenzabständen zwischen den 64 Tönen: 5,4 Hz, 10,8 Hz und 21,6 Hz. QRA64A belegt $(64/65) \times 177,6 \text{ Hz} = 174,87 \text{ Hz}$ Bandbreite, daher passen rund zehn QRA64 Signale nebeneinander in ein Spektrum mit 2kHz Bandbreite. Entsprechend belegt QRA64B die doppelte Bandbreite (= 349,74 Hz) und QRA64C die vierfache (= 699,47 Hz).

Die folgende Tabelle listet die üblichen Frequenzbereiche für QRA64 (Stand 2017). Die "Dial Frequency" gibt dabei die Frequenz des (unterdrückten) Trägers an. Dies ist also die angezeigte Frequenz am Funkgerät. Das Funkgerät moduliert das obere Seitenband (USB-Modulation).

Dial Frequency

| | |
|------|-------------|
| 160m | 1,836 MHz |
| 80m | 3,574 MHz |
| 60m | 5,35x MHz |
| 40m | 7,074 MHz |
| 30m | 10,136 MHz |
| 20m | 14,074 MHz |
| 17m | 18,100 MHz |
| 15m | 21,074 MHz |
| 12m | 24,915 MHz |
| 10m | 28,074 MHz |
| 6m | 50,274 MHz |
| 2m | 144,xxx MHz |

Die PC-Uhr muss auf 2 Sekunden genau sein In einer Aussendung werden maximal 13 ASCII Zeichen übertragen. Es werden nur folgende Informationen übertragen: Rufzeichen, Rapport in dB und LOC (4 Stellen).

Der QRA64 Signalrapport für das S/N ist beschränkt auf den Bereich zwischen -SNRLO und -SNRHI dB.

Nico Palermo, [IV3NWV](#), stellte den neuen Übertragungsmodus QRA64 in seinem Beitrag "[Q-ary Repeat-Accumulate Codes for Weak Signals Communications](#)" zur [17th International EME Conference](#), Venedig, Italien, 19.-21. August 2016 vor. QRA64 ist entwickelt worden für sehr schwache und langsam veränderliche Funkverbindungen, wie sie beispielsweise vorkommen im Zusammenhang mit Troposcatter und Erde-Mond-Erde Verbindungen. Die Empfangssignale sind erfolgreich dekodierbar bei Signal- zu Rauschverhältnissen bis ca. -27 dB oder -18 dB. Dies erlaubt das Arbeiten von Stationen, deren Signale mit dem menschlichen Ohr nicht mehr wahrgenommen werden können.

Weitere Informationen: [WSJT \(Wikipedia\)](#), [WSJT](#), [AC4M Digital Radio Site](#) und [WSJT-X](#).

Siehe auch: [Grundlagen Digitale Betriebsarten](#), [JT4](#), [JT9](#), [JT65](#), [WSPR](#), [MSK144](#), [FSK441](#), [FT8](#), [FT4](#), [FST4](#) und [JT6M](#).