

	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 Deutsche Elektrotechnische Kommission im DIN und VDE (DKE)		

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 500094:1992 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.*

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> (service-following-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 ($t2 \geq 0$).

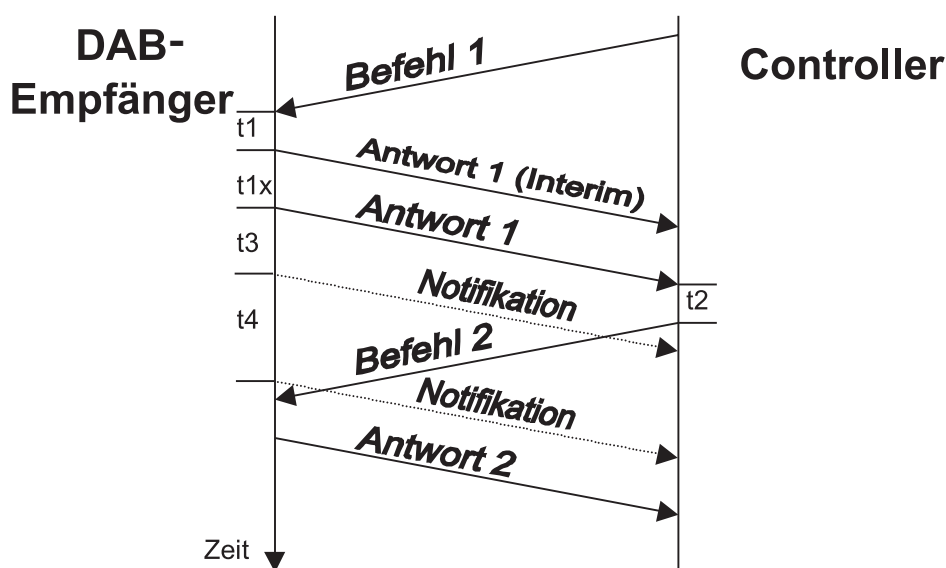


Bild 1 – Allgemeine Befehl-Antwort-Struktur

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

2 Bits	6 Bits	α Bytes
b_1 b_0 category_code	b_5 b_0 reference_code	$b_{\alpha*8-1}$ b_0 Parameter

2 Referenztable

		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:

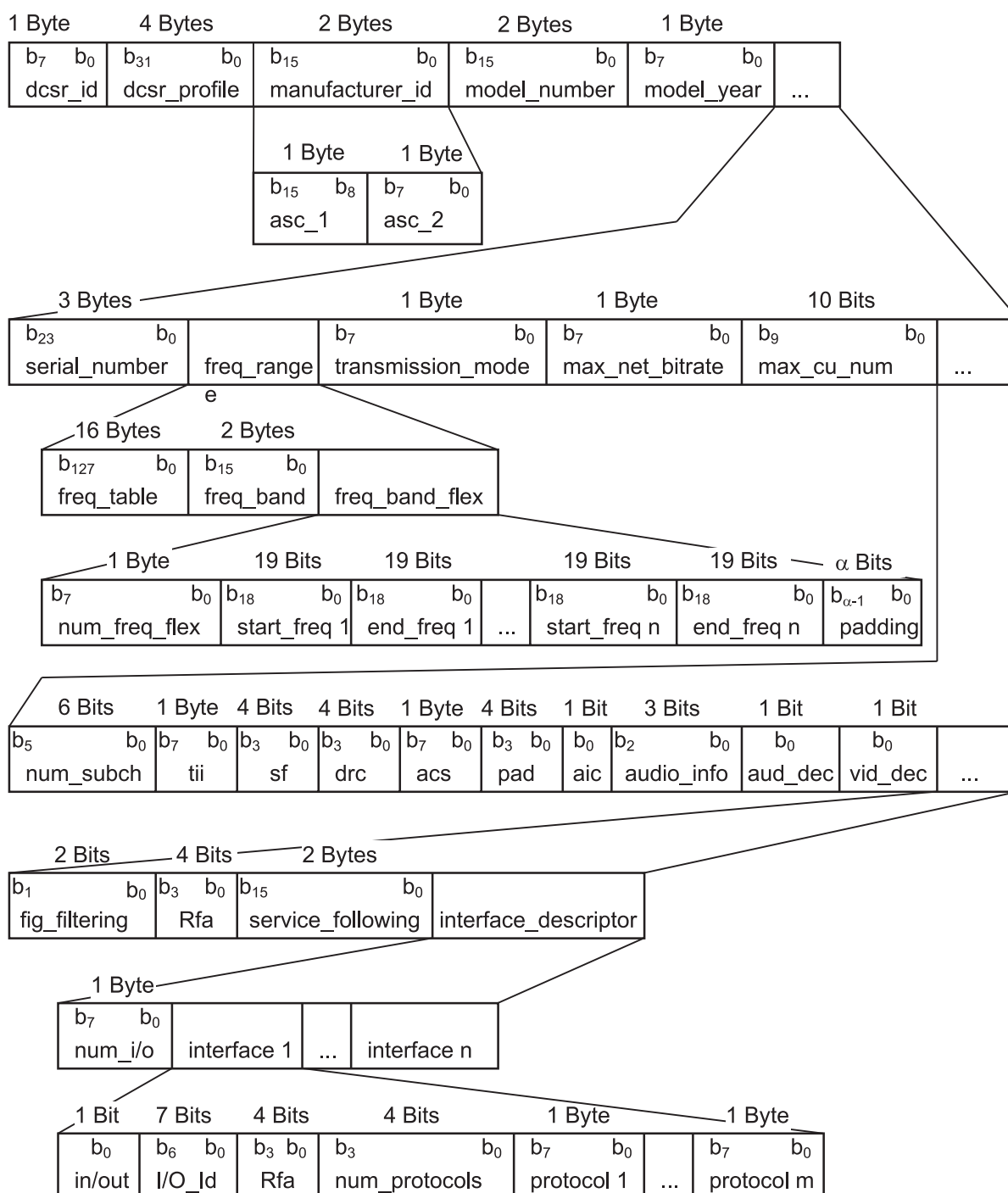
Meldung (obligatorisch)

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 bi (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 Iidentifier 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:

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 Übertragungsmodus in jedem Frequenzband unterstützt.

Bit-Flag	
b0	Übertragungsmodus I
b1	Übertragungsmodus II
b2	Übertragungsmodus III
b3	Übertragungsmodus 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)

1 Byte		1 Bit	1 Bit	3 Bits	19 Bits
b ₇	b ₀	b ₀	b ₀	b ₂	b ₀
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-Mittenfrequenz 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.