

Digitales Tonrundfunksystem
Spezifikation des DAB-Befehlssatzes für Empfänger (DCSR)
Deutsche Fassung EN 50320:2000

DIN
EN 50320

ICS 33.160.20

Digital audio broadcasting system – Specification of the DAB
command set for receivers (DCSR);
German version EN 50320:2000

Système de radiodiffusion sonore numérique – Spécification du
jeu de commande DAB pour le récepteur (DCSR);
Version allemande EN 50320:2000

Die Europäische Norm EN 50320:2000 hat den Status einer Deutschen Norm.

Beginn der Gültigkeit

Die EN 50320 wurde am 2000-01-01 angenommen.

Nationales Vorwort

Für die vorliegende Norm ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimedia-systeme, -geräte und -komponenten“ der Deutschen Elektrotechnischen Kommission im DIN und VDE (DKE) zuständig.

Norm-Inhalt war veröffentlicht als E DIN EN 50320:1999-10.

Fortsetzung Seite 2
und 63 Seiten EN

Nationaler Anhang NA (informativ)

Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist nachstehend wiedergegeben. Zum Zeitpunkt der Veröffentlichung dieser Norm waren die angegebenen Ausgaben gültig.

IEC hat 1997 die Benummerung der IEC-Publikationen geändert. Zu den bisher verwendeten Normnummern wird jeweils 60000 addiert. So ist zum Beispiel aus IEC 68 nun IEC 60068 geworden.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm	Klassifikation im VDE-Vorschriftenwerk
–	ETS 300401:1997	DIN ETS 300401:1998-03	–
EN 50094:1992 EN 50094/A1:1995	–	DIN EN 50094:1993-03 DIN EN 50094/A1:1995-08	–
–	ISO 10486:1992	DIN ISO 10486:1993-10	–
EN 50248:1997	–	DIN EN 50248:1998-07	–

Nationaler Anhang NB (informativ)

Literaturhinweise

DIN EN 50094, *Zugriffskontrollsysteme für die MAC/Paket-Familie: Eurocrypt; Deutsche Fassung EN 50094:1992.*

DIN EN 50094/A1, *Zugriffskontrollsysteme für die MAC/Paket-Familie: EUROCRYPT; Deutsche Fassung EN 50094:1992/A1:1995.*

DIN EN 50248, *Eigenschaften von DAB-Empfängern; Deutsche Fassung EN 50248:1997.*

DIN ISO 10486, *Personenkraftwagen; Kennzeichnungsnummer für Autoradio (CRIN); Identisch mit ISO 10486:1992.*

DIN ETS 300401, *Rundfunk-Systeme – Digitaler Tonrundfunk (DAB) für mobile, tragbare und ortsfeste Empfänger; Englische Fassung ETS 300401:1997.*

ICS 33.160.20

Deutsche Fassung

Digitales Tonrundfunksystem
Spezifikation des DAB-Befehlssatzes für Empfänger (DCSR)

Digital audio broadcasting system – Specification of
the DAB command set for receivers (DCSR)

Système de radiodiffusion sonore numérique –
Spécification du jeu de commande DAB pour le
récepteur (DCSR)

Diese Europäische Norm wurde von CENELEC am 2000-01-01 angenommen.

Die CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Management-Zentrum oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Management-Zentrum mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.

CENELEC

EUROPÄISCHES KOMITEE FÜR ELEKTROTECHNISCHE NORMUNG
EUROPEAN COMMITTEE FOR ELECTROTECHNICAL STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION ELECTROTECHNIQUE

Management-Zentrum: rue de Stassart, 35 B-1050 Brüssel

Vorwort

Diese Europäische Norm wurde vom Technischen Komitee CENELEC TC 206 „Geräte der Unterhaltungs- und Informationselektronik und verwandte Systeme für den Allgemeingebrauch“ erstellt.

Der Text des Entwurfs wurde dem Einstufigen Annahmeverfahren unterworfen und von CENELEC am 2000-01-01 als EN 50320 angenommen.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2001-05-01
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 2003-01-01

Inhalt

	Seite		Seite
1 Allgemeine Einführung	3	4.11 <i>notify_fig</i> (Schnell-Informationsgruppe-Notifikation)	30
1.1 Anwendungsbereich	3	4.12 <i>select_figs</i> (Schnell-Informationsgruppen auswählen)	30
1.2 Struktur der DCSR	3	4.13 <i>get_channel</i> (Kanal angeben)	31
2 Referenztafel	4	4.14 <i>notify_channel</i> (Kanal-Meldung)	33
3 Antworten	5	4.15 <i>select_channel</i> (Kanal auswählen)	34
3.1 <i>accepted</i> (akzeptiert)	5	4.16 <i>get_selection_status</i> (Auswahlstatus angeben)	36
3.2 <i>rejected</i> (zurückgewiesen)	5	4.17 <i>notify_selection_status</i> (Auswahlstatus-Meldung)	37
3.3 <i>interim</i> (interim)	6	4.18 <i>search_for_ensemble</i> (Ensemblesuche)	38
3.4 <i>command_not_implemented</i> (Befehl nicht implementiert)	6	4.19 <i>notify_search_for_ensemble</i> (Ensemblesuche-Meldung)	42
3.5 <i>busy</i> (belegt)	6	4.20 <i>set_drc</i> (Dynamikbereichsregelung einschalten)	43
3.6 <i>syntax_error</i> (Syntaxfehler)	6	4.21 <i>get_audio_info</i> (Audio-Information angeben)	44
4 Befehle und Meldungen	7	4.22 <i>notify_audio_info</i> (Audio-Information-Meldung)	45
4.1 <i>get_receiver_capability</i> (Empfängermerkmale angeben)	7	4.23 <i>get_dab_status</i> (DAB-Status angeben)	46
4.2 <i>notify_receiver_capability</i> (Empfängermerkmale-Meldung)	7	4.24 <i>notify_dab_status</i> (DAB-Status-Meldung)	46
4.3 <i>tune</i> (Abstimmen)	17	4.25 <i>set_dab_status_auto_notification</i> (autom. Meldung DAB-Status einschalten)	49
4.4 <i>get_tii</i> (Sender-Identifikationsinformation angeben)	18	4.26 <i>get_active_info</i> (Aktivinformation angeben)	51
4.5 <i>notify_tii</i> (Sender-Identifikationsinformation-Meldung)	20	4.27 <i>notify_active_info</i> (Aktivinformation-Meldung)	52
4.6 <i>select_tii</i> (Sender-Identifikationsinformation auswählen)	22	4.28 <i>notify_service_following</i> (<i>service-following</i> -Meldung)	52
4.7 <i>get_pad</i> (programmbezogene Daten angeben)	23		
4.8 <i>notify_pad</i> (programmbezogene Daten-Meldung)	24		
4.9 <i>select_pad</i> (programmbezogene Daten auswählen)	25		
4.10 <i>get_figs</i> (Schnell-Informationsgruppen angeben)	26		

4.29 <i>manufacturer_specific_command</i> (herstellerspezifischer Befehl)	53	6 Richtlinie für Datenfelder, die für zukünftige Verwendung (Rfus) oder Zusätze (Rfas) reserviert sind, und Tabelleneinträge, die für zukünftige Definitionen (Rfds) reserviert sind	62
4.30 <i>manufacturer_specific_notification</i> (herstellerspezifische Meldung)	54	7 Abkürzungen	63
4.31 <i>notify_error_message</i> (Fehlermeldung-Meldung)	54	8 Literaturhinweise	63
5 Beispiel von DCSR-Codierung	56		

1 Allgemeine Einführung

1.1 Anwendungsbereich

Dieses Schriftstück beschreibt einen Befehlssatz, der für die Steuerung von DAB-Empfängern benutzt werden sollte, und auch die Codierung der Befehle. Dieser Befehlssatz ist dazu bestimmt, in verschiedenen physikalischen Bussystemen benutzt zu werden. Die Codierung sollte an verschiedenen physikalischen Schnittstellen transparent konvertiert werden.

Auswahl-Konfliktmanagement, dynamische Bandbreitenprobleme und die Adressierung der Baueinheiten liegen nicht im Anwendungsbereich dieser Norm.

1.2 Struktur der DCSR

Das DCSR-Schriftstück beschreibt den Befehlsvorrat, der aus drei Arten von Nachrichten besteht:

Der „Befehl“ (en: Command) wird von einer Steuereinrichtung benutzt, um den DAB-Empfänger anzuweisen, eine bestimmte Aktion durchzuführen, mit der er eine bestimmte Information gibt oder einen bestimmten Zustand einnimmt.

Die „Antwort“ (en: Response) enthält nur eine sofortige Reaktion auf den „Befehl“ ($t1 < 100$ ms, die Übertragungszeit des Busses nicht eingeschlossen), z. B. akzeptiert, zurückgewiesen, belegt. Wenn die endgültige Antwort nicht innerhalb von 100 ms gegeben werden kann, muss innerhalb der 100 ms eine Interimsantwort gegeben werden. $t1x$ muss Null sein, wenn keine Interimsantwort gegeben wird. Diese Antwort ist obligatorisch und muss an den Absender des Befehls gegeben werden.

Die „Meldung“ (en: Notification) enthält die vollständige Antwort auf den „Befehl“. Die Befehle werden vom Standpunkt des Empfängers aus betrachtet. Die Meldungen werden an den Controller oder an den angegebenen Ausgang gesendet.

Ein neuer Befehl darf nicht gesendet werden, bevor die endgültige Antwort auf den vorhergehenden Befehl empfangen wurde ($t2 \geq 0$).

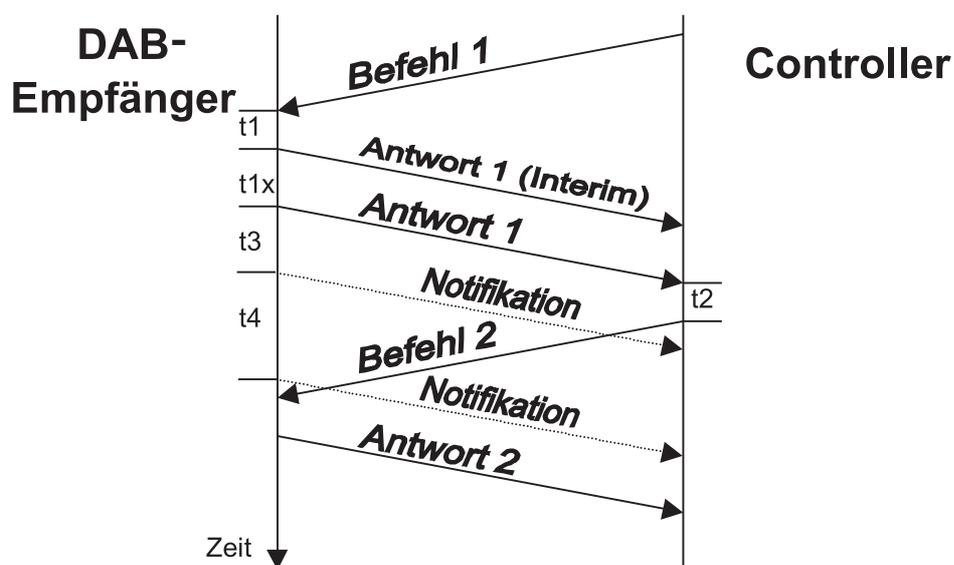


Bild 1 – Allgemeine Befehl-Antwort-Struktur

ANMERKUNG Wenn nicht anders angegeben, sind alle Parameterfelder immer vorhanden.
Die DCSR-Struktur wird wie folgt codiert (Kategoriecode, Referenzcode, Parameter):



2 Referenztabelle

		Kategorie- code	Referenz- code
get_receiver_capability	Empfängermerkmale angeben	0x1	0x01
tune	abstimmen	0x1	0x02
get_tii	Sender-Identifikationsinformation angeben	0x1	0x03
select_tii	Sender-Identifikationsinformation auswählen	0x1	0x04
get_pad	programmbezogene Daten angeben	0x1	0x05
select_pad	programmbezogene Daten auswählen	0x1	0x06
get_figs	Schnell-Informationsgruppen angeben	0x1	0x07
select_figs	Schnell-Informationsgruppen auswählen	0x1	0x08
get_channel	Kanal angeben	0x1	0x09
select_channel	Kanal auswählen	0x1	0x0A
get_selection_status	Auswahlstatus angeben	0x1	0x0B
search_for_ensemble	Ensemblesuche	0x1	0x0C
set_drc	Dynamikbereichsregelung einschalten	0x1	0x0D
get_audio_info	Audioinformation angeben	0x1	0x0E
get_dab_status	DAB-Status angeben	0x1	0x0F
set_dab_status_auto_notification	autom. Meldung DAB-Status einschalten	0x1	0x10
get_active_info	Aktivinformation angeben	0x1	0x11
manufacturer_specific_command	herstellerspezifischer Befehl	0x1	0x20
accepted	akzeptiert	0x2	0x01
rejected	zurückgewiesen	0x2	0x02
interim	interim	0x2	0x03
command_not_implemented	Befehl nicht implementiert	0x2	0x04
busy	belegt	0x2	0x05
syntax_error	Syntaxfehler	0x2	0x06
notify_receiver_capability	Empfängermerkmale-Meldung	0x3	0x01
notify_tii	Sender-Identifikationsinformation-Meldung	0x3	0x03

		Kategorie- code	Referenz- code
notify_pad	programmbezogene Daten-Meldung	0x3	0x05
notify_fig	Schnellinformation-Meldung	0x3	0x07
notify_channel	Kanal-Meldung	0x3	0x09
notify_selection_status	Auswahl-Status-Meldung	0x3	0x0B
notify_search_for_ensemble	Ensemblesuche-Meldung	0x3	0x0C
notify_audio_info	Audio-Information-Meldung	0x3	0x0E
notify_dab_status	DAB-Status-Meldung	0x3	0x0F
notify_active_info	aktive Information-Meldung	0x3	0x11
notify_service_following	nahtloser Empfang-Meldung	0x3	0x12
manufacturer_specific_notification	herstellerspezifische Meldung	0x3	0x20
notify_error_message	Fehlermeldung	0x3	0x30

Kategoriecode 0x1: Befehl
0x2: Antwort
0x3: Meldung

3 Antworten

3.1 *accepted* (akzeptiert)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er mit dem Ausführen des Befehls beginnen kann. Es darf keine notify_error_message folgen, Meldungen dürfen aber folgen.

Syntax:

accepted ()

Parameter:

keine

3.2 *rejected* (zurückgewiesen)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er den Befehl nicht verarbeiten kann (z. B. Parameter außerhalb des Bereiches).

Es folgt eine notify_error_message, oder im Fall eines manufacturer_specific_commands folgt eine manufacturer_specific_notification (die eine Fehlermeldung enthält).

Syntax:

rejected ()

Parameter:

keine

3.3 *interim* (interim)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er die Ausführung des Befehls innerhalb von 100 ms nicht beginnen kann. Anschließend an die Anfangsantwort *interim* darf der Empfänger keine zusätzlichen *interim*-Antworten senden. Der Empfänger muss eine endgültige Antwort senden, wenn die Ausführung des Befehls beginnt. Weitere Befehle sind nicht erlaubt, bevor die Abschlussantwort empfangen wurde.

Syntax:

interim ()

Parameter:

keine

3.4 *command_not_implemented* (Befehl nicht implementiert)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er den Befehl nicht verarbeiten kann, weil dieser Befehl im Empfänger nicht implementiert ist.

Es folgt keine Meldung.

Syntax:

command_not_implemented ()

Parameter:

keine

3.5 *busy* (belegt)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er im Augenblick den Befehl nicht verarbeiten kann, weil der Empfänger belegt ist. Der Controller sollte diesen Befehl später erneut senden.

Es folgt keine Meldung.

Syntax:

busy ()

Parameter:

keine

3.6 *syntax_error* (Syntaxfehler)

Kategorie:

Antwort (obligatorisch)

Zweck:

Der Empfänger zeigt dem Controller an, dass er den Befehl nicht verarbeiten kann, weil der empfangene Befehl einen Syntaxfehler enthält (z. B. eine falsche Anzahl der Parameter).

Es folgt keine Meldung.

Syntax:

syntax_error ()

Parameter:

keine

4 Befehle und Meldungen

Alle bedeutungslosen Befehlsparameter müssen vom Empfänger ignoriert (als „don't care“ behandelt) werden.

4.1 *get_receiver_capability* (Empfängermerkmale angeben)

Kategorie:

Befehl (obligatorisch)

Zweck:

Der Controller fordert vom DAB-Empfänger die Angabe der Empfängermerkmale.

Syntax:

get_receiver_capability ()

Parameter:

keine

Reaktion:

Antwort (accepted)

 notify_receiver_capability(...)

 Der Empfänger teilt seine Merkmale mit.

Antwort(rejected)

 notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.2 *notify_receiver_capability* (Empfängermerkmale-Meldung)

Kategorie:

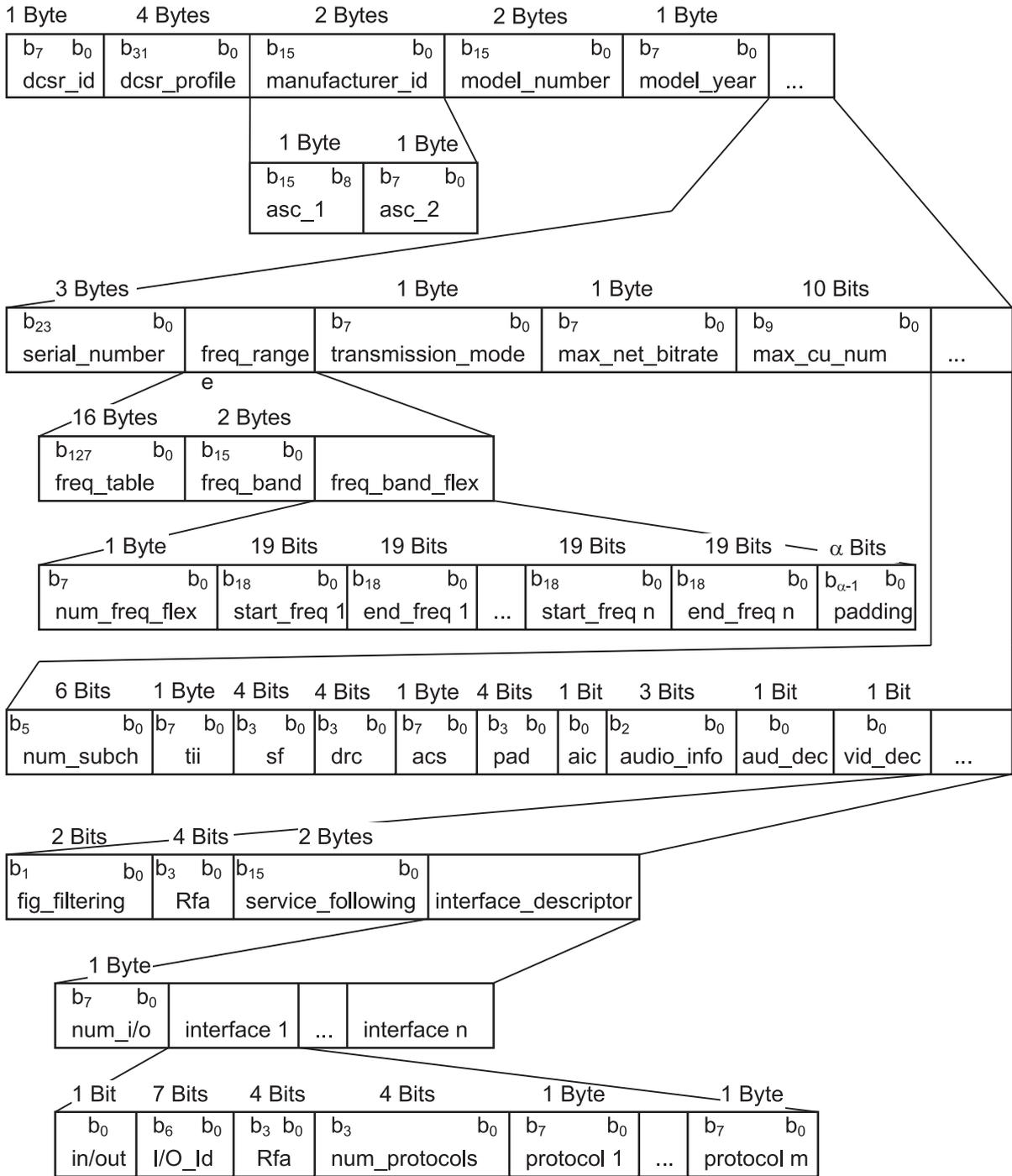
Meldung (obligatorisch)

Zweck:

Der DAB-Empfänger teilt seine Merkmale dem Controller mit.

Syntax:

notify_receiver_capability (dcsr_id, dcsr_profile, manufacturer_id, model_number, model_year, serial_number, freq_range, transmission_mode, max_net_bitrate, max_cu_num, num_subch, tii, sf, drc, acs, pad, aic, audio_info, aud_dec, vid_dec, fig_filtering, Rfa, service_following, interface_descriptor)



Parameter:

dcsr_id (DCSR-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld muss die DCSR-Identifikation (1 bis 254) angeben. Die DCSR-Identifikation bestimmt die unterstützte DCSR-Version. Da die DCSR-Versionen rückwärts kompatibel sind, bedeutet eine bestimmte *dcsr_id*, dass alle niedrigeren *dcsr-ids* auch unterstützt werden.

b₇ ... b₀	
0000 0000	kennzeichnet „keine Information“
0000 0001	DCSR-Version 1
0000 0010	DCSR-Version 2
...	...
1111 1110	DCSR-Version 254
1111 1111	Rfd (Abbruchmöglichkeit)

dcsr_profile (DCSR-Profil)

Dieses 32-Bit-Feld zeigt an, welche wahlfreien Befehle und Meldungen im Empfänger implementiert sind.

Bit-Flag	
b0	get_tii
b1	get_pad
b2	select_tii
b3	select_pad
b4	select_figs
b5	set_drc
b6	get_audio_info
b7	get_channel
b8	notify_channel
b9	manufacturer specific command
b10..b31	Rfa

Die Flags b_i ($i = 0$ bis 31) müssen wie folgt codiert werden.

- 0: Wahlfreier Befehl nicht implementiert
- 1: Wahlfreier Befehl implementiert

manufacturer_id (Hersteller-Identifikation)

Dieses als 2 ASCII-Zeichen codierte 16-Bit-Feld muss, basierend auf CRIN (Car Radio Identifier Number, siehe [5]), die Hersteller-Identifikation angeben.

ASC_1: Dieses als ASCII-Zeichen codierte 8-Bit-Feld muss das erste Zeichen der Hersteller-Identifikation (A-Z) nach [5] angeben.

ASC_2: Dieses als ASCII-Zeichen codierte 8-Bit-Feld muss das zweite Zeichen der Hersteller-Identifikation (A-Z) nach [5] angeben.

model_number (Modellnummer)

Dieses als vorzeichenlose Binärzahl codierte 16-Bit-Feld muss, basierend auf CRIN [5], die Modellnummer angeben. Dieses 16-Bit-Feld stellt die 4-stellige (dezimale) Modellnummer nach CRIN dar.

b₁₅ ... b₀	decimal model_number
0000 0000 0000 0000	0000
0000 0000 0000 0001	0001
...	...
0010 0111 0000 1111	9999
0010 0111 0001 0000	darf nicht benutzt werden
...	...
1111 1111 1111 1110	darf nicht benutzt werden
1111 1111 1111 1111	darf nicht benutzt werden

model_year (Modelljahr)

Dieses als ASCII-Zeichen codierte 8-Bit-Feld muss, basierend auf [5], das Modelljahr angeben.

serial_number (Seriennummer)

Dieses als vorzeichenlose Binärzahl codierte 24-Bit-Feld muss, basierend auf CRIN [5], die Seriennummer angeben. Dieses 24-Bit-Feld ist die binäre Darstellung der 7-stelligen (dezimalen) Seriennummer nach CRIN.

b₂₃	...	b₀	decimal serial_number
0000 0000 0000 0000 0000 0000			000 0000
...			...
1001 1000 1001 0110 0111 1111			999 9999
1001 1000 1001 0110 1000 0000			darf nicht benutzt werden
...			...
1111 1111 1111 1111 1111 1111			darf nicht benutzt werden

freq_range (freq_table, freq_band, freq_band_flex) (Frequenzbereich)

Die Struktur *freq_range* liefert Informationen darüber, welche Abstimmung-/Suchlauf-Strategien vom Empfänger zur Verfügung gestellt werden und welche Frequenzen/Frequenzbänder unterstützt werden.

freq_table (Frequenztabelle)

Dieses 128-Bit-Feld zeigt die vordefinierte Frequenztabelle an, die für den Befehl *search_for_ensemble* benutzt wird. Wenn dem Frequenzband eine Frequenztabelle zugeordnet ist, die vom Empfänger nicht vollständig unterstützt wird, dann werden nur solche Frequenztableneinträge benutzt, die innerhalb des unterstützten Bandes liegen (das vollständige Ensemble muss innerhalb des unterstützten Bandes liegen).

Bit-Flag	
b0	16-kHz-Stufen
b1	CEPT-Frequenztabelle für Band III, [7]
b2	CEPT-Frequenztabelle für L-Band, [7]
b3	Canada-Frequenztabelle für L-Band
b4–b127	Rfa

Die Flags b_i ($i = 0$ bis 127) müssen wie folgt codiert werden.

- 0: Frequenztabelle nicht unterstützt
- 1: Frequenztabelle unterstützt

freq_band (Frequenzband)

Dieses 16-Bit-Feld zeigt die unterstützten vordefinierten DAB-Frequenzbänder an.

Bit-Flag	Frequenzband (freq_band)
b0	Band III (174,000 MHz bis 240,000 MHz)
b1	L-Band (1 452,000 MHz bis 1 492,000 MHz)
b2–b15	Rfa

Die Flags b_i ($i = 0$ bis 15) müssen wie folgt codiert werden.

0: Frequenzband nicht unterstützt

1: Frequenzband unterstützt

freq_band_flex (num_freq_flex, start_freq x, end_freq x, padding) (flexible Frequenzbänder)

Die Struktur *freq_band_flex* liefert die Möglichkeit, für den Empfänger zusätzlich zur Verfügung gestellte DAB-Frequenzbänder zu signalisieren, die nicht durch den Parameter *freq_table* (siehe oben) angezeigt werden können. Wenn keine zusätzlichen Frequenzbänder zur Verfügung gestellt werden, muss *num_freq_flex* auf Null gesetzt werden. *freq_band_flex* besteht aus den folgenden Feldern:

num_freq_flex: Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld muss die Anzahl von zusätzlichen Bändern angeben, die durch eine Liste der Start- und Stopfrequenzen (maximal 255) definiert sind. Der Wert ‚0‘ bedeutet, dass es kein unterstütztes zusätzliches Frequenzband gibt. In diesem Fall sind die *start_freq-x*- und die *end_freq-x*-Felder nicht vorhanden.

start_freq x: Dieses als vorzeichenlose Binärzahl codierte 19-Bit-Feld muss die Startfrequenz (in Vielfachen von 16 kHz) eines zusätzlichen empfängerdefinierten Bandes *x* angeben, siehe [1].

end_freq x: Dieses als vorzeichenlose Binärzahl codierte 19-Bit-Feld muss die Endfrequenz (in Vielfachen von 16 kHz) eines zusätzlichen empfängerdefinierten Bandes *x* angeben, siehe [1].

padding: Dieses variable Feld von (auf Null gesetzten) Auffüllbits dient dazu, eine byteabgestimmte Codierung zu gewährleisten. Die Länge dieses Feldes wird wie folgt berechnet:

$$\text{Anzahl der Auffüllbits} = q \cdot 8 \text{ Bits} - \text{num_freq_flex} \cdot 2 \cdot 19 \text{ Bits}$$

Dabei ist:

$$0 \leq \text{Anzahl der Auffüllbits} < 8$$

q ein ganzzahliger Wert.

transmission_mode (Übertragungsmode)

Dieses 8-Bit-Feld zeigt die Modes an, die unterstützt werden. Es ist möglich, dass nicht jeder Empfänger jeden Übertragungsmode in jedem Frequenzband unterstützt.

Bit-Flag	
b0	Übertragungsmode I
b1	Übertragungsmode II
b2	Übertragungsmode III
b3	Übertragungsmode IV
b4–b6	Rfa
b7	automatische Erkennung des Mode

Die Flags b_i ($i = 0$ bis 7) müssen wie folgt codiert werden:

0: Mode nicht unterstützt

1: Mode unterstützt

max_net_bitrate (maximale Netto-Bitrate)

Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld muss die maximale Bitrate (ausschließlich FIC) am Ausgang des Viterbi-Decoders in Vielfachen von 8 Bits angeben.

max_cu_num (maximale Anzahl CUs)

Dieses als vorzeichenlose Binärzahl codierte 10-Bit-Feld muss die maximale Anzahl von CUs pro CIF angeben, die durch den Empfänger verarbeitet werden können.

num_subch (Anzahl Subkanäle)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss die maximale Anzahl von MSC-Subkanälen angeben, die decodiert werden können.

tii (Sender-Identifikationsinformation)

Dieses 8-Bit-Feld zeigt die Unterstützung von TII an.

Bit-Flag	
b0	main_id und sub_id
b1	Feldstärkeinformation
b2	Trägerinformation
b3	vom TII-Signal abgeleitete Verzögerungsinformation
b4	grobe Position, definiert durch die vom TII-Signal und der FIC-Information abgeleitete Region-ID
b5	Position des Empfängers, definiert durch vom TII-Signal und der FIC-Information abgeleitete Länge und Breite
b6	Rfa
b7	Rfa

Die Flags bi (i = 0 bis 5) müssen wie folgt codiert werden.

0: TII-Verarbeitung nicht unterstützt

1: TII-Verarbeitung unterstützt

sf (Abtastfrequenz)

Dieses 4-Bit-Feld zeigt die Unterstützung von einzelnen Abtastfrequenzen für Audio an. Die Unterstützung der Abtastfrequenz 48 kHz ist obligatorisch, wenn ein Audiodecoder vorhanden ist.

Bit-Flag	
b0	24 kHz (LSF)
b1	Rfa
b2	Rfa
b3	Rfa

Die Flags bi (i = 0,1) müssen wie folgt codiert werden.

0: Abtastfrequenz nicht unterstützt

1: Abtastfrequenz unterstützt

drc (Dynamikbereichsregelung)

Dieses 4-Bit-Feld zeigt die Unterstützung der Steuerung des Dynamikbereiches an.

Bit-Flag	
b0	DRC (benutzt F-PAD, siehe [1])
b1	DRC (vom Empfänger erzeugt)
b2	Rfa
b3	Rfa

Die Flags b_i ($i = 0$ bis 3) müssen wie folgt codiert werden.

0: nicht unterstützt

1: unterstützt

acs (Zugriffs-Steuerungs-System)

Dieses 8-Bit-Feld zeigt die Unterstützung von Systemen für die Zugriffssteuerung (ACS) an.

Bit-Flag	
b0	NR-MSK [3]
b1	Eurocrypt EN 50094 [4]
b2	Rfa
b3	Rfa
b4	Rfa
b5	Rfa
b6	Rfa
b7	Rfa

Die Flags b_i ($i = 0$ bis 7) müssen wie folgt codiert werden.

0: System für Zugriffssteuerung nicht unterstützt

1: System für Zugriffssteuerung unterstützt

pad (programmbezogene Daten)

Dieses 4-Bit-Feld zeigt die Unterstützung der PAD-Ausgabe an.

Bit-Flag	
B0	PAD-Ausgangssignal des gewählten Audio-Subkanals, der zu dem eingebauten Audiodecoder gesendet wird
B1	PAD-Ausgangssignal von jedem gewählten Audio-Subkanal
B2	PAD-Ausgangssignal von einem nicht gewählten Audio-Subkanal
B3	PAD-Ausgangsströme von mehr als einem parallelen Audio-Subkanal

Die Flags b_i ($i = 0$ bis 3) müssen wie folgt codiert werden.

0: nicht unterstützt

1: unterstützt

aic (Hilfs-Informationskanal)

Dieses Flag zeigt die Unterstützung von AIC an.

b0 = 0: Der Empfänger kann die FIGs des AIC nicht zur Verfügung stellen.

b0 = 1: Der Empfänger kann die FIGs des AIC zur Verfügung stellen, solange die gewählten Subkanäle nicht die verfügbare Decodierungskapazität belegen.

audio_info (Audioinformation)

Dieses 3-Bit-Feld zeigt die Unterstützung von audio_info an.

Bit-Flag	
B0	audio_info des gewählten Audio-Subkanals, der zu dem eingebauten Tondecoder gesendet wird
B1	audio_info von jedem gewählten Audio-Subkanal
B2	audio_info von einem nicht gewählten Audio-Subkanal

Die Flags bi (i = 0 bis 2) müssen wie folgt codiert werden.

0: nicht unterstützt

1: unterstützt

aud_dec (Audiodecoder)

Dieses Flag zeigt an, dass ein Tonquellendecoder vorhanden ist.

b0 = 0: kein Tonquellendecoder vorhanden

b0 = 1: mindestens ein Tonquellendecoder vorhanden

vid_dec (Videodecoder)

Dieses Flag zeigt an, dass ein Videoquellendecoder vorhanden ist.

b0 = 0: kein Videoquellendecoder vorhanden

b0 = 1: mindestens ein Videoquellendecoder vorhanden

fig_filtering (FIG-Filterung)

Dieses 2-Bit-Feld zeigt den Grad der Unterstützung der Filterung von Schnell-Informationsgruppen (FIG) an.

b ₁ b ₀	
00	Empfänger kann FIG-Filterung nicht durchführen
01	Empfänger kann Basis-FIG-Filterung durch type/extension durchführen
10	Empfänger kann Basis-FIG-Filterung durch type/extension und FIG-Datenfeld-Headerbits (C/N, OE, P/D, D1, D2, TCId) durchführen
11	Empfänger kann erweiterte FIG-Filterung durch type/extension, FIG-Datenfeld-Headerbits (C/N, OE, P/D, D1, D2, TCId) und durch Bestimmen der Version der Dienstinformation (SIV, siehe [1]) durchführen

service_following

Dieses 16-Bit-Feld zeigt nahtlosen Empfang zu den verschiedenen Rundfunksystemen an. Wenn der Empfänger nahtlosen Empfang zu FM oder AM unterstützt, dann wird ein anderer Befehlssatz benötigt, um den FM- oder AM-Teil des Empfängers zu steuern.

Bit-Flag	
b0	DAB-Ensemble, keine lokalen Fenster
b1	DAB-Signal, mit lokalen Fenstern
b2	Rfa
b3	Rfa
b4	FM mit RDS
b5	FM ohne RDS
b6	AM (MW in 9-kHz-Stufen & LW)
b7	AM (MW in 5-kHz-Stufen & KW)
b8..b15	Rfa

Die Flags b_i ($i = 0$ bis 15) müssen wie folgt codiert werden.

0: nahtloser Empfang nicht unterstützt

1: nahtloser Empfang unterstützt

***interface_descriptor* (num_i/o, interface x)** (Schnittstellen-Deskriptor)

Diese *interface_descriptor*-Liste liefert die Information, welche Schnittstellen von dem DAB-Empfänger zur Verfügung gestellt werden und welche Protokolle an diesen Schnittstellen unterstützt werden.

num_i/o: Dieses 8-Bit-Feld zeigt die Gesamtzahl der Schnittstellen an, die vom Empfänger unterstützt werden (bidirektionale Schnittstellen müssen wie zwei getrennte Schnittstellen behandelt werden). Der Wert Null ist nicht zugelassen.

interface x (in/out, I/O_Id, num_protocols, protocol x)

Das *interface-x*-Feld liefert die Information über eine bestimmte Schnittstelle.

in/out: Dieses 1-Bit-Flag zeigt die Richtung der Schnittstelle an. Für bidirektionale Schnittstellen muss der Typ (I/O-Id) für ein und aus angezeigt werden.

0: Ausgang des Empfängers

1: Eingang des Empfängers

I/O-Id: Dieses als vorzeichenlose Binärzahl codierte 7-Bit-Feld muss den Typ der Schnittstelle (I/O-Identifikation) angeben, wie in der folgenden Tabelle beschrieben.

b₆ ... b₀	Physikalische Schnittstelle
000 0000	nicht festgelegt (Vorgabe)
000 0001	Lautsprecher
000 0010	Kopfhörer
000 0011	Tonleitung mit festem Pegel
000 0100	Tonleitung mit variablem Pegel
000 0101	Digitalton (optisch)
000 0110	Digitalton (elektrisch)
000 0111	IEEE 1394
000 1000	RF-Eingang
000 1001	Video-RGB
000 1010	Video-FBAS
000 1011	S-Video
000 1100	RS 232
000 1101	D2B
000 1110	IEEE 1284
000 1111	IEC958
001 0000	SPI
001 0001	I ² C
001 0010	USB
001 0011	IrDA
001 0100	Rfd
...	...
111 1110	Rfd
111 1111	Rfd (Abbruchmöglichkeit)

Die Position n der entsprechenden Schnittstelle in der interface_descriptor-Liste wird für weitere Referenzen benutzt. Für die erste Schnittstelle in dieser Liste muss $n = 1$ sein, für die letzte Schnittstelle $n = \text{num_i/o}$. Um innerhalb des Befehls die x -te Schnittstelle zu adressieren, die als Eingangs- oder Ausgangs-Schnittstelle benutzt wird, wird x als input_interface_reference bzw. output_interface_reference benutzt.

num_protocols: Dieses 4-Bit-Feld zeigt die Gesamtzahl der Protokolle an, die vom Empfänger an der einzelnen Schnittstelle unterstützt werden. Wenn dieser Wert auf Null gesetzt ist, wird kein Protokoll signalisiert. Dann sollte ein Vorgabe-Protokoll benutzt werden.

protocol x: Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld zeigt ein Protokoll an, das vom Empfänger an der einzelnen Schnittstelle unterstützt wird.

b₇ ... b₀	Protokoll
0000 0000	nicht festgelegt (Vorgabe); dieser Code sollte benutzt werden, wenn die physikalische Schnittstelle das Protokoll enthält
0000 0001	RDI-Mode geringer Kapazität
0000 0010	RDI-Mode hoher Kapazität (normaler Rahmen)
0000 0011	RDI-Mode hoher Kapazität (erweiterter Rahmen)
0000 0100	SPDIF
0000 0101	AES-EBU
0000 0110	I ² S
0000 0111	DCSR-Universalprotokoll für byteorientierte Schnittstellen
0000 1000	Rfd
...	...
1111 1110	Rfd
1111 1111	Rfd (Abbruchmöglichkeit)

4.3 *tune* (Abstimmen)

Kategorie:

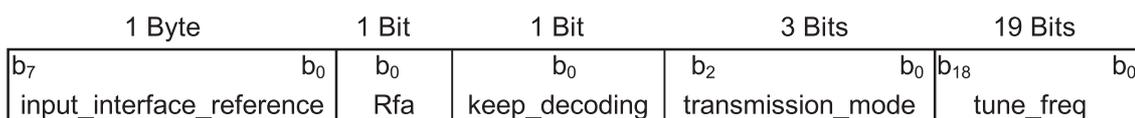
Befehl (obligatorisch)

Zweck:

Der Controller fordert den DAB-Empfänger auf, eine bestimmte Eingangs-Schnittstelle zu benutzen, um direkt auf eine spezielle DAB-Frequenz mit dem gegebenen Übertragungsmodus abzustimmen.

Syntax:

tune (input_interface_reference, Rfa, keep_decoding, transmission_mode, tune_freq)



Parameter:

input_interface_reference (Eingangs-Schnittstellen-Referenz)

Dieses vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss für die Auswahl die Eingangs-Schnittstelle (RF-Eingang) adressieren. Wenn input_interface_reference auf Null gesetzt ist, wechselt die Eingangs-Schnittstelle nicht. Wenn vorher keine Eingangs-Schnittstelle festgelegt war, wird der Vorgabe-Eingang gewählt.

keep_decoding (Decodieren beibehalten)

Dieses Flag zeigt an, ob der Empfänger versuchen sollte, das Decodieren ohne Unterbrechung fortzusetzen oder nicht, während dieser Befehl ausgeführt wird.

- b₀ = 0: nicht versuchen, das Decodieren fortzusetzen
- b₀ = 1: versuchen, das Decodieren fortzusetzen

transmission_mode (Übertragungsmodus)

Dieses 3-Bit-Feld legt den DAB-Übertragungsmodus fest.

b₂ ... b₀	
000	den aktuellen Übertragungsmodus beibehalten
001	Übertragungsmodus I
010	Übertragungsmodus II
011	Übertragungsmodus III
100	Übertragungsmodus IV
101	Rfd
110	Rfd
111	erkennt den entsprechenden Übertragungsmodus automatisch

Wenn der Übertragungsmodus für eine gegebene Frequenz nicht unterstützt wird, wird der Befehl zurückgewiesen.

Der `transmission_mode` wird z. B. auf „000“ gesetzt, wenn der Controller den Empfänger auffordert, nur den Antenneneingang zu wechseln oder zu einer anderen Frequenz in demselben MFN zu wechseln.

tune_freq (Frequenz abstimmen)

Dieses als vorzeichenlose Binärzahl codierte 19-Bit-Feld muss die DAB-Mittelfrequenz angeben, siehe [1].

`tune_freq = 0` bedeutet die aktuell abgestimmte Frequenz.

Reaktion:

Antwort(accepted)

`notify_dab_status()`

Der Empfänger liefert den DAB-Status.

Antwort(rejected)

`notify_error_message(...)`

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.4 **get_tii** (Sender-Identifikationsinformation angeben)

Kategorie:

Befehl (wahlfrei)

Zweck:

Der Controller fordert vom DAB-Empfänger die Ausgabe der Sender-Identifikationsinformation (TII).

Dieser Befehl mit seiner zugehörigen Meldung (`notify_tii`) erlaubt dem Controller, die Anzahl und Identifikation der Sender zu überwachen, die zu dem empfangenen DAB-Signal beitragen. Diese Daten können auf viele Arten dargestellt werden, der Empfänger signalisiert, welche Darstellungen er in seinen Empfängermerkmalen unterstützt und jedes davon kann in dem `tii_select`-Parameter ausgewählt werden. Die Anwendung dieser Darstellungen:

- Die `main_id` und `sub_id` können verwendet werden, um die Region zu identifizieren, in der das DAB-Signal augenblicklich empfangen wird.
- Die Trägerinformation erlaubt dem Controller, die Verzögerungen von den Sendern zu schätzen oder eigene TII-Verarbeitung durchzuführen.
- Die Verzögerungsinformation ist ein Indikator für den Abstand zu jedem Sender und kann zur Durchführung der Triangulation benutzt werden.

- Die durch die Regionen-Identifikation angezeigte grobe Position hat Anwendung in MMI oder in Datenterminals.
- Die Position ist das Ergebnis der Triangulation einer Anzahl von Sendern und wird in Verkehrs-Managementsystemen angewandt.

Syntax:

get_tii (tii_select, Rfa, continuous_tii)

1 Byte		7 Bits		1 Bit
b ₇	b ₀	b ₆	b ₀	b ₀
tii_select		Rfa		continuous_tii

Parameter:

tii_select (Sender-Identifikationsinformation wählen)

Dieses 8-Bit-Feld zeigt die Art der TII-Daten an. Wenn alle Flags auf Null gesetzt sind, wird die Ausgabe angehalten. Jede darauf folgende Verwendung dieses Befehls mit derselben output_interface_reference verursacht, dass der tii_select-Parameter von dem neuen Wert ersetzt wird. Wenn tii_select ein Bit gesetzt hat, das in den Empfängermerkmalen nicht definiert ist, muss der Befehl mit einer Fehlermeldung „Parameterwert nicht unterstützt“ zurückgewiesen werden.

Bit-Flag	
b0	main_id und sub_id
b1	Trägerinformation
b2	vom TII-Signal abgeleitete Verzögerungsinformation
b3	grobe Position, definiert durch die vom TII-Signal und der FIC-Information abgeleitete Region-ID
b4	Position des Empfängers, definiert durch vom TII-Signal und der FIC-Information abgeleitete Länge und Breite
b5	Rfa
b6	Rfa
b7	Rfa

continuous_tii (kontinuierliche Sender-Identifikationsinformation)

Dieses Flag zeigt an, ob der Controller vom DAB-Empfänger die kontinuierliche Ausgabe der Sender-Identifikationsinformation (TII) anfordert.

b0 = 0: nur eine notify_tii folgt

b0 = 1: der Empfänger liefert notify_tii so oft wie möglich

Reaktion:

Antwort(accepted)

notify_tii(...)

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not implemented)

Antwort(syntax_error)

4.5 notify_tii (Sender-Identifikationsinformation-Meldung)

Kategorie:

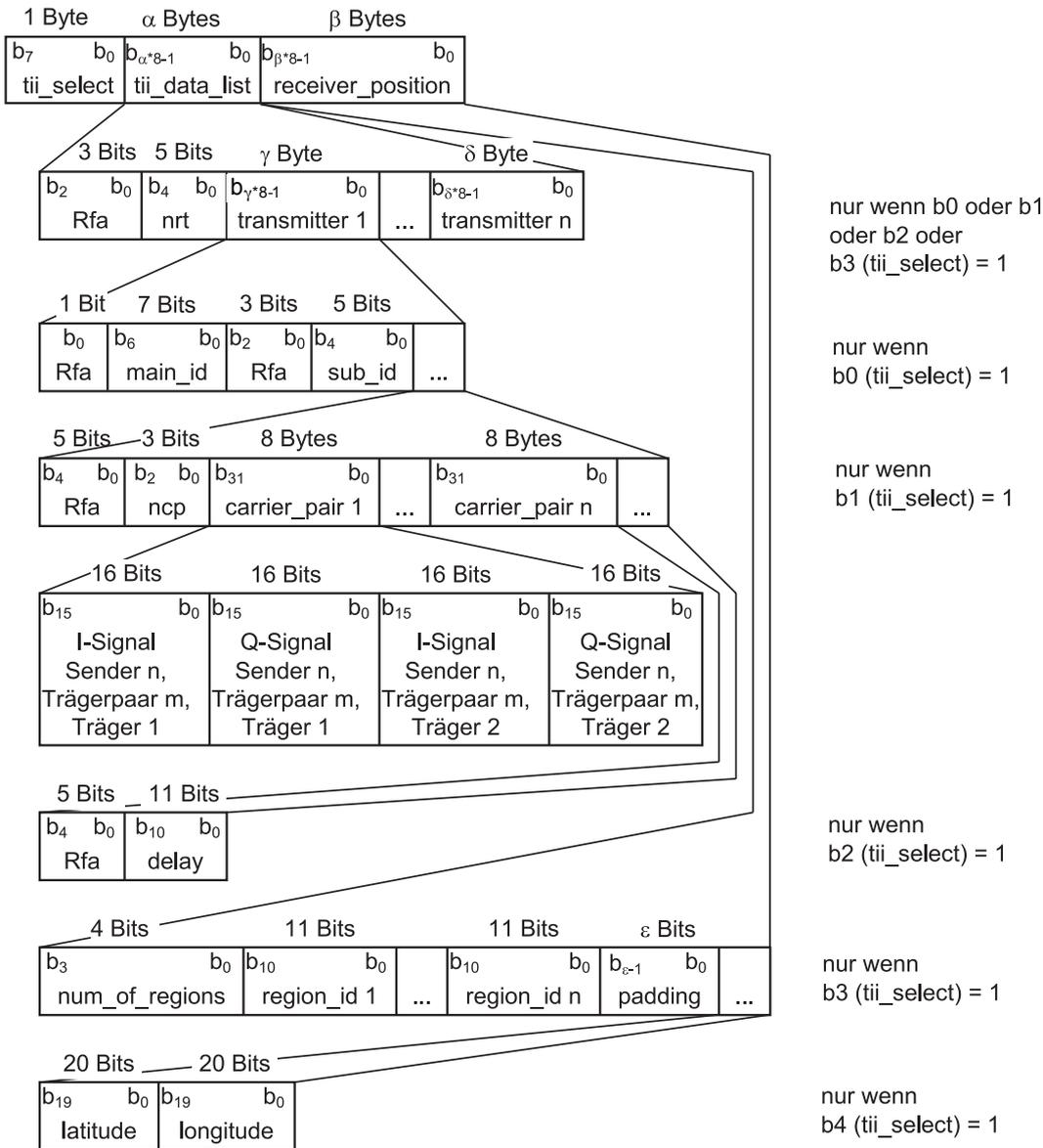
Meldung (optional)

Zweck:

Der DAB-Empfänger stellt die TII-Daten zur Verfügung. Bei großen Datenmengen können diese Daten in verschiedene Meldungen aufgeteilt werden. Wenn kein Sender identifiziert wird, dann darf keine Meldung erzeugt werden.

Syntax:

notify_tii (tii_select, tii_data_list, receiver_position)



Parameter:

tii_select (Sender-Identifikationsinformation auswählen)

siehe get_tii

tii_data_list (Sender-Identifikationsinformation-Datenliste)

nrt (Anzahl empfangener Sender)

Dieses 5-Bit-Feld zeigt die Anzahl der empfangenen Sender an. Diese im Bereich 1 bis 24 codierte vorzeichenlose Binärzahl muss die Anzahl der Sender angeben, für die Information (main_id, sub_id, Feldstärkeinformation, Trägerinformation, Verzögerungsinformation) zur Verfügung gestellt wird. Die anderen Codes sind reserviert.

Dieses Feld ist vorhanden, wenn b0 oder b1 oder b2 von `tii_select` gesetzt ist.

main_id (Haupt-Identifikation)

Dieses 7-Bit-Feld zeigt die `main_id` eines empfangenen Senders an.

Dieses Feld ist vorhanden, wenn b0 von `tii_select` gesetzt ist.

sub_id (Sub-Identifikation)

Dieses 5-Bit-Feld zeigt die `sub_id` eines empfangenen Senders an.

Dieses Feld ist vorhanden, wenn b0 von `tii_select` gesetzt ist.

ncp (Anzahl Trägerpaare)

Dieses als vorzeichenlose Binärzahl codierte 3-Bit-Feld muss die Anzahl der Trägerpaare angeben, für die für diesen Sender Information zur Verfügung gestellt wird.

Dieses Feld ist vorhanden, wenn b1 von `tii_select` gesetzt ist.

***I-Signal-Sender n, Trägerpaar m, Träger k,
Q-Signal-Sender n, Trägerpaar m, Träger k***

Diese als Zweierkomplement-Zahlen codierten 16-Bit-Felder müssen Real- oder Imaginärteil des FFT-Ergebnisses an den Abtastwerten des Null-Symbols für den k -ten Träger des m -ten Trägerpaars des n -ten Senders enthalten.

Diese Felder sind vorhanden, wenn b1 von `tii_select` gesetzt ist.

delay (Verzögerung)

Dieses als eine vorzeichenlose Binärzahl (im Bereich von 0 bis 2047) codierte 11-Bit-Feld muss die Schätzung der relativen Verzögerung des empfangenen Senders in Mikrosekunden angeben, bezogen auf den Sender, für den eine Verzögerung Null signalisiert wird (Verzögerung = 0).

Diese Felder sind vorhanden, wenn b2 von `tii_select` gesetzt ist.

receiver_position (Empfängerposition)

num_of_regions (Anzahl der Regionen)

Dieses als vorzeichenlose Binärzahl codierte 4-Bit-Feld muss die Anzahl von Regionen angeben, die sich auf die augenblickliche Position beziehen.

Dieses Feld ist vorhanden, wenn b3 von `tii_select` gesetzt ist.

region_id (Region-ID)

Dieses 11-Bit-Feld, das als ein 5-Bit-Oberteil und ein 6-Bit-Unterteil organisiert ist, muss die Region kennzeichnen (siehe [1]).

Dieses Feld ist vorhanden, wenn b3 von `tii_select` gesetzt ist und `num_of_regions` ungleich Null ist.

padding: (Auffüllen)

Dieses Variablenfeld mit auf Null gesetzten Auffüllbits wird benutzt, um eine an Bytes angepasste Codierung zu gewährleisten. Die Länge dieses Feldes wird wie folgt berechnet:

Anzahl von Auffüllbits = $q \cdot 8 - (\text{num_of_regions} \cdot 11 \text{ Bits} + 4 \text{ Bits})$

Dabei ist

$$0 \leq \text{Anzahl von Auffüllbits} < 8$$

q ist ein ganzzahliger Wert

latitude (Breite)

Dieses als Zweierkomplement-Zahl codierte 20-Bit-Feld muss die grobe Breite angeben (siehe [1]).

Dieses Feld ist vorhanden, wenn b4 von `tii_select` gesetzt ist.

longitude (Länge)

Dieses als Zweierkomplement-Zahl codierte 20-Bit-Feld muss die grobe Länge angeben (siehe [1]).

Dieses Feld ist vorhanden, wenn b4 von `tii_select` gesetzt ist.

4.6 select_tii (Sender-Identifikationsinformation auswählen)

Kategorie:

Befehl (wahlfrei)

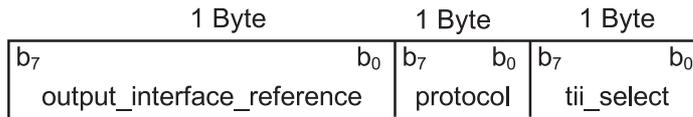
Zweck:

Der Controller fordert vom DAB-Empfänger die kontinuierliche Ausgabe der Sender-Identifikationsinformation (TII). Dieser Befehl erlaubt dem Controller, die TII-Daten an jede Ausgangs-Schnittstelle zu senden. Es wird kein notify_tii an den Controller geschickt.

siehe: get_tii

Syntax:

select_tii (output_interface_reference, protocol, tii_select)



Parameter:

output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses von dem Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss die Schnittstelle angeben, an die die Auswahl gerichtet werden muss. Wenn die output_interface_reference nicht definiert ist oder wenn es sich um einen Eingang handelt, muss der Befehl mit einer Fehlermeldung „Parameterwert ungültig“ zurückgewiesen werden.

protocol (Protokoll)

Dieses von dem Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben, das an der Ausgangs-Schnittstelle benutzt werden muss.

tii_select (auszuwählende Sender-Identifikationsinformation)

Dieses 8-Bit-Feld zeigt die Art der TII-Daten an. Wenn alle Flags auf Null gesetzt sind, wird die Ausgabe angehalten. Jede darauf folgende Verwendung dieses Befehls mit derselben output_interface_reference verursacht, dass der tii_select-Parameter von dem neuen Wert ersetzt wird. Wenn tii_select ein Bit gesetzt hat, das in den Empfängermerkmalen nicht definiert ist, muss der Befehl mit einer Fehlermeldung „Parameterwert nicht unterstützt“ zurückgewiesen werden.

Bit-Flag	
b0	main_id und sub_id
b1	Trägerinformation
b2	vom TII-Signal abgeleitete Verzögerungsinformation
b3	grobe Position, definiert durch die Region-ID, abgeleitet vom TII-Signal und der FIC-Information
b4	Position des Empfängers, definiert durch vom TII-Signal und der FIC-Information abgeleiteten Länge und Breite
b5	Rfa
b6	Rfa
b7	Rfa

Reaktion:

Antwort(accepted)

Der Empfänger stellt die TII-Daten der angegebenen Ausgangs-Schnittstelle zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not implemented)

Antwort(syntax_error)

4.7 get_pad (programmbezogene Daten angeben)

Kategorie:

Befehl (wahlfrei)

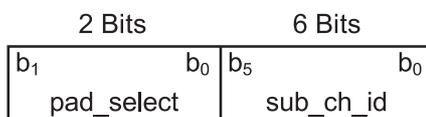
Zweck:

Der Controller fordert vom DAB-Empfänger die kontinuierliche Ausgabe der programmbezogenen Daten (PAD).

Dieser Befehl erlaubt im Zusammenhang mit der notify_pad-Meldung dem Controller, die PAD von einem Audio-Subkanal zu beurteilen (siehe [1], 7.4). Die Empfängermerkmale signalisieren, ob dieser Subkanal auf den augenblicklich gewählten Audio-Subkanal beschränkt ist oder nicht und ob ein oder mehrere Subkanäle für PAD verarbeitet werden können.

Syntax:

get_pad (pad_select, sub_ch_id)



Parameter:

pad_select (auszuwählende programmbezogene Daten)

Dieses 2-Bit-Feld zeigt die Art der PAD an. Die kontinuierliche Ausgabe kann durch Verwenden des Befehls mit dem pad_select-Parameter „nichts gewählt“ angehalten werden. Jede darauf folgende Verwendung dieses Befehls mit derselben output_interface_reference und sub_ch_id verursacht, dass der pad_select-Parameter durch den neuen Wert ersetzt wird.

b ₁	b ₀	
0	0	nichts gewählt
0	1	F-PAD
1	0	F-PAD + X-PAD
1	1	Rfd

sub_ch_id

(Subkanal-Identifikation): Dieses 6-Bit-Feld muss den Subkanal kennzeichnen, der die PAD enthält, die die Audio-Dienstkomponente übertragen (siehe [1]). Wenn der angegebene Subkanal kein gültiger Audio-Subkanal ist, muss der Befehl mit der Fehlermeldung „Parameterwert nicht verfügbar“ zurückgewiesen werden. Wenn der Befehl versucht, die maximale Anzahl der Subkanäle zu überschreiten, die verarbeitet werden kann, muss der Befehl mit der Fehlermeldung „nicht ausreichende Verarbeitungs-/Decodierungsleistung“ zurückgewiesen werden.

Reaktion:

Antwort(accepted)

notify_pad(...)

Der Empfänger stellt die PAD zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not implemented)

Antwort(syntax_error)

4.8 notify_pad (programmbezogene Daten-Meldung)

Kategorie:

Meldung (wahlfrei)

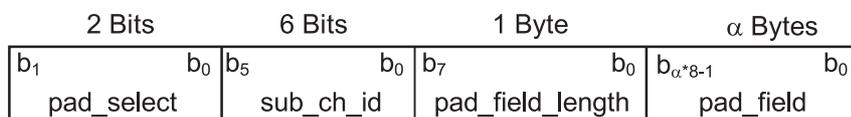
Zweck:

Der DAB-Empfänger stellt die angeforderte PAD eines speziellen Audio-Subkanals zur Verfügung.

Eine Meldung überträgt die PAD-Information eines MPEG-Rahmens. In einem MPEG-Rahmen wird die PAD in umgekehrter Reihenfolge übertragen. Die Meldung enthält diese PAD-Daten in ihrer logischen Reihenfolge (F-PAD-Byte L zuerst), sodass sie der Controller leicht interpretieren kann. Es sind entweder die F-PAD oder sowohl die F-PAD als auch die X-PAD verfügbar. Die F-PAD ist erforderlich, wenn X-PAD decodiert wird, um Kontext zu ihrem Format zu ergeben.

Syntax:

notify_pad (pad_select, sub_ch_id, pad_field_length, pad_field)



Parameter:

pad_select (programmbezogene Daten auswählen)

Dieses 2-Bit-Feld zeigt die Art der PAD. Wenn F-PAD + X-PAD angefordert wurde und X-PAD in dem angegebenen Subkanal nicht vorhanden ist, dann wird nur die F-PAD in dem pad_select-Feld signalisiert und in dem pad_field dargestellt.

b₁	b₀	
0	0	nichts gewählt
0	1	F-PAD
1	0	F-PAD + X-PAD
1	1	Rfd

sub_ch_id

(Subkanal-Identifikation): Dieses 6-Bit-Feld muss den Subkanal kennzeichnen, in dem die Dienstkomponente übertragen wird.

pad_field_length (programmbezogene Daten-Feldlänge)

Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld zeigt die Anzahl der Bytes in dem PAD-Feld an.

pad_field (programmbezogene Daten-Feld)

Das pad_field ist eine Sequenz von Bytes, die die gewünschte PAD enthält (siehe [1]).

4.9 select_pad (programmbezogene Daten auswählen)

Kategorie:

Befehl (wahlfrei)

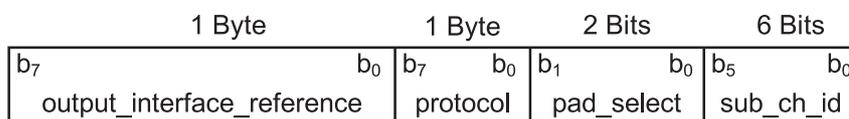
Zweck:

Der Controller fordert vom DAB-Empfänger die kontinuierliche Ausgabe der programmbezogenen Daten (PAD, siehe [1], 7.4). Dieser Befehl erlaubt dem Controller, den Empfänger aufzufordern, die PAD von einem Audio-Subkanal zu einer Ausgangs-Schnittstelle zu senden. Es wird keine notify_pad zu dem Controller gesendet.

Die Empfängermerkmale signalisieren, ob dieser Subkanal auf den augenblicklich gewählten Audio-Subkanal beschränkt ist oder nicht und ob ein oder mehrere Subkanäle für die PAD bearbeitet werden können.

Syntax:

select_pad (output_interface_reference, protocol, pad_select, sub_ch_id)



Parameter:

output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses 8-Bit-Feld (von den Empfängermerkmalen abgeleitet) muss die Schnittstelle angeben, an die die Auswahl gerichtet ist. Wenn die output_interface_reference nicht definiert ist oder wenn es sich um einen Eingang handelt, dann muss der Befehl mit der Fehlermeldung „Parameterwert ungültig“ zurückgewiesen werden.

protocol (Protokoll)

Dieses vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben, das an der Ausgangs-Schnittstelle benutzt werden muss.

pad_select (ausgewählte programmbezogene Daten)

Dieses 2-Bit-Feld zeigt die Art der PAD an. Die kontinuierliche Ausgabe kann durch diesen Befehl mit dem pad_select-Parameter „nichts gewählt“ angehalten werden. Jede darauf folgende Verwendung dieses Befehls mit derselben output_interface_reference und sub_ch_id verursacht, dass der pad_select-Parameter von dem neuen Wert ersetzt wird.

b₁	b₀	
0	0	nichts gewählt
0	1	F-PAD
1	0	F-PAD + X-PAD
1	1	Rfd

sub_ch_id

(Subkanal-Identifikation): Dieses 6-Bit-Feld muss den Subkanal kennzeichnen, der die PAD enthält, die die Audio-Dienstkomponente überträgt (siehe [1]). Wenn der angegebene Subkanal kein gültiger Audio-Subkanal ist, muss der Befehl mit der Fehlermeldung „Parameterwert nicht verfügbar“ zurückgewiesen werden. Wenn der Befehl versucht, die maximale Anzahl der Subkanäle zu überschreiten, die verarbeitet werden kann, muss der Befehl mit der Fehlermeldung „nicht ausreichende Verarbeitungs-/Decodierungsleistung“ zurückgewiesen werden.

Reaktion:

Antwort(accepted)

Der Empfänger stellt die PAD der angegebenen Ausgangs-Schnittstelle zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not implemented)

Antwort(syntax_error)

4.10 get_figs (Schnell-Informationsgruppen angeben)

Kategorie:

Befehl (obligatorisch)

Zweck:

Der Controller fordert vom DAB-Empfänger die kontinuierliche Ausgabe von (einer) speziellen Erweiterung(en) eines bestimmten FIG-Typs. Wenn der Empfänger in der Lage ist, die FIGs von dem AIC zur Verfügung zu stellen, ist dies nur möglich, solange der gewählte Subkanal nicht die erforderliche Decodierungskapazität belegt.

Syntax:

get_figs (Rfa, control, Rfa, type, fig_filters, extensions)

1 Bit	3 Bits	1 Bit	3 Bits	1 Byte	64 Bits
b ₀	b ₂ b ₁ b ₀	b ₀	b ₂ b ₁ b ₀	b ₇ b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀	b ₆₃ b ₆₂ b ₆₁ b ₆₀ b ₅₉ b ₅₈ b ₅₇ b ₅₆ b ₅₅ b ₅₄ b ₅₃ b ₅₂ b ₅₁ b ₅₀ b ₄₉ b ₄₈ b ₄₇ b ₄₆ b ₄₅ b ₄₄ b ₄₃ b ₄₂ b ₄₁ b ₄₀ b ₃₉ b ₃₈ b ₃₇ b ₃₆ b ₃₅ b ₃₄ b ₃₃ b ₃₂ b ₃₁ b ₃₀ b ₂₉ b ₂₈ b ₂₇ b ₂₆ b ₂₅ b ₂₄ b ₂₃ b ₂₂ b ₂₁ b ₂₀ b ₁₉ b ₁₈ b ₁₇ b ₁₆ b ₁₅ b ₁₄ b ₁₃ b ₁₂ b ₁₁ b ₁₀ b ₉ b ₈ b ₇ b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀
Rfa	control	Rfa	type	fig_filters	extensions

Parameter:

control (Steuerung)

Dieses 3-Bit-Feld muss die angeforderte Aktion angeben:

b ₂ ... b ₀	
000	hinzufügen (die FIG-type/extensions zu der augenblicklichen Liste)
001	löschen (die FIG-type/extensions in der augenblicklichen Liste)
010	alle wählen (verfügbare FIGs – es werden keine type/extensions und fig_filters angegeben, in diesem Fall sind entsprechende Parameter bedeutungslos)
011	alle löschen (vorher angeforderte FIGs – es werden keine type/extensions und keine fig_filter angegeben)
100	Rfd
101	Rfd
110	Rfd
111	Rfd

type (Typ)

Dieses 3-Bit-Feld kennzeichnet den FIG-Typ (0...7). Die Typen 2 und 4 sind für zukünftige Verwendungen reserviert. Siehe auch [1].

fig_filters (Schnell-Informationsgruppe-Filter)

Dieses 8-Bit-Feld muss zusätzliche Filter zum Anfordern bestimmter Gruppen von FIGs einschalten (z. B. nur solche, die Information über das abgestimmte Ensemble (OE = 0) geben). Dieses Feld kann nur benutzt werden, wenn es in *notify_receiver_capability* signalisiert wird.

Der Parameter „fig_filters“ ist für die FIG-Typen 0, 1 und 5 unterschiedlich definiert.

FIG-Typ 0:

Die folgenden zusätzlichen Filter werden für die Auswahl von FIGs vom Typ 0 angewendet.

C/N (b0, b1): Diese Flags müssen anzeigen:

a) ob Feld Typ 0 für die aktuelle oder die nächste Version der Multiplex-Konfiguration (gilt für Erweiterungen 0 bis 4, 7 und 8) wie folgt gilt:

b0 = 0: FIGs, die Information für die aktuelle Konfiguration enthalten, nicht senden (C/N = 0)

b0 = 1: FIGs, die Information für die aktuelle Konfiguration enthalten, senden (C/N = 0)

b1 = 0: FIGs, die Information für die nächste Konfiguration enthalten, nicht senden (C/N = 1)

b1 = 1: FIGs, die Information über die nächste Konfiguration enthalten, senden (C/N = 1)

b) oder die Serviceinformations-Version („version number“ in [1]) SIV des Typ-0-Feldes für Übertragungen von Datenbasisteilen (wie Frequenzinformation, gilt für Erweiterungen 6, 9, 11, 18, 21 bis 25, 27 und 30) wie folgt gilt:

b0 = 0: FIGs, die Änderungsereignisse enthalten, nicht senden

b0 = 1: FIGs, die Änderungsereignisse enthalten, senden

b1 = 0: FIGs, die kontinuierliche Übertragung von Datenbasisteilen enthalten, nicht senden (Beginn und Fortführung)

b1 = 1: FIGs, die kontinuierliche Übertragung von Datenbasisteilen enthalten, senden (Beginn und Fortführung)

Die C/N-Flags werden für die Erweiterungen 1 bis 4, 6 bis 9, 11, 18, 21 bis 25, 27 und 31 benutzt.

Für alle anderen Erweiterungen sind die Bits b0 und b1 für zukünftige Zusätze reserviert.

Detaillierte Beschreibungen siehe [1] und besonders [2].

Diese 2 Bits müssen auf „1“ gesetzt werden, wenn der Empfänger nicht in der Lage ist, erweiterte FIG-Filterung durchzuführen (in der *notify_receiver_capability* signalisiert), andernfalls muss dieser Befehl zurückgewiesen werden.

Wenn diese Bits beide auf „0“ gesetzt sind, muss dieser Befehl zurückgewiesen werden.

OE (b2, b3): Diese Flags müssen wie folgt anzeigen, ob sich die Information auf dieses oder ein anderes Ensemble bezieht:

OE = 0 (b2 auf „1“ gesetzt): dieses Ensemble;

OE = 1 (b3 auf „1“ gesetzt): anderes Ensemble (oder FM- oder AM-Dienste).

Die OE-Flags werden für die Erweiterungen 6, 12, 16, 17, 21, 23, 24 und 30 benutzt.

Für alle anderen Erweiterungen sind die Bits b2 und b3 für zukünftige Zusätze reserviert.

Detaillierte Beschreibungen siehe [1] und besonders [2].

P/D (b4, b5): Diese Flags müssen wie folgt anzeigen, ob die Dienst-Identifikationen (SIDs) im 16-Bit- oder im 32-Bit-Format dargestellt sind:

P/D = 0 (b4 auf „1“ gesetzt): 16-Bit-SID wird für Programm-Dienste benutzt;

P/D = 1 (b5 auf „1“ gesetzt): 32-Bit-SID wird für Daten-Dienste benutzt.

Die P/D-Flags werden für die Erweiterungen 2, 6, 8, 9, 23 und 24 benutzt.

Für alle anderen Erweiterungen sind die Bits b4 und b5 für zukünftige Zusätze reserviert.

Detaillierte Beschreibungen siehe [1] und besonders [2].

Bit-Flag	FIGs senden mit
b0	aktuelle Konfiguration (C/N=0) oder Änderungsereignisse der Datenbasis oder Rfa
b1	nächste Konfiguration (C/N=1) oder Beginn und Fortführung der Datenbasis oder Rfa
b2	OE = 0 oder Rfa
b3	OE = 1 oder Rfa
b4	PD = 0 oder Rfa
b5	PD = 1 oder Rfa
b6	Rfa
b7	Rfa

ANMERKUNG Das Setzen aller zutreffenden Flags im Bereich b0 bis b5 auf „0“ ergibt eine Antwort(syntax_error).

FIG-Typ 1:

Die folgenden zusätzlichen Filter werden auf die Auswahl von FIGs des Typs 1 angewendet:

OE (b2, b3): Diese Flags müssen anzeigen, ob sich die angeforderte Information auf dieses oder ein anderes oder auf beide Ensembles bezieht:

OE = 0 (b2 auf „1“ gesetzt): dieses Ensemble;

OE = 1 (b3 auf „1“ gesetzt): anderes Ensemble (oder FM- oder AM-Dienste).

Die Verwendung der OE-Flags ist nur für die Erweiterungen 0, 1, 4 und 5 definiert; für andere Erweiterungen sind die Bits b2 und b3 für zukünftige Zusätze reserviert.

Bit-Flag	FIGs senden mit
b0	Rfa
b1	Rfa
b2	OE = 0 oder Rfa
b3	OE = 1 oder Rfa
b4	Rfa
b5	Rfa
b6	Rfa
b7	Rfa

ANMERKUNG Das Setzen von sowohl b2 als auch b3 auf „0“ ergibt eine Antwort(syntax_error).

FIG-Typ 5:

Die folgenden zusätzlichen Filter werden auf die Auswahl von FIGs des Typs 5 angewendet:

D1 (b0, b1), D2 (b2, b3): Diese Flags werden für jede Erweiterung individuell definiert (siehe [1], 8.2).

TCId (Typ-Komponente-Identifikation, b6 bis b4): Dieses 3-Bit-Feld muss eine von 8 verschiedenen Dienstkomponenten kennzeichnen, die unter derselben Erweiterungsnummer übertragen werden können.

Alle TCIds (b7): Wenn dieses Flag auf „1“ gesetzt ist, wird der TCId-Filtermechanismus ausgeschaltet und alle Dienstkomponenten mit derselben Erweiterungsnummer, aber unterschiedlichen TCIDs werden gleichzeitig angefordert.

Bit-Flag	FIGs übertragen mit
b0	D1 = 0
b1	D1 = 1
b2	D2 = 0
b3	D2 = 1
b4 bis b6	TCId
b7	Alle TCIds

ANMERKUNG Das Setzen aller Flags b0 bis b3 auf „0“ ergibt eine Antwort(syntax_error).

extensions (Erweiterungen)

Dieses 64-Bit-Feld muss die angeforderten Erweiterungen des angeforderten FIG-Typs wie folgt kennzeichnen:

Bit-Flag	Angeforderte Erweiterungen
b0	0 or Rfa
b1	1 or Rfa
b2	2 or Rfa
...	... oder Rfas
b63	63 or Rfa

Gültigkeit:

Typ 0 → Erweiterungen: 0 bis 31 (bis jetzt sind die Erweiterungen 0 bis 12 und 16 bis 31 definiert, siehe [1]).

Typ 1 → Erweiterungen: 0 bis 7 (bis jetzt sind die Erweiterungen 0 bis 5 definiert, siehe [1]).

Typ 2 → reserviert, siehe [1].

Typ 3 → reserviert, siehe [1].

Typ 4 → reserviert, siehe [1].

Typ 5 → Erweiterungen: 0 bis 7 (bis jetzt sind die Erweiterungen 0 bis 2 definiert, siehe [1]).

Typ 6 → Erweiterungen: 0 bis 63 (alle 64 Erweiterungen sind definiert, siehe [1]).

Typ 7 → Felder „fig_filters“ und „extensions“ sind bedeutungslos (FIGs dieses Typs mit einem „Länge“-Wert 31 (die „End-Marke“) werden nicht zur Verfügung gestellt), siehe [1].

Alle Flags, die Erweiterungen beschreiben, die bis jetzt in Verbindung mit den entsprechenden FIG-Typen nicht definiert sind, werden für zukünftige Zusätze mit dem Parameter „extensions“ reserviert. Diese Rfas müssen auf „0“ gesetzt werden (siehe [2]).

Reaktion:

Antwort(accepted)

notify_fig(...)

Der Empfänger liefert die angeforderten FIGs.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.11 *notify_fig* (Schnell-Informationsgruppe-Notifikation)

Kategorie:

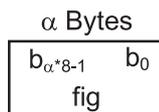
Meldung (obligatorisch)

Zweck:

Der DAB-Empfänger stellt die gewählten FIGs zur Verfügung, sobald sie empfangen wurden. Wenn der Empfänger in der Lage ist, die FIGs vom AIC zu liefern, ist dies nur solange möglich, wie die gewählten Subkanäle die notwendige Decodierungskapazität nicht völlig belegen.

Syntax:

notify_fig (*fig*)



Parameter:

fig (*siehe [1]*) (Schnell-Informationsgruppe)

Bemerkung: Der Empfänger liefert eine FIG (FIG-Header zuerst, danach FIG-Datenfeld) je Meldung. Es werden nur FIGs aus FIBs ohne CRC-Fehler geliefert. Es gibt keinen Weg, die Nichtbereitstellung von FIGs durch FIB-CRC-Fehler zu signalisieren.

4.12 *select_figs* (Schnell-Informationsgruppen auswählen)

Kategorie:

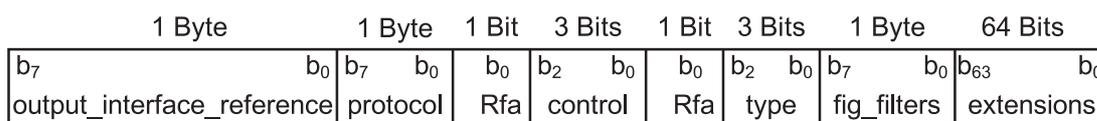
Befehl (wahlfrei)

Zweck:

Der Controller fordert den DAB-Empfänger zur kontinuierlichen Ausgabe einer oder mehrerer spezieller Erweiterungen eines bestimmten FIG-Typs an jeder Ausgangs-Schnittstelle auf. Es wird kein *notify_fig* zum Controller gesendet. Wenn der Empfänger in der Lage ist, die FIGs vom AIC zu liefern, wird dies nur möglich, solange die gewählten Subkanäle die notwendige Decodierungskapazität nicht völlig belegen.

Syntax:

select_figs (*output_interface_reference*, *protocol*, *Rfa*, *control*, *Rfa*, *type*, *fig_filters*, *extensions*)



Parameter:

output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses 8-Bit-Feld (von den Empfängermerkmalen abgeleitet, siehe 4.2) muss die Schnittstelle angeben, an die die Auswahl gerichtet werden muss. Wenn die angeforderte Schnittstelle mit „*notify_receiver_capability*“ als eine Eingangs-Schnittstelle beschrieben wird, würde eine Antwort(*syntax_error*) gegeben werden.

protocol (Protokoll)

Dieses von dem Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben, das an der Ausgangs-Schnittstelle benutzt werden muss.

control (Steuerung)

siehe *get_figs*

type (Typ)

siehe *get_figs*

fig_filters (Schnell-Informationsgruppe-Filter)

siehe get_figs

extensions (Erweiterungen)

siehe get_figs

Reaktionen:

Antwort(accepted)

Der Empfänger stellt die angeforderten FIGs der angegebenen Ausgangs-Schnittstelle zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.13 **get_channel** (Kanal angeben)

Kategorie:

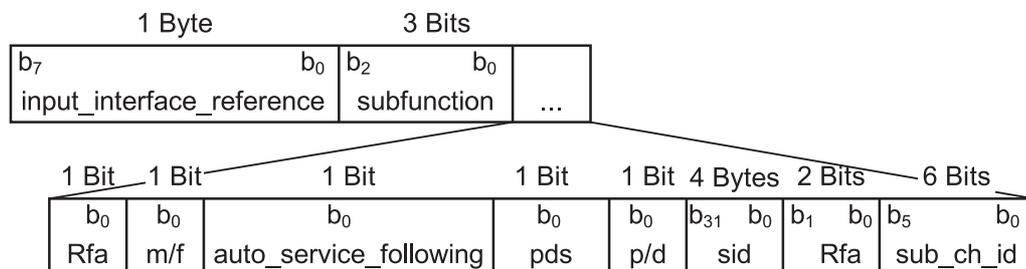
Befehl (wahlfrei)

Zweck:

Der Controller fordert vom DAB-Empfänger eine kontinuierliche Ausgabe eines speziellen Subkanals oder von FICs von dem abgestimmten Ensemble an derselben Schnittstelle, an der der Befehl empfangen wurde.

Syntax:

get_channel (input_interface_reference, subfunction, Rfa, mf_flag, auto_service_following, pds, p/d, sid, Rfa, sub_ch_id)



Parameter:

input_interface_reference (Eingangs-Schnittstellen-Referenz)

Dieses 8-Bit-Feld (abgeleitet von den Empfängermerkmalen, siehe 4.2) muss die Eingangs-Schnittstelle (z. B. „RF-input“) für die Auswahl angeben.

subfunction (Subfunktion)

Dieses als vorzeichenlose Binärzahl codiertes 3-Bit-Feld gibt die Subfunktion wie folgt an:

b₂ ... b₀	
000	clear: Die Ausgabe aller Auswahlen an der angegebenen Schnittstelle anhalten. Alle anderen Parameter sind in diesem Fall ungültig. Sogar wenn die angegebene Schnittstelle nicht benutzt wird, muss der Befehl akzeptiert werden.
001	remove: Die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle aus dem Ausgangsstrom nehmen. Wenn die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle nicht vorhanden ist, muss der Befehl zurückgewiesen werden.
010	append: Die angegebene Auswahl zu der angegebenen Ausgangs-Schnittstelle hinzufügen (multiplex). Es ist zulässig, zu einer anderen vorher unbenutzten Ausgangs-Schnittstelle hinzuzufügen.
011	replace: Die augenblickliche Auswahl von der angegebenen Ausgangs-Schnittstelle nehmen und die angegebene Auswahl dorthin liefern. Wenn es an der angegebenen Ausgangs-Schnittstelle keine Auswahl gibt, muss der Befehl zurückgewiesen werden.
100	new: Die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle ausgeben, wenn der Ausgang augenblicklich nicht benutzt wird. Andernfalls diesen Befehl zurückweisen.
101 bis 111	Rfd

m/f (Haupt-Dienstkanal/Schnell-Informationsgruppe)

(MSC/FIC-Flag): Wenn dieses Flag auf „1“ gesetzt ist, sendet der Empfänger den gesamten FIC und der Parameter „sub_ch_id“ wird ignoriert. Wenn dieses Flag nicht gesetzt ist, wird der angegebene Subkanal gesendet.

auto_service_following

Dieses Flag zeigt an, ob der nahtlose Empfang zu dem entsprechenden Übertragungssystem automatisch erfolgen sollte (wie mit „notify_receiver_capability“ gekennzeichnet, bei nicht befriedigender Signalqualität oder Multiplex-Rekonfigurationen). Wenn dieses Bit auf „1“ gesetzt wird, der Empfänger aber die *auto service following*-Merkmale nicht unterstützt, würde das Ergebnis eine Antwort(syntax_error) sein.

Wenn *auto service following* eingeschaltet sein sollte, müssen das „pds“-Flag, das „p/d“-Flag und die Dienst-Identifikation (Feld „sid“) für den Dienst signalisiert werden, der nahtlos empfangen werden sollte. Außerdem muss die Auswahl von mindestens der entsprechenden Primär-Dienstkomponente aktiv werden.

pds

Dieses Flag zeigt wie folgt an, ob die folgenden Parameter „p/d“ und „sid“ gültig sind:

b0 = 0: Parameter „p/d“ und „sid“ sind nicht gültig

b0 = 1: Parameter „p/d“ und „sid“ sind ungültig

p/d

Dieses Flag muss wie folgt anzeigen, ob die Dienst-Identifikation (auf das Feld „sid“ folgend) im 16-Bit- oder im 32-Bit-Format dargestellt wird:

b0 = 0: 16-Bit-SId

b0 = 1: 32-Bit-SId

sid

(Dienst-Identifikation): Dieses 32-Bit-Feld muss den Dienst kennzeichnen. Ein 16-Bit-SId (p/d = 0) muss mit 16 führenden Nullen erweitert werden, um in dieses 32-Bit-Feld zu passen. Dieser Parameter muss zur Verfügung gestellt werden, wenn der Empfänger z. B. nahtlosen Empfang unterstützen sollte, weil der Empfänger wissen muss, welchem dieser Dienste – der/die gewählte Subkanal/Dienstkomponente ist ein Teil davon – automatisch gefolgt werden sollte.

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den angeforderten Subkanal angeben (Subkanal-Identifikation, siehe [1]).

Reaktion:

Antwort(accepted)

Antwort(rejected)

 notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.14 notify_channel (Kanal-Meldung)

Kategorie:

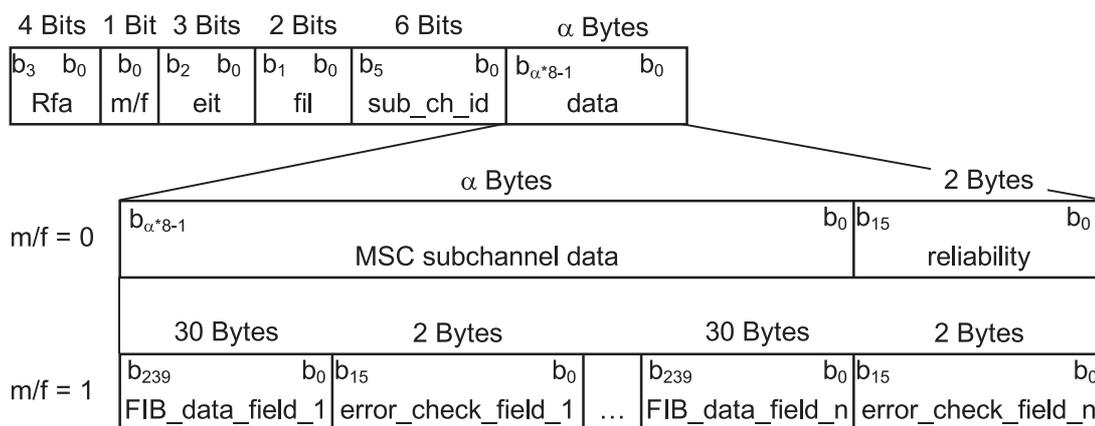
Meldung (wahlfrei)

Zweck:

Der DAB-Empfänger liefert den FIC oder die gewählten MSC-Subkanaldaten in derselben physikalischen Reihenfolge der Bits wie in dem DAB-Bitstrom.

Syntax:

notify_channel (Rfa, m/f, eit, fil, sub_ch_id, data)



Parameter:

m/f

(MSC/FIC-Flag): Wenn dieses Flag auf „1“ gesetzt ist, sendet der Empfänger den gesamten FIC und der Parameter „sub_ch_id“ wird ignoriert. Wenn dieses Flag nicht gesetzt ist, wird der angegebene Subkanal gesendet.

b0 = 0: MSC-Subkanaldaten werden in dem Datenfeld übertragen

b0 = 1: FIC-Daten werden in dem Datenfeld übertragen

eit

(Art der Fehleranzeige): Dieses 3-Bit-Feld muss wie folgt die Art der Daten angeben, die in dem error_check_field übertragen werden, wenn m/f auf „1“ gesetzt ist:

b₂ ... b₀	
000	keine Fehleranzeige, error_check_field ist reserviert
001	error_check_field enthält CRC wie empfangen
010	error_check_field enthält die bitweise Summe von empfangener und lokal errechneter CRC
011 bis 111	Rfd

fil

(erster/dazwischenliegend/letzter): Dieses 2-Bit-Feld signalisiert, ob das Datenfeld erster, dazwischenliegender, letzter oder einziger Teil des FIC oder der MSC-Subkanaldaten innerhalb 24 ms ist.

b₁	b₀	
0	0	dazwischenliegend (dazwischenliegender Teil des FIC oder der MSC-Subkanaldaten)
0	1	erster (erster Teil des FIC oder der MSC-Subkanaldaten)
1	0	letzter (letzter Teil des FIC oder der MSC-Subkanaldaten)
1	1	erster und letzter (einziger Teil des FIC oder der MSC-Subkanaldaten)

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den Subkanal angeben (Subkanal-Identifikation, siehe [1]), der in diesem Datenfeld übertragen wird. Dieses Feld ist reserviert, wenn m/f auf „1“ gesetzt ist.

data (Daten)

Dieses α -Bytes-Feld muss den FIC oder MSC-Subkanaldaten der Viterbi-Decoder-Ausgabe übertragen. Das zeitlich erste Datenbit des FIC oder MSC-Subkanals wird zuerst übertragen. Die physikalische Reihenfolge der Bits des DAB-Bitstromes muss beibehalten werden.

reliability (Zuverlässigkeit)

Dieses als vorzeichenlose Binärzahl codierte 16-Bit-Feld muss die Anzahl der korrigierten Fehler angeben, die durch Rückcodierung erkannt wurden. Ausnahmsweise muss der Code „1111 1111 1111 1111“ anzeigen, dass diese Information nicht signalisiert wird.

error_check_field (Fehlerprüffeld)

Dieses als vorzeichenlose Binärzahl codierte 16-Bit-Feld überträgt die im eit-Feld angegebenen Daten.

4.15 select_channel (Kanal auswählen)

Kategorie:

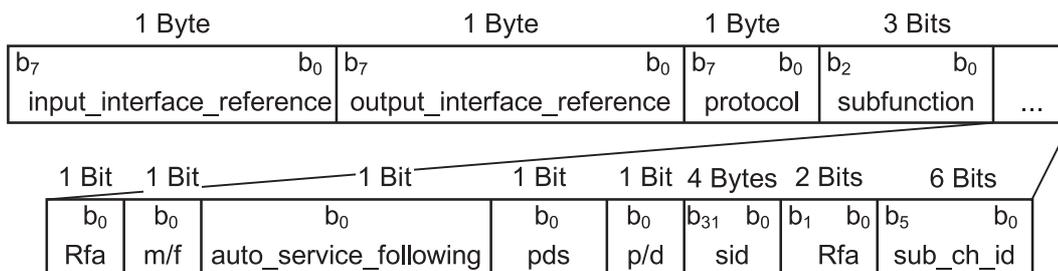
Befehl (obligatorisch)

Zweck:

Der Controller fordert vom DAB-Empfänger das Auswählen eines speziellen Subkanals oder FIC des abgestimmten Ensembles und die Weitergabe seines Inhaltes an eine spezielle Schnittstelle.

Syntax:

select_channel (input_interface_reference, output_interface_reference, protocol, subfunction, Rfa, m/f, auto_service_following, pds, p/d, sid, Rfa, sub_ch_id)



Parameter:

input_interface_reference (Eingangs-Schnittstellen-Referenz)

Dieses (von den Empfängermerkmalen abgeleitete, siehe 4.2) 8-Bit-Feld muss die Eingangs-Schnittstelle (z. B. „RF-Eingang“) für das Auswählen angeben.

output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses (von den Empfängermerkmalen abgeleitete, siehe 4.2) 8-Bit-Feld muss die Ausgangs-Schnittstelle (z. B. „IEC 60958“) für das Auswählen angeben.

protocol (Protokoll)

Dieses von dem Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben (z. B. „SPDIF“), das an dieser Ausgangs-Schnittstelle benutzt werden muss.

subfunction (Subfunktion)

Dieses als vorzeichenlose Binärzahl codierte 3-Bit-Feld gibt die Subfunktionen wie folgt an:

b ₂ ... b ₀	
000	clear: Die Ausgabe aller Auswahlen an der angegebenen Schnittstelle anhalten. Alle anderen Parameter sind in diesem Fall ungültig. Sogar wenn die angegebene Schnittstelle nicht benutzt wird, muss der Befehl akzeptiert werden.
001	remove: Die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle aus dem Ausgangsstrom nehmen. Wenn die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle nicht vorhanden ist, muss der Befehl zurückgewiesen werden.
010	append: Die angegebene Auswahl zu der angegebenen Ausgangs-Schnittstelle hinzufügen (multiplex). Es ist zulässig, zu einer anderen vorher unbenutzten Ausgangs-Schnittstelle hinzuzufügen.
011	replace: Die augenblickliche Auswahl von der angegebenen Ausgangs-Schnittstelle nehmen und die angegebene Auswahl dorthin liefern. Wenn es an der angegebenen Ausgangs-Schnittstelle keine Auswahl gibt, muss der Befehl zurückgewiesen werden.
100	new: Die angegebene Auswahl an der angegebenen Ausgangs-Schnittstelle ausgeben, wenn der Ausgang augenblicklich nicht benutzt wird. Andernfalls diesen Befehl zurückweisen.
101 bis 111	Rfd

m/f

(MSC/FIC-Flag): Wenn dieses Flag auf „1“ gesetzt ist, sendet der Empfänger den gesamten FIC und der Parameter „sub_ch_id“ wird ignoriert. Wenn dieses Flag nicht gesetzt ist, wird der angegebene Subkanal gesendet.

auto_service_following

Dieses Flag zeigt an, ob der nahtlose Empfang zu dem entsprechenden Übertragungssystem automatisch erfolgen sollte (wie mit „notify_receiver_capability“ gekennzeichnet, bei nicht befriedigender Signalqualität oder Multiplex-Rekonfigurationen). Wenn dieses Bit auf „1“ gesetzt wird, der Empfänger aber die auto-service-following-Merkmale nicht unterstützt, würde das Ergebnis eine Antwort(syntax_error) sein.

Wenn *auto service following* eingeschaltet sein sollte, müssen das „pds“-Flag, das „p/d“-Flag und die Dienst-Identifikation (Feld „sid“) für den Dienst signalisiert werden, der nahtlos empfangen werden sollte. Außerdem muss die Auswahl von mindestens der entsprechenden Primär-Dienstkomponente aktiv werden.

pds

Dieses Flag zeigt wie folgt an, ob die Parameter „p/d“ und „sid“ gültig sind:

b0 = 0: Parameter „p/d“ und „sid“ sind nicht gültig

b0 = 1: Parameter „p/d“ und „sid“ sind gültig

p/d

Dieses Flag muss wie folgt anzeigen, ob die Dienst-Identifikation (folgendes Feld „sid“) im 16-Bit- oder im 32-Bit-Format angegeben ist:

b0 = 0: 16-Bit-SId

b0 = 1: 32-Bit-SId

sid

(Dienst-Identifikation): Dieses 32-Bit-Feld muss den Dienst kennzeichnen. Eine 16-Bit-SId (p/d = 0) muss mit 16 führenden Nullen erweitert werden, um in dieses 32-Bit-Feld zu passen. Dieser Parameter muss zur Verfügung gestellt werden, wenn der Empfänger z. B. nahtlosen Empfang unterstützen sollte, weil der Empfänger wissen muss, welchem dieser Dienste – der/die gewählte Subkanal/Dienstkomponente ist ein Teil davon – automatisch gefolgt werden sollte.

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den angeforderten Subkanal angeben (Subkanal-Identifikation, siehe [1]).

Reaktion:

Antwort(accepted)

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.16 get_selection_status (Auswahlstatus angeben)

Kategorie:

Befehl (obligatorisch)

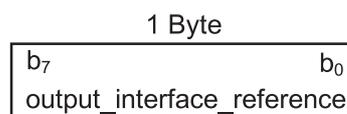
Zweck:

Der Controller fordert vom DAB-Empfänger den Auswahlstatus einer gegebenen `output_interface_reference`.

Syntax:

`get_selection_status (output_interface_reference)`

Parameter:



output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses von dem Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss die Schnittstelle adressieren, für die der Auswahlstatus angefordert wurde.

Reaktion:

Antwort(accepted)

notify_selection_status()

Der Empfänger stellt den Auswahlstatus einer gegebenen output_interface_reference zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.17 notify_selection_status (Auswahlstatus-Meldung)

Kategorie:

Meldung (obligatorisch)

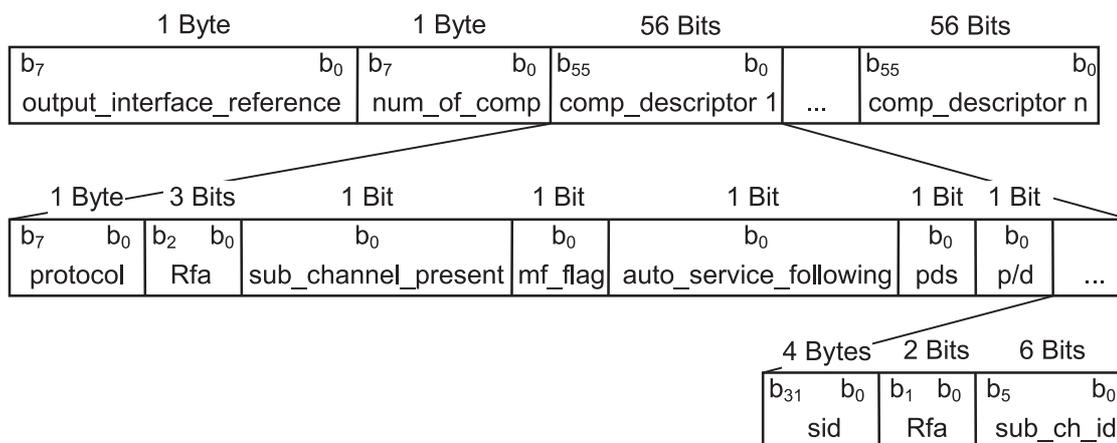
Zweck:

Der Empfänger stellt den Auswahlstatus der gegebenen output_interface_reference zur Verfügung.

Syntax:

notify_selection_status (output_interface_reference, num_of_comp, comp_descriptor 1, ... comp_descriptor n)

Parameter:



output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses (vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete) 8-Bit-Feld muss die Schnittstelle adressieren, zu der der Auswahlstatus gehört.

num_of_comp (Anzahl der Komponenten)

Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld gibt die Anzahl n der folgenden Komponenten-Deskriptoren an.

comp_descriptor x (1 ≤ x ≤ n) (Komponenten-Deskriptor x)

Die Auswahlstatus-Meldung besteht aus einer Anzahl (num_of_comp) von Komponenten-Deskriptoren (comp_descriptor1,...,comp_descriptor n). Diese Felder beschreiben alle Auswahlen, die sich auf die gegebene output_interface_reference beziehen. Ein Deskriptor besteht aus den folgenden Feldern:

protocol (Protokoll)

Dieses vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben, das an der Ausgangs-Schnittstelle für diese Auswahl benutzt wird.

sub_channel_present (Subkanal vorhanden)

b0 = 0: Der gewählte Subkanal ist augenblicklich nicht vorhanden.
 b0 = 1: Der gewählte Subkanal ist augenblicklich vorhanden.

mf_flag

(MSC_FIC_flag): Wenn dieses Flag gesetzt ist, gibt der Empfänger den vollständigen FIC aus. In diesem Fall sind das sub_ch_field, das auto_service_following_field und das status_field nicht gültig. Wenn dieses Flag nicht gesetzt ist, wird der angegebene Subkanal ausgegeben.

auto_service_following

Dieses Flag zeigt an, ob der nahtlose Empfang erfolgen wird, wenn das Signal schwach wird.

pds

Dieses Flag zeigt die Gültigkeit der folgenden Parameter.

b0 = 0: p/d- und sid-Parameter sind nicht gültig
 b0 = 1: p/d- und sid-Parameter sind gültig

p/d (Programm/Daten)

Dieses Flag zeigt wie folgt an, ob die Dienst-Identifikation ein 16-Bit-(Programmdienst) oder ein 32-Bit-(Datendienst)Format hat.

b0 = 0: 16-Bit-Sld, für Programmdienst benutzt
 b0 = 1: 32-Bit-Sld, für Datendienst benutzt

sid

(Dienst-Identifikation): Dieses 32-Bit-Feld muss den Dienst kennzeichnen. Ein 16-Bit-Sld (p/d = 0) muss mit 16 führenden Nullen erweitert werden, um in das 32-Bit-Feld zu passen.

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den Subkanal angeben, der an dieser speziellen Schnittstelle gewählt ist (Subkanal-Identifikation, siehe [1]).

4.18 search_for_ensemble (Ensemblesuche)

Kategorie:

Befehl (obligatorisch)

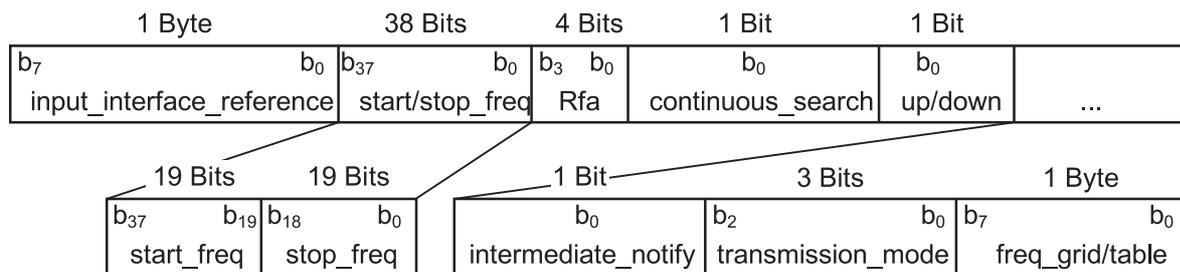
Zweck:

Der Controller fordert den DAB-Empfänger auf, nach einem neuen Ensemble zu suchen. Der Empfänger beendet die Suche, wenn ein Ensemble gefunden (der Übertragungsmodus erkannt) wurde. Diese Suche kann durch einen Abstimmbefehl beendet oder durch einen neuen Befehl search_for_ensemble aufgehoben werden.

Syntax:

search_for_ensemble (input_interface_reference, start/stop_freq, Rfa, continuous_search, up/down, intermediate_notify, transmission_mode, freq_grid/table)

Parameter:



input_interface_reference (Eingangs-Schnittstellen-Referenz)

Dieses vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss die Schnittstelle (RF-Eingang) für den Suchbefehl adressieren.

start/stop_freq (Start- und Stopffrequenz)

Dieses 38-Bit-Feld gibt die Start- und Stopffrequenz für die Suche eines Ensembles an. Die Codierung ist in [1] definiert. Die Startfrequenz darf größer sein als die Stopffrequenz. Wenn der Controller keine Start- und/oder Stopffrequenz angibt, dann sollte der Wert auf Null gesetzt werden.

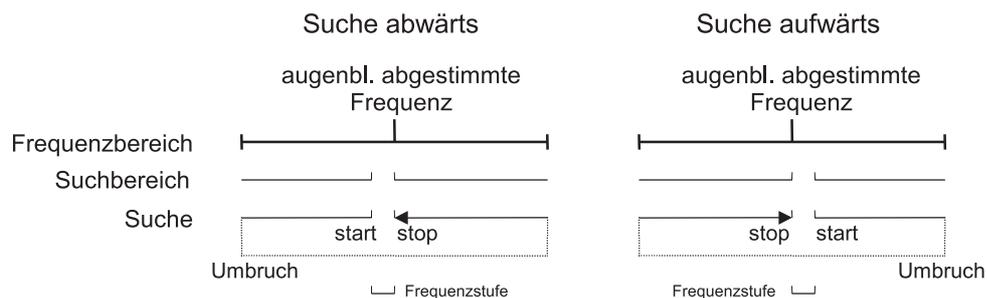
Auf Grund der angegebenen Frequenz gibt es in den beiden Frequenzfeldern vier mögliche Fälle, wie der Controller die Suche anfordern kann:

1) Start- und Stopffrequenz sind auf Null gesetzt

Der Controller fordert den Empfänger auf, eine Suche auf- oder abwärts von einer Stufe nach oder vor der augenblicklich abgestimmten Frequenz durchzuführen, Umbruch an den Grenzen der definierten Frequenztafel oder des Frequenzbandes und Fortsetzen der Suche bis die augenblicklich abgestimmte Frequenz wieder erreicht ist. Wenn der Empfänger nicht auf eine spezielle Frequenz abgestimmt ist (z. B. nicht aktivierter Zustand nach dem Einschalten), wird er die erste und letzte Frequenz als Start- und Stopffrequenz benutzen. Wenn keine Tabelle, sondern das 16-kHz-Raster angegeben ist, wird er die tiefste und höchste abstimmbare Frequenz eines unterstützten Frequenzbandes als Start- und Stopffrequenz benutzen.

Dieser Fall wird empfohlen, wenn der Controller die Frequenztafeln nicht kennt. Dies ermöglicht es, einen sehr einfachen Suchbefehl zu senden. Der Suchbereich enthält alle Frequenzen der Frequenztafel oder des Frequenzbandes.

start_freq=stop_freq=0

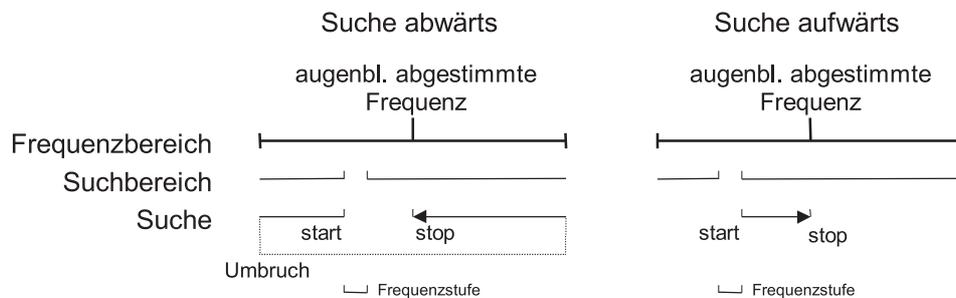


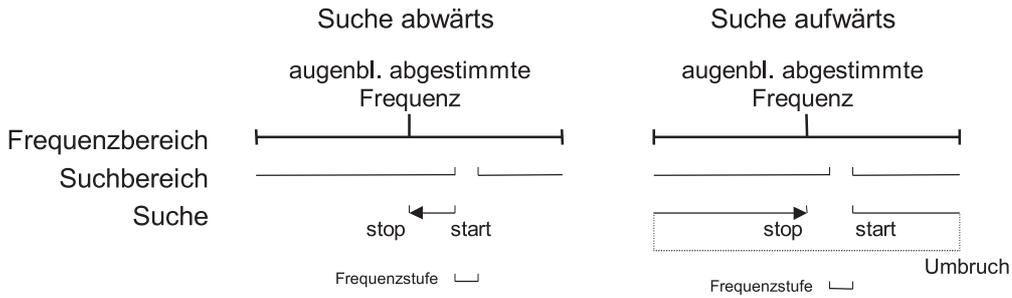
2) Startfrequenz ist definiert, Stopffrequenz ist auf Null gesetzt

Der Controller fordert den Empfänger auf, Suchlauf auf- oder abwärts von einer definierten Frequenz an durchzuführen, Umbruch an den Grenzen der definierten Frequenztafel oder des Frequenzbandes und Fortsetzen der Suche bis eine Stufe vor oder nach der Startfrequenz.

Dieser Fall wird empfohlen, wenn die Suche bei einer angegebenen Frequenz beginnen muss. Der Suchbereich enthält alle Frequenzen der Frequenztafel oder des Frequenzbandes.

start_freq, stop_freq=0



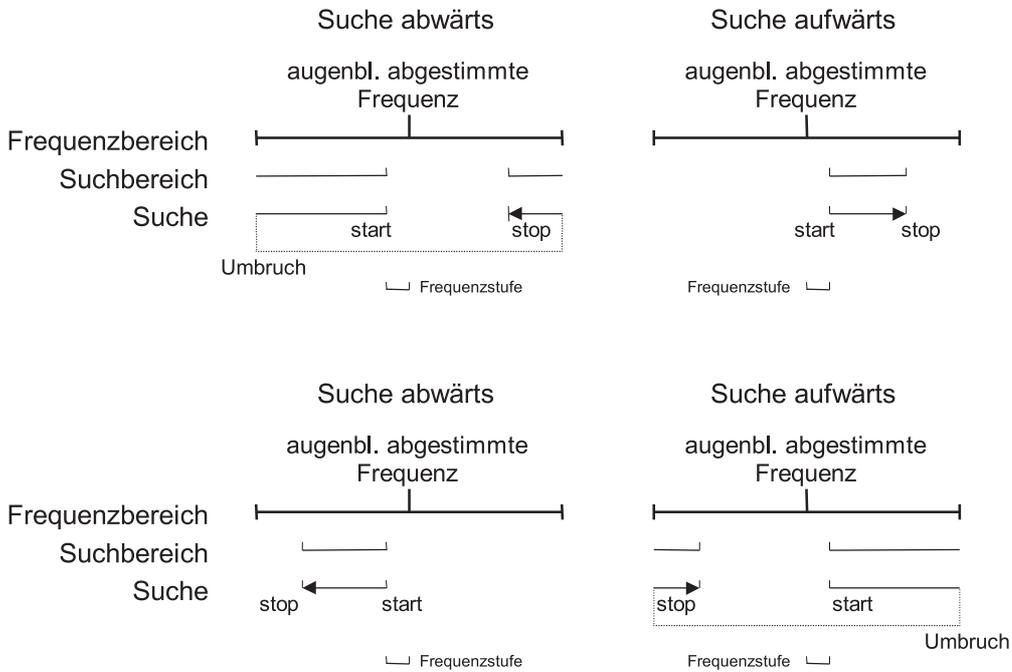


3) Stoppfrequenz ist definiert, Startfrequenz ist auf 0 gesetzt

Der Controller fordert den Empfänger auf, Suchlauf auf- oder abwärts von einer Frequenzstufe nach oder vor der augenblicklich abgestimmten Frequenz zu beginnen, Umbruch an den Grenzen der Frequenztafel oder des Frequenzbandes und Fortführen des Suchlaufs, bis die Stoppfrequenz erreicht ist.

Dieser Fall wird empfohlen, wenn die Suche bei einer Frequenzstufe nach oder vor der augenblicklich abgestimmten Frequenz beginnen muss.

start_freq=0, stop_freq

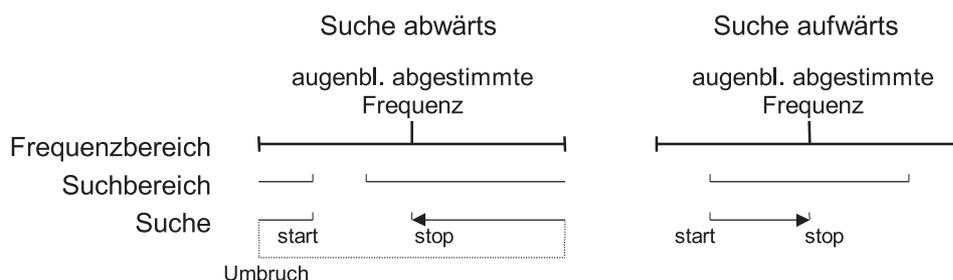


4) Start- und Stoppfrequenz sind definiert

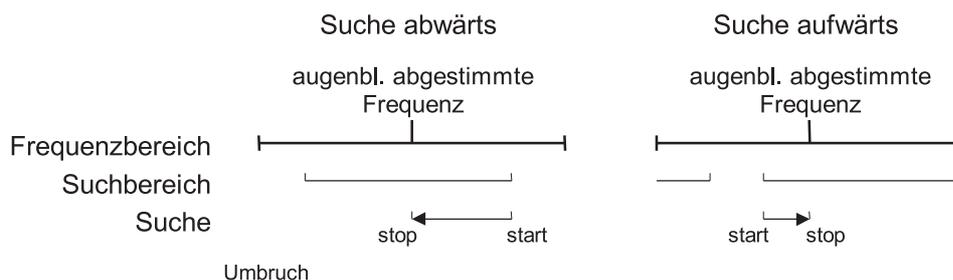
Der Controller fordert den Empfänger auf, Suchlauf auf- oder abwärts zwischen den definierten Frequenzen durchzuführen. Wenn der Empfänger den Anfang oder das Ende des angegebenen Frequenzbereiches (Tabelle oder Band) erreicht, dann sollte er umbrechen und den Suchlauf fortsetzen. Start_frequency und stop_frequency müssen innerhalb desselben Bandes liegen.

Dieser Fall wird empfohlen, wenn die Suche, abhängig von der Suchrichtung (auf-/abwärts), bei einer definierten Startfrequenz beginnen und bei einer definierten Stoppfrequenz enden muss.

start_freq < stop_freq



start_freq > stop_freq



In allen Fällen hängt der Suchbereich von der Suchrichtung ab. Der Suchlauf endet, wenn ein Ensemble gefunden wurde oder wenn das Ende des Suchbereiches, wie beschrieben, erreicht wurde (continuous_search = 0).

Wenn start_freq und stop_freq identisch sind, aber ungleich „0“, dann stimmt der Empfänger nur auf diese Frequenz ab.

Wenn es keinen freq_table-Eintrag zwischen start_freq und stop_freq gibt, wird der Befehl zurückgewiesen.

continuous_search (kontinuierliche Suche)

Dieses Flag muss angeben, ob die Suche nach einem Ensemble am Ende des angegebenen Suchbereiches fortgeführt wird oder nicht.

b0 = 0: Der Empfänger beendet den Suchlauf, wenn ein Ensemble gefunden wurde oder wenn das Ende des angegebenen Suchbereiches erreicht ist. In beiden Fällen sendet der Empfänger einen notify_dab_status.

b0 = 1: Der Empfänger setzt den Suchlauf fort, auch dann, wenn das Ende des Suchbereiches erreicht ist. Der Empfänger sendet immer ein notify_search_for_ensemble, wenn das Ende des Suchbereiches erreicht ist. Der Empfänger beendet die Suche, wenn ein Ensemble gefunden wurde und sendet ein notify_dab_status.

up/down (auf/ab)

Dieses Flag muss die Richtung des Suchlaufs angeben. Aufgrund der Werte der Start- und Stopffrequenz gibt es vier mögliche Fälle, wie die Suche aufwärts und abwärts durchgeführt wird. Siehe Beschreibung in start/stop_freq.

b0 = 0: Suche abwärts

b0 = 1: Suche aufwärts

intermediate_notify (Zwischen-Meldung)

Dieses Flag gibt die Art der Meldung an.

b0 = 0: Der Empfänger sendet keine Zwischen-Meldungen (notify_search_for_ensemble) nach jeder Frequenzstufe.

b0 = 1: Der Empfänger sendet Zwischen-Meldungen (notify_search_for_ensemble) nach jeder Frequenzstufe.

transmission_mode (Übertragungsmode)

Dieses 3-Bit-Feld zeigt den transmission_mode des DAB-Ensembles an.

b₂ ... b₀	
000	kein Übertragungsmode signalisiert
001	Übertragungsmode I
010	Übertragungsmode II
011	Übertragungsmode III
100	Übertragungsmode IV
101	Rfd
110	Rfd
111	Rfd

freq_grid/table (Frequenz-Raster/Tabelle)

Dieses als vorzeichenlose Binärzahl codierte 8-Bit-Feld gibt die Art der Suche an. Die Suche kann durch Angeben einer Start- und einer Stopffrequenz innerhalb eines 16-kHz-Rasters, eines Frequenzbandes oder durch Angabe einer Frequenztafel durchgeführt werden.

b₇ ... b₀	
0000 0000	16-kHz-Stufe, [1]
0000 0001	CEPT-Frequenztafel Band III, [6]
0000 0010	CEPT-Frequenztafel L-Band, [6]
0000 0011	Canada-Frequenztafel L-Band, [6]
0000 0100 bis 1111 1111	Rfd

Reaktion:

Antwort(accepted)

notify_search_for_ensemble(...)

Der Empfänger teilt den augenblicklichen Übertragungsmode und die Frequenz während der Ausführung des Befehls search_for_ensemble nach jeder Frequenzstufe mit (wenn das Zwischen-Meldungs-Flag auf „0“ gesetzt war). Diese Meldung wird nicht nach der letzten Frequenzstufe gesendet. In diesem Fall wird ein notify_dab_status gesendet.

notify_dab_status(...)

Der Empfänger teilt den Status nach der Ausführung des Befehls search_for_ensemble mit (wenn ein Ensemble gefunden wurde oder zusätzlich wenn das Ende des angegebenen Suchbereiches erreicht ist, wenn das continuous_search-Bit auf „0“ gesetzt war).

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.19 notify_search_for_ensemble (Ensemblesuche-Meldung)

Kategorie:

Meldung (obligatorisch)

Zweck:

Der Empfänger meldet den Status der zwischenzeitlichen Ausführung des Befehls `search_for_ensemble`.

Syntax:

`notify_search_for_ensemble` (Rfa, transmission_mode, tune_freq)

Parameter:

2 Bits	3 Bits	19 Bits
b ₁ b ₀	b ₂ b ₀	b ₁₈ b ₀
Rfa	transmission_mode	tune_freq

transmission_mode (Übertragungsmodus)

Dieses 3-Bit-Feld zeigt den `transmission_mode` der augenblicklich abgestimmten Frequenz an.

b ₂ ... b ₀	
000	bei der/m gegebenen Frequenz/Frequenzbereich wurde kein DAB-Ensemble mit dem gegebenen Übertragungsmodus gefunden
001	Übertragungsmodus I
010	Übertragungsmodus II
011	Übertragungsmodus III
100	Übertragungsmodus IV
101	Rfd
110	Rfd
111	Rfd

tune_freq (abgestimmte Frequenz)

Dieses 19-Bit-Feld zeigt die augenblicklich abgestimmte Frequenz an. Die Codierung ist in [1] angegeben.

Wenn kein Ensemble gefunden wurde, definiert die abgestimmte Frequenz das Ende des Suchbereiches (siehe Befehl `search_for_ensemble`, Parameter `start/stop_freq`).

4.20 set_drc (Dynamikbereichsregelung einschalten)

Kategorie:

Befehl (wahlfrei)

Zweck:

Der Controller schaltet die entsprechende DRC für den gewählten Tonausgang ein oder aus.

Syntax:

`set_drc` (output_interface_reference, protocol, switch_drc, sub_ch_id)

1 Byte	1 Byte	2 Bits	6 Bits
b ₇ b ₀	b ₇ b ₀	b ₁ b ₀	b ₅ b ₀
output_interface_reference	protocol	switch_drc	sub_ch_id

Parameter:

output_interface_reference (Ausgangs-Schnittstellen-Referenz)

Dieses (von den Empfängermerkmalen abgeleitete) 8-Bit-Feld muss die Ausgangs-Schnittstelle angeben, an der die Dynamikbereichsregelung durchgeführt werden muss.

protocol (Protokoll)

Dieses vom Schnittstellen-Deskriptor der Empfängermerkmale-Meldung abgeleitete 8-Bit-Feld muss das Protokoll angeben, das an der speziellen Ausgangs-Schnittstelle benutzt wird.

switch_drc (Schalter Dynamikregelung)

Dieses 2-Bit-Feld zeigt die Anwendung der DRC auf eine output_interface_reference an.

b ₁	b ₀	
0	0	DRC aus (alle)
0	1	DRC ein (benutzt F-PAD, siehe [1])
1	0	DRC ein (vom Empfänger erzeugt)
1	1	Rfd

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den Subkanal angeben, in dem die Audio-Dienstkomponenten übertragen werden (Subkanal-Identifikation, siehe [1]).

Reaktion:

Antwort(accepted)

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not_implemented)

Antwort(syntax_error)

4.21 get_audio_info (Audio-Information angeben)

Kategorie:

Befehl (wahlfrei)

Zweck:

Der Controller fordert den DAB-Empfänger auf, Information über einen Audio-Subkanal zu liefern, der vom MPEG-Audio-Header abgeleitet wurde. Wenn er diese Information nicht liefern kann, muss er diesen Befehl zurückweisen und notify_error_message senden (z. B. weil unzureichende Verarbeitungs-/Decodierungs-Leistung oder weil der Subkanal nicht geeignet ist).

Syntax:

get_audio_info (Rfa, auto_notify_enable, sub_ch_id)

1 Bit	1 Bit	6 Bits
b ₀	b ₀	b ₅ b ₀
Rfa	auto_notify_enable	sub_ch_id

Parameter:

auto_notify_enable (automatische Meldung eingeschaltet)

Dieses 1-Bit-Feld gibt an, wann die notify_audio_info gesendet werden sollte.

b₀	
0	nur eine Meldung senden
1	eine Meldung senden und immer, wenn eine Änderung in der notify_audio_info eintritt (ausgenommen Änderungen im Rfa-Feld und ausgenommen Übergänge zwischen <i>stereo</i> und <i>joint stereo</i>)

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den Subkanal angeben, in dem die Audio-Dienstkomponente übertragen wird (Subkanal-Identifikation, siehe [1]).

Reaktion:

Antwort(accepted)

notify_audio_info(...)

Empfänger stellt die geforderte Information zur Verfügung

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(not implemented)

Antwort(syntax_error)

4.22 notify_audio_info (Audio-Information-Meldung)

Kategorie:

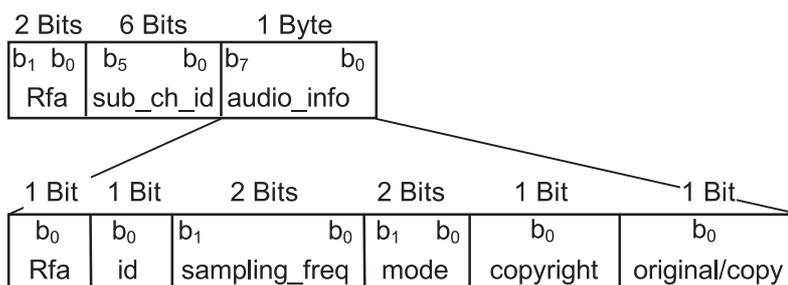
Meldung (wahlfrei)

Zweck:

Der DAB-Empfänger liefert die angeforderte Audio-Information als Reaktion auf den Befehl get_audio_info(...).

Syntax:

notify_audio_info (Rfa, sub_ch_id, audio_info)



Parameter:

sub_ch_id (Subkanal-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss den Subkanal angeben, in dem die Audio-Dienstkomponente übertragen wird (Subkanal-Identifikation, siehe [1]).

audio_info (Audio-Information)

Dieses 8-Bit-Feld liefert die Information über den gewählten Audio-Subkanal. Für die Definition von *id*, *sampling_frequency*, *mode*, *copyright* und *original/copy* siehe [1].

4.23 *get_dab_status* (DAB-Status angeben)

Kategorie:

Befehl (obligatorisch)

Zweck:

Der Controller fordert den DAB-Empfänger auf, den augenblicklichen Status des Empfängers und die die Decodierung des DAB-Ensembles betreffende Leistungsfähigkeits-Indikatoren anzugeben.

Syntax:

`get_dab_status ()`

Parameter:

keine

Reaktion:

Antwort(accepted)

`notify_dab_status(...)`

Der Empfänger stellt seinen augenblicklichen Status und seine Leistungsfähigkeits-Indikatoren zur Verfügung.

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.24 *notify_dab_status* (DAB-Status-Meldung)

Kategorie:

Meldung (obligatorisch)

Zweck:

Der Empfänger stellt seinen augenblicklichen Status und seine Leistungsfähigkeits-Indikatoren zur Verfügung.

Syntax:

`notify_DAB_status (notify_reason, search, transmission_mode, ber_fic, mute, tune_freq, reconf, sync)`

4 Bits		1 Bit	3 Bits		3 Bits		2 Bits		19 Bits		4 Bits		4 Bits	
b ₃	b ₀	b ₀	b ₂	b ₀	b ₂	b ₀	b ₁	b ₀	b ₁₈	b ₀	b ₃	b ₀	b ₃	b ₀
notify_reason	search	transmission_mode	ber_fic	mute	tune_freq	reconf	sync							

Parameter:

notify_reason (Angabe des Grundes)

Dieses als vorzeichenlose Binärzahl codierte 4-Bit-Feld zeigt den Grund für die Meldung an.

Die Gründe `get_dab_status` und `tune` zeigen auf die entsprechenden Befehle eine direkte Meldung an.

Die Gründe `sync`, `reconf`, `ber_fic` und `mute` zeigen eine Meldung an, die gesendet wurde, weil eine durch `set_dab_status_auto_notification(...)` gestellte Bedingung wahr wurde.

b₃ ... b₀	
0000	get_dab_status
0001	tune
0010	search_for_ensemble
0011	Rfd
0100	sync
0101	reconf
0110	ber_fic
0111	mute
1000 bis 1111	Rfd

search (Suche)

Dieses Flag zeigt eine laufende Suche an. Es wird gesetzt, wenn der Empfänger im Augenblick nach einem Ensemble sucht, das den Befehl search_for_ensemble(...) ausführt.

b0 = 0: keine laufende Suche

b0 = 1: laufende Suche

transmission_mode (Übertragungsmodus)

Dieses als vorzeichenlose Binärzahl codierte 3-Bit-Feld zeigt den DAB-Übertragungsmodus des Ensembles bei der augenblicklich abgestimmten Frequenz an.

b₂ ... b₀	
000	kein Übertragungsmodus, d. h. nicht auf ein Ensemble abgestimmt
001	Übertragungsmodus I
010	Übertragungsmodus II
011	Übertragungsmodus III
100	Übertragungsmodus IV
101	Rfd
110	Rfd
111	Rfd

ber_fic (Bitfehlerrate des Schnell-Informationskanals)

Diese vorzeichenlose 3-Bit-Zahl beschreibt die augenblickliche FIC-Kanal-Bitfehlerrate BER (vor dem Viterbi-Decoder) wie folgt:

b₂ ... b₀	BER-Schwelle
000	Rfd
001	Rfd
010	BER-Schwelle 1 (BER < 5e-4)
011	BER-Schwelle 2 (BER < 5e-3)
100	BER-Schwelle 3 (BER < 5e-2)
101	BER-Schwelle 4 (BER < 1e-1)
110	BER-Schwelle 5 (BER ≥ 1e-1)
111	keine BER-Schwelle verfügbar

Dieser Wert muss ein Mittelwert sein, der von den FIBs ermittelt wurde, die mit dem neuesten Block von 16 Perioden zu 24 ms (= 384 ms) verbunden sind.

mute (Stummschalten)

Diese vorzeichenlose 2-Bit-Zahl muss den augenblicklichen Stummschaltungspegel eines vorhandenen Tonquellendecoders anzeigen.

b₁ ... b₀	
00	keine Audiorahmen stummgeschaltet, Stummschaltungspegel 0
01	einige Audiorahmen stummgeschaltet, Stummschaltungspegel 1
10	dauernde Stummschaltung, Stummschaltungspegel 2
11	keine Information über Stummschaltungspegel verfügbar

Dieser Wert muss von den Audiorahmen ermittelt werden, die mit dem neuesten Block von 16 Perioden zu 24 ms (= 384 ms) verbunden sind.

tune_freq (abgestimmte Frequenz)

Diese vorzeichenlose 19-Bit-Binärzahl zeigt die augenblicklich abgestimmte Frequenz an wie sie in [1] definiert wird. Ausgenommen davon bedeutet *tune_freq* = 0, dass der Empfänger in einem unbelegten Zustand ist (auf keine Frequenz abgestimmt).

reconf (Rekonfiguration)

Dieses 4-Bit-Feld zeigt an, wenn es eine Dienst- oder Subkanal-Rekonfiguration gibt, die in der Multiplex-Rekonfigurationsinformation (MCI) signalisiert wird.

Bit-Flag	
b ₀	Subkanal-Rekonfiguration b ₀ = 0: Subkanal-Rekonfiguration nicht erwartet b ₀ = 1: Subkanal-Rekonfiguration erwartet
b ₁	Dienst-Rekonfiguration b ₁ = 0: Dienst-Rekonfiguration nicht erwartet b ₁ = 1: Dienst-Rekonfiguration erwartet
b ₂	Rfa
b ₃	Rfa

sync (Synchronisation)

Dieses 4-Bit-Feld zeigt an, inwieweit der Empfänger auf das DAB-Ensemble synchronisiert ist.

Bit-Flag	
b0	DAB-Signal erkannt Der Empfänger hat ein DAB-Signal erkannt, weil er den Übertragungsmodus erkennen konnte. b0 = 0: kein DAB-Signal erkannt b0 = 1: DAB-Signal erkannt
b1	DAB-Empfänger ist vollständig synchronisiert Der Empfänger beendet Zeit- und Frequenz-Synchronisierung, sodass er physikalisch auf das Signal synchronisiert ist. b1 = 0: nicht vollständig synchronisiert b1 = 1: vollständig synchronisiert
b2	Fähigkeit zur MCI-Decodierung Der Empfänger kann FIC decodieren (d. h. die FIC-Bitfehlerraten sind niedrig genug), um Information über das DAB-Multiplex zu erhalten. b2 = 0: kann die MCI nicht decodieren b2 = 1: kann die MCI decodieren
b3	Rfa

4.25 *set_dab_status_auto_notification* (autom. Meldung DAB-Status einschalten)

Kategorie:

Befehl (obligatorisch)

Zweck:

Der Controller fordert laufende Meldung des DAB-Empfängerstatus unter bestimmten Bedingungen. Jedes Mal wenn eine bestimmte Bedingung wahr wird, muss der Empfänger eine Meldung `notify_dab_status` senden.

Dieser Befehl wird auch benutzt, um vorher aktivierte automatische Meldungen abzuschalten.

Vor dem Senden der Befehle `tune(...)` oder `search_for_ensemble(...)` sollten automatische Meldungen durch den Controller ausgeschaltet werden, um den Empfang von zu vielen Meldungen zu vermeiden.

Syntax:

`set_dab_status_auto_notification (sync, reconf, Rfa, mute, ber_fic)`

4 Bits		4 Bits		3 Bits		2 Bits		3 Bits	
b ₃	b ₀	b ₃	b ₀	b ₂	b ₀	b ₁	b ₀	b ₂	b ₀
sync		reconf		Rfa		mute		ber_fic	

Parameter:

sync (Synchronisation)

Dieses 4-Bit-Feld muss anzeigen, bei welchen Änderungen der Synchronisation der Empfänger eine Meldung senden muss.

Bit-Flag	
b0	Ausfall oder Erkennen des DAB-Signals Der DAB-Empfänger hat ein DAB-Signal verloren oder erkannt.
b1	Änderung des Zustandes „voll synchronisiert“ Der Empfänger verliert oder erreicht Zeit- und Frequenz-Synchronisation.
b2	Änderung der Fähigkeit, die MCI zu decodieren Der DAB-Empfänger verliert oder erreicht die Fähigkeit, die MCI zu decodieren (d. h. die FIC-Bitfehlerrate wird hinreichend niedrig oder zu hoch).
b3	Rfa

Die Flags b_i ($i = 0$ bis 3) müssen wie folgt codiert werden.

0: keine Meldung über Ausfall, Erkennen oder Ändern senden

1: Meldung über Ausfall, Erkennen oder Ändern senden

reconf (Rekonfiguration)

Dieses 4-Bit-Feld muss anzeigen, in welcher Art von Multiplex-Rekonfiguration der Empfänger eine Meldung senden muss.

Bit-Flag	
b0	Subkanal-Rekonfiguration erwartet
b1	Dienst-Rekonfiguration erwartet
b2	Rfa
b3	Rfa

Die Flags b_i ($i = 0,1$) müssen wie folgt codiert werden.

0: kein Meldung über erwartete Rekonfiguration senden

1: Meldung über erwartete Rekonfiguration senden

mute (Stummschalten)

Dieses 2-Bit-Feld ist eine vorzeichenlose Zahl, die die Bedingungen bezüglich des Stummschaltungspegels beschreibt, wie sie für die Meldung `notify_dab_status(...)` signalisiert werden.

$b_1 \dots b_0$	
00	keine Meldungen über Änderung des Stummschaltungspegels
01	Meldung senden, wenn es eine Änderung von oder zu dem Stummschaltungspegel 0 gibt
10	Rfd
11	Rfd

ber_fic (Schnell-Informationskanal-Bitfehlerrate)

Diese vorzeichenlose 3-Bit-Zahl muss eine Schwelle für den FIC-Bitfehlerraten-Pegel beschreiben. Jedes Mal wenn die BER, wie für die Meldung notify_dab_status(...) signalisiert, sich von „unter der Schwelle“ zu „über der Schwelle“ oder umgekehrt ändert, muss der Empfänger eine Meldung senden.

b₂ ... b₀	
000	keine Meldung über Änderung der FIC-BER-Schwelle
001	Rfd
010	BER-Schwelle 1 = 5e-4
011	BER-Schwelle 2 = 5e-3
100	BER-Schwelle 3 = 5e-2
101	BER-Schwelle 4 = 1e-2
110	Rfd
111	Rfd

Reaktion:

Antwort(accepted)

notify_dab_status(...)

Der Empfänger stellt seinen augenblicklichen Status einschließlich der Gründe für die Meldung zur Verfügung.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.26 get_active_info (Aktivinformation angeben)

Kategorie:

Befehl (obligatorisch)

Zweck:

Der Controller fordert den DAB-Empfänger auf zu melden, ob er noch aktiv ist. In diesem Fall sendet der Empfänger eine Meldung mit einer Wiederholrate von 3,5 Sekunden. Wenn der Controller den Empfänger einmal aufgefordert hat, erhält der Controller diese Meldungen, bis er sie beendet.

Syntax:

get_active_info (Rfa, select)

Parameter:

7 Bits	1 Bit
b ₆ b ₀	b ₀
Rfa	select

select

Dieses 1-Bit-Feld gibt an, ob die kontinuierliche Ausgabe begonnen oder beenden werden muss.

b ₀	select
0	kontinuierliche Ausgabe beenden
1	kontinuierliche Ausgabe beginnen

Reaktion:

Antwort(accepted)

notify_active_info(...)

Der Empfänger gibt eine aktive Anzeige.

Antwort(rejected)

notify_error_message(...)

Antwort(interim)

Antwort(busy)

Antwort(syntax_error)

4.27 notify_active_info (Aktivinformation-Meldung)

Kategorie:

Meldung (obligatorisch)

Zweck:

Der Empfänger gibt wiederholend eine aktive Anzeige, wenn dies durch einen Befehl get_active_info angefordert wurde. Sie wird alle 3 bis 5 Sekunden gegeben.

Syntax:

notify_active_info ()

Parameter:

keine

4.28 notify_service_following (service-following-Meldung)

Kategorie:

Meldung (wahlfrei)

Zweck:

Der Empfänger gibt die Information, dass der nahtlose Empfang mit anderem DAB-Ensemble oder anderem Rundfunksystem erfolgt. Diese Meldung wird nur gesendet, wenn die *service-following*-Merkmale von dem Empfänger unterstützt und in den Empfängermerkmalen signalisiert werden.

Syntax:

notify_service_following (Rfa, system, Rfa, p/d, transmission_mode, tune_freq, new_sid)

4 Bits		4 Bits		1 Bit	1 Bit	3 Bits			19 Bits		4 Bytes		
b ₃	b ₀	b ₃	b ₀	b ₀	b ₀	b ₂		b ₀	b ₁₈		b ₀	b ₃₁	b ₀
Rfa	system	Rfa	p/d			transmission_mode			tune_freq			new_sid	

Parameter:

system (System)

Dieses 4-Bit-Feld gibt das Rundfunksystem an, zu dem nahtloser Empfang durchgeführt wird.

b₃ ... b₀	
0000	DAB
0001	FM
0010	AM
0011..1111	Rfd

Nur wenn nahtloser Empfang nach DAB durchgeführt wird, sind die nachstehenden Parameter gültig, andernfalls müssen sie auf Null gesetzt werden.

p/d (Programm/Daten)

siehe select_channel

transmission_mode (Übertragungsmode)

siehe notify_dab_status

tune_freq (abgestimmte Frequenz)

siehe notify_dab_status

new_sid (neue Dienst-Identifikation)

Dieses als vorzeichenlose Binärzahl codierte 32-Bit-Feld signalisiert die neue Dienst-Identifikation des Dienstes, dem der Empfänger gefolgt war.

siehe sid bei select_channel

4.29 **manufacturer_specific_command** (herstellerspezifischer Befehl)

Kategorie:

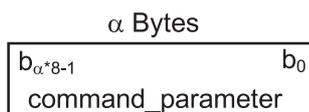
Befehl (wahlfrei)

Zweck:

Ein herstellerspezifischer Befehl wird vom Controller zu einem Empfänger gesendet.

Syntax:

manufacturer_specific_command (command_parameter)



Parameter:

command_parameter (Befehlsparameter)

Dieses α -Bytes-Feld muss den herstellerspezifischen Befehl enthalten.

Reaktion:

Antwort(accepted)

manufacturer_specific_notification(...)

Antwort(rejected)

manufacturer_specific_notification(...)

Antwort(interim)

Antwort(busy)

Antwort(not_implemented)

Antwort(syntax_error)

4.30 *manufacturer_specific_notification* (herstellerspezifische Meldung)

Kategorie:

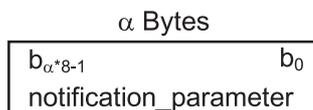
Meldung (wahlfrei)

Zweck:

Eine herstellerspezifische Meldung wird benutzt, um Daten vom Empfänger an den Controller zu senden.

Syntax:

manufacturer_specific_notification (*notification_parameter*)



Parameter:

notification_parameter (Meldungs-Parameter)

Dieses α-Byte-Feld muss die herstellerspezifische Meldung enthalten.

4.31 *notify_error_message* (Fehlermeldung-Meldung)

Kategorie:

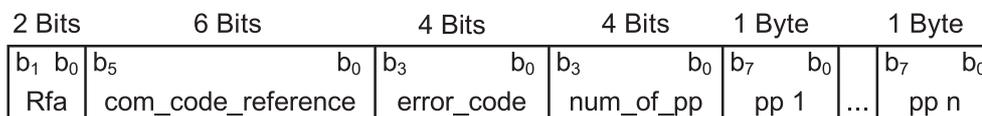
Meldung (obligatorisch)

Zweck:

Der DAB-Empfänger teilt dem Controller die Gründe mit, weshalb er die Anforderungen nicht ausführen kann, nachdem er eine Antwort 'zurückgewiesen' gesendet hat.

Syntax:

notify_error_message (*Rfa*, *com_code_reference*, *error_code*, *num_of_pp*, *pp 1*,...*pp n*)



Parameter:

com_code_reference (Befehlscode-Referenz)

Dieses als vorzeichenlose Binärzahl codierte 6-Bit-Feld muss die Befehlscode-Referenz angeben, mit der die Fehlermeldung verbunden ist.

error_code (Fehlermeldung)

Dieses 4-Bit-Feld beschreibt die Fehler, die die Ausführung der entsprechenden Befehle teilweise oder vollständig unmöglich macht.

b₃ ... b₀	
0000	Rfd
0001	Parameterwert nicht unterstützt (Begrenzung des Empfängers)
0010	Parameterwert nicht verfügbar (im Rundfunksignal)
0011	Parameterwert an der Schnittstelle nicht verfügbar
0100	Parameterwert ungültig (nicht mit [1] konform)
0101	ungenügende Schnittstellenkapazität
0110	Busmode unterstützt nicht diesen Befehl
0111	ungenügende Verarbeitungs-/Decodierungs-Leistung
1000	Schnittstelle nicht geeignet
1001	Subkanal nicht geeignet
1010 bis 1111	Rfd

num_of_pp (Anzahl der Parameterzeiger)

Dieses als vorzeichenlose Zahl codierte 4-Bit-Feld muss die Anzahl *n* der folgenden Parameterzeiger im Fall von error_code 0001..0100 anzeigen, in allen anderen Fällen muss dieses Feld auf Null gesetzt werden.

pp x ($0 \leq x \leq n$) (Parameterzeiger)

Dieses als vorzeichenlose Zahl codierte 8-Bit-Feld muss die Parameterzeiger angeben. Diese Zeiger zeigen auf die falschen oder nicht unterstützten Parameter in dem entsprechenden Befehl (die Rfa-Felder werden auch berücksichtigt). Der Wert „0“ bezieht sich auf den ersten Parameter.

5 Beispiel von DCSR-Codierung

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung
1	get_receiver_capability 0x 41		der Controller fordert die Angabe der Empfängermerkmale
2		accepted 0x 81	der Empfänger akzeptiert das Ausführen des Befehls get_receiver_capabilities
3		notify_receiver_capability 0x C1	der Empfänger informiert den Controller über seine Merkmale
		dcsr_id: 0000 0001 dcsr_profile: 0000 0000 0000 0000 0000 0000 0110 0011 manufacturer_id 0101 1000 0101 1001 model_number 0000 0000 0000 0111 model_year 0101 0111 serial_number 0000 0000 0000 0000 0010 1010 freq_range: freq_table 0000 0000 0000 0000 0000 0000 0000 0111 freq_band 0000 0000 0000 0011 freq_band_flex: num_freq_flex 0000 0001 start_freq 1 000 0001 0101 0011 1110 end_freq 1 000 0001 1010 0101 1110 padding 00 transmission_mode 1000 1111 max_net_bitrate 0011 0000 max_cu_num 00 1100 0000 num_subch 00 0010 tii 0001 0001 sf 0001	DCSR-Version 1 wahlfreie Befehle implementiert: get_tii, get_pad set_drc, get_audio_info CRIN-Hersteller-ID = „XY“ CRIN-Modellnummer = 7 CRIN-Modelljahr = „W“ (= 1998) CRIN-Seriennummer = 42 unterstützte Frequenztabellen: 16-kHz-Stufen CEPT-Tabelle für Band III CEPT-Tabelle für L-Band unterstützte Bänder: Band III (174 MHz bis 240 MHz) L-Band (1 452 MHz bis 1 492 MHz) ein zusätzliches Band unterstützt von 87,008 MHz bis 108,000 MHz unterstützt Modes I, II, III, IV und automatische Mode-Erkennung maximale Netto-Bitrate = 384 kbit/s maximale Anzahl von CUs = 192 kann zwei Subkanäle gleichzeitig decodieren kann zur Verfügung stellen Haupt- und Sub-Identifikation Grobe Position unterstützt 24-kHz-Abtastfrequenz

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung
		drc 0001	unterstützt DRC unter Verwendung von F-PAD
		acs 0000 0000	unterstützt kein Zugangssystem
		pad 0001	kann PAD des gewählten Audio-Subkanals zur Verfügung stellen, die zu dem eingebauten Audiodecoder gesendet werden
		aic 1	kann FIGs von dem AIC zur Verfügung stellen
		audio_info 0001	kann die audio_info nur des gewählten Audio-Subkanals zur Verfügung stellen
		aud_dec 1	es ist mindestens ein Audiodecoder vorhanden
		vid_dec 0	es ist kein Videoquellendecoder vorhanden
		fig_filtering 10	kann Basis-FIG-Filterung durch Typ/Erweiterung und FIG-Datenfeld-Headerbits durchführen
		Rfa 0000	
		service_following 0000 0000 0000 0001	kann nahtlosen Empfang zu DAB-Ensembles durchführen (keine lokalen Fenster)
		interface_descriptor num_io 0000 0101	fünf Schnittstellen input_interface_reference = 1
		interface 1 in/out 0	ist eine Ausgangs-Schnittstelle
		I/O_id 000 1100	RS 232
		Rfa 0000	
		num_protocols 0001	ein Protokoll
		protocol 1 0000 0111	DCSR-Protokoll
		interface 2 in/out 1	output_interface_reference = 2 ist eine Eingangs-Schnittstelle
		I/O_id 000 1100	RS 232
		Rfa 0000	
		num_protocols 0001	ein Protokoll
		protocol 1 0000 0111	DCSR-Protokoll

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung
	transmission_mode 001 freq_grid/table 0000 0001		DAB-Ensemble mit Übertragungsmodus I suchen bei den in der CEPT-Frequenztabelle von Band III definierten Frequenzen suchen
5		interim 0x 83	aus irgendeinem Grund kann der Empfänger nicht beginnen, den Befehl search_for_ensemble innerhalb von 100 ms auszuführen
6		accepted 0x 81	der Empfänger akzeptiert, den Befehl search_for_ensemble auszuführen
7		notify_dab_status 0x CF	der Empfänger informiert über das Ergebnis des Ausführens des Befehls search_for_ensemble
		notify_reason 0010 search 0 transmission_mode 001 ber_fic 010 mute 00 tune_freq 000 0011 0111 0001 0111 reconf 0000 sync 0100	der Befehl search_for_ensemble verursachte diese Meldung keine laufende Suche auf der augenblicklich abgestimmten Frequenz gibt es ein DAB-Ensemble mit Übertragungsmodus I BER-Schwelle 1 (BER < 5e-4) es werden keine Audiorahmen stummgeschaltet Stummschaltungspegel 0 die augenblicklich abgestimmte Frequenz ist 225,648 MHz (Kanal 12 B) keine Subkanal- oder Dienst-Rekonfiguration erwartet DAB-Empfänger kann FIC/MCI decodieren
8	get_figs 0x 47		der Controller fordert den Empfänger auf, FIGs zu senden
	Rfa 0 control 010 Rfa 0 type 000 fig_filters 0000 0000		alle verfügbaren FIGs sind angefordert in Verbindung mit dem gewählten Wert des Steuerparameters ohne Bedeutung in Verbindung mit dem gewählten Wert des Steuerparameters ohne Bedeutung

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung
	extensions 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000		in Verbindung mit dem gewählten Wert des Steuerparameters ohne Bedeutung
9		accepted 0x 81	der Empfänger akzeptiert, den Befehl get_figs auszuführen
10		notify_fig 0x C7	Empfänger sendet empfangenen/s FIG-Header und FIG-Datenfeld (FIG aus FIB ohne CRC)
		0x 35 01 D3 01 20 20 20 20 20 53 44 52 20 31 20 20 20 20 20 1F 70	eine FIG (Typ 1/Erweiterung 1)
11		notify_fig 0x C7	Empfänger sendet empfangenen/s FIG-Header und FIG-Datenfeld (FIG aus FIB ohne CRC)
		0x 05 11 D3 05 00 0A	eine FIG (Typ 0/Erweiterung 17)
12		notify_fig 0x C7	Empfänger sendet empfangenen/s FIG-Header und FIG-Datenfeld (FIG aus FIB ohne CRC)
		0x 35 01 D3 03 20 20 20 20 20 53 44 52 20 33 20 20 20 20 20 1F 70	eine FIG (Typ 1/Erweiterung 1)
13	select_channel 0x 4A		der Controller fordert den Empfänger auf, einen speziellen Subkanal auszuwählen
	Input_interface_reference 0000 0011 output_interface_reference 0000 0101 protocol 0000 0010 subfunction 010 Rfa 0 m/f 0 auto_service_following 1 pds 1 p/d 0 sid 0000 0000 0000 0000 1101 0011 0000 0001 Rfa 00 sub_ch_id 00 1010		input_interface_reference = 3 (RF-Eingang) output_interface_reference = 5 (IEC 60958) RDI-Mode hoher Kapazität normaler Rahmen die angegebene Auswahl zum angegebenen Ausgang hinzufügen der FIC sollte nicht ausgegeben werden, jedoch der angegebene Subkanal der Empfänger sollte nahtlosen Empfang durchführen die Parameter „p/d“ und „sid“ sind gültig 16-Bit-Sid Dienst-Identifikation: 0x 0000 D301 Subkanal-Identifikation: 10

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung
14		rejected 0x 82	der Empfänger führt den Befehl select_channel nicht aus. Der Grund wird in notify_error_message angegeben
15		notify_error_message 0x F0	der Empfänger informiert, warum der Befehl select_channel zurückgewiesen wurde
		Rfa com_code_reference error_code num_of_p9 pp 1	 00 00 1010 Befehlscode Reference = 0x0A (select channel) 0010 Fehlercode = 2 (Parameterwert im Sendesignal nicht verfügbar) 0001 ein Parameter 0000 1011 Parameter 11 (zählt von 0 bis 11) (es ist nicht möglich, den Subkanal 10 zu wählen, weil dieser Subkanal im Sendesignal nicht verfügbar ist)
16	select_channel 0x 4A		der Controller fordert den Empfänger auf, einen speziellen Subkanal zu wählen
	Input_interface_reference 0000 0011 output_interface_reference 0000 0101 protocol 0000 0010 subfunction 010 Rfa 0 m/f 0 auto_service_following 1 pds 1 p/d 0 sid 0000 0000 0000 0000 1101 0011 0000 0001 Rfa 00 sub_ch_id 00 0010		input_interface_reference = 3 (RF-Eingang) output_interface_reference = 5 (IEC 958) RDI-Mode hoher Kapazität normaler Rahmen den angegebenen Subkanal zu der angegebenen Ausgangs-Schnittstelle hinzufügen der FIC sollte nicht ausgeben, jedoch der angegebene Subkanal der Empfänger sollte dem Dienst automatisch folgen die Parameter „p/d“ und „sid“ sind gültig 16-Bit-Sid Dienst-Identifikation: 0x 0000 D301 Subkanal-Identifikation: 2

Nr.	Controller sendet ... msb bis lsb	Empfänger sendet ... msb bis lsb	Beschreibung																								
17		accepted 0x 81	der Empfänger akzeptiert, den Befehl select_channel auszuführen																								
18	get_dab_status 0x 4F		der Controller fordert vom Empfänger, den aktuellen DAB-Status zu senden																								
19		interim 0x 83	aus irgendeinem Grund kann der Empfänger nicht beginnen, den Befehl search_for_ensemble innerhalb von 100 ms auszuführen																								
20		notify_fig 0x C7	Empfänger sendet empfangenen/s FIG-Header und FIG-Datenfeld (FIG aus FIB ohne CRC)																								
		0x 35 01 D3 05 20 20 20 20 44 41 53 44 49 4E 47 20 20 20 20 0F F0	eine FIG (Type 1/Erweiterung 1)																								
21		accepted 0x 81	der Empfänger akzeptiert, den Befehl get_dab_status auszuführen																								
22		notify_dab_status 0x CF	der Empfänger informiert über den aktuellen DAB-Status																								
		<table border="0"> <tr> <td>notify_reason</td> <td>0000</td> <td>der Befehl get_dab_status verursacht diese Meldung</td> </tr> <tr> <td>search</td> <td>0</td> <td>keine laufende Suche</td> </tr> <tr> <td>transmission_mode</td> <td>001</td> <td>auf der augenblicklich abgestimmten Frequenz gibt es ein DAB-Ensemble mit Übertragungsmode I</td> </tr> <tr> <td>ber_fic</td> <td>010</td> <td>BER-Pegel 1 (BER < 5e-4)</td> </tr> <tr> <td>mute</td> <td>00</td> <td>es werden keine Audiorahmen stummgeschaltet Stummschaltungspegel 0</td> </tr> <tr> <td>tune_freq</td> <td>000 0011 0111 0001 0111</td> <td>die aktuell abgestimmte Frequenz ist 225,648 MHz (Kanal 12 B)</td> </tr> <tr> <td>reconf</td> <td>0000</td> <td>keine Subkanal- oder Dienst-Rekonfiguration erwartet</td> </tr> <tr> <td>sync</td> <td>0100</td> <td>DAB-Empfänger kann die FIC/MCI decodieren</td> </tr> </table>	notify_reason	0000	der Befehl get_dab_status verursacht diese Meldung	search	0	keine laufende Suche	transmission_mode	001	auf der augenblicklich abgestimmten Frequenz gibt es ein DAB-Ensemble mit Übertragungsmode I	ber_fic	010	BER-Pegel 1 (BER < 5e-4)	mute	00	es werden keine Audiorahmen stummgeschaltet Stummschaltungspegel 0	tune_freq	000 0011 0111 0001 0111	die aktuell abgestimmte Frequenz ist 225,648 MHz (Kanal 12 B)	reconf	0000	keine Subkanal- oder Dienst-Rekonfiguration erwartet	sync	0100	DAB-Empfänger kann die FIC/MCI decodieren	
notify_reason	0000	der Befehl get_dab_status verursacht diese Meldung																									
search	0	keine laufende Suche																									
transmission_mode	001	auf der augenblicklich abgestimmten Frequenz gibt es ein DAB-Ensemble mit Übertragungsmode I																									
ber_fic	010	BER-Pegel 1 (BER < 5e-4)																									
mute	00	es werden keine Audiorahmen stummgeschaltet Stummschaltungspegel 0																									
tune_freq	000 0011 0111 0001 0111	die aktuell abgestimmte Frequenz ist 225,648 MHz (Kanal 12 B)																									
reconf	0000	keine Subkanal- oder Dienst-Rekonfiguration erwartet																									
sync	0100	DAB-Empfänger kann die FIC/MCI decodieren																									

6 Richtlinie für Datenfelder, die für zukünftige Verwendung (Rfus) oder Zusätze (Rfas) reserviert sind, und Tabelleneinträge, die für zukünftige Definitionen (Rfds) reserviert sind

Um Normen zu erarbeiten, die Erfordernissen angepasst werden können, die gegenwärtig nicht bekannt sind, aber in Zukunft auftreten können, wurden spezielle Bits als Hinweise beschrieben, die Erweiterungen und Anhänge auf rückwärts kompatible Art ermöglichen.

Rfa: „Reserviert für zukünftige Zusätze“

„Rfas“ belegen Datenfelder fester Größe. Diese Bits müssen auf „0“ gesetzt werden, bis ihnen eine Funktion zugeordnet wird.

„Rfas“ wirken selbstständig und ändern nicht die Bedeutung von anderen Teilen eines Informationsträgers (dies ist zum Beispiel ein Befehl, ein Header-Erweiterungsparameter, eine Schnell-Informationsgruppe usw.). Der Empfänger, der keine Kenntnis von der neuen Definition dieser ehemaligen „Rfa“-Bits/Felder hat, kann die verbleibenden Datenfelder des entsprechenden Informationsträgers (siehe oben) noch verwenden. Er kann nur von der neu definierten zusätzlichen Information keinen Gebrauch machen.

Rfu: „Reserviert für zukünftige Verwendung“

„Rfus“ belegen Datenfelder fester Größe auf die gleiche Art wie „Rfas“. Diese Bits müssen auf „0“ gesetzt werden, bis ihnen eine Bedeutung zugeordnet wird.

Aber sobald „Rfus“ durch eine Definition ersetzt werden, könnten sie auf bestimmte Felder des Informationsträgers wirken (dies ist zum Beispiel ein Befehl, ein Header-Erweiterungsparameter, eine Schnell-Informationsgruppe usw.); sie sind Teil davon und ändern ihre Bedeutung und Länge (und dadurch die Länge des vollständigen Informationsträgers). Die Felder, auf die die neue Funktionalität der früheren „Rfus“ wirkt, müssen schon innerhalb der Beschreibung der „Rfus“ beachtet werden. Aus gegebenen Gründen kann der vollständige Informationsträger möglicherweise durch die alten Empfänger nicht mehr bewertet werden, die keine Kenntnis von der neuen Bedeutung haben.

Rfd: „Reserviert für zukünftige Definition“

Einträge in Tabellen, die bis jetzt nicht zugewiesen wurden, sind „Rfds“. Hierdurch haben sie auch eine feste Größe. Sie werden durch die Tabellenbeschreibung vordefiniert. Üblicherweise werden auf diese Art innerhalb einer Tabelle Wertebereiche reserviert. (Beispiel: In der DCSR-Norm sind die Einträge „000“bin bis „100“bin und „111“bin für den 3 Bits breiten Tabelle/Parameter-„transmission_mode“ innerhalb des tune-Befehls schon definiert, während die Einträge „101“bin und „110“bin für zukünftige Definition reserviert sind. Sobald einer der beiden letzteren Einträge vorkommt, kann ein „alter“ Empfänger den vollständigen Befehl nicht mehr verstehen (weil er den geforderten Übertragungsmodus nicht kennt).)

Ein oder wenige dieser reservierten Einträge können auch später als Abbruch-Wahlmöglichkeiten definiert werden, die dem Empfänger mitteilen, dass er die entscheidende Information woanders findet (zum Beispiel in einer Erweiterungstabelle).

7 Abkürzungen

MFN Multi Frequency Network

USB Universal Serial Bus

IrDA Infrared Data Association

Weitere Abkürzungen siehe [1].

8 Literaturhinweise

- [1] ETS 300401:1997, *Rundfunk-Systeme – Digitaler Tonrundfunk (DAB) für mobile, tragbare und ortsfeste Empfänger.*
- [2] Eureka Project 147, Issue 2.2 (1995), *Digital Audio Broadcasting Systems; Guidelines for Implementation and Operation.*
- [3] Norwegian Telecom, Issue 2 (1989), *NR-MSK Access Control System.*
- [4] EN 50094:1992, *Zugriffskontrollsysteme für die MAC/Paket-Familie: Eurocrypt.*
- [5] ISO 10486:1992, *Passenger cars – Car radio identification number (CRIN).*
- [6] EN 50248:1997, *Eigenschaften von DVB-Empfängern.*