Command 0x12. SET_PROPERTY

Sets a property shown in Table 9, “FM/RDS Receiver Property Summary,” on page 55. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This command may only be sent when in powerup mode. See Figure 30, “CTS and SET_PROPERTY Command Complete tCOMP Timing Model,” on page 235 and Table 48, “Command Timing Parameters for the FM Receiver,” on page 237.

Command Arguments: Five

Response bytes: None

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	0	1	0	0	1	0
ARG1	0	0	0	0	0	0	0	0
ARG2	$PROP_H$[7:0]							
ARG3	$PROP_L$[7:0]							
ARG4	$PROPD_H$[7:0]							
ARG5	$PROPD_L$[7:0]							

ARG	Bit	Name	Function
1	7:0	Reserved	Always write to 0.
2	7:0	$PROP_H$[7:0]	**Property High Byte.** This byte in combination with $PROP_L$ is used to specify the property to modify.
3	7:0	$PROP_L$[7:0]	**Property Low Byte.** This byte in combination with $PROP_H$ is used to specify the property to modify.
4	7:0	$PROPD_H$[7:0]	**Property Value High Byte.** This byte in combination with $PROPD_L$ is used to set the property value.
5	7:0	$PROPD_L$[7:0]	**Property Value Low Byte.** This byte in combination with $PROPD_H$ is used to set the property value.

Property 0x4000. RX_VOLUME (Not applicable to Si4749)

Sets the audio output volume. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read when in powerup mode. The default is 63.

Default: 0x003F

Step: 1

Range: 0–63

Bit	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name	0	0	0	0	0	0	0	0	0	0	VOL[5:0]					

Bit	Name	Function
15:6	Reserved	Always write to 0.
5:0	VOL	**Output Volume.** Sets the output volume level, 63 max, 0 min. Default is 63.

Property 0x4001. RX_HARD_MUTE (Not applicable to Si4749)

Mutes the audio output. L and R audio outputs may be muted independently. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read when in powerup mode. The default is unmute (0x0000).

Default: 0x0000

Bit	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LMUTE	RMUTE

Bit	Name	Function
15:2	Reserved	Always write to 0.
1	LMUTE	Mutes L Audio Output.
0	RMUTE	Mutes R Audio Output.

Command 0x14. GET_INT_STATUS

Updates bits 6:0 of the status byte. This command should be called after any command that sets the STCINT, RDSINT, or RSQINT bits. When polling this command should be periodically called to monitor the STATUS byte, and when using interrupts, this command should be called after the interrupt is set to update the STATUS byte. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This command may only be set when in powerup mode.

Command arguments: None

Response bytes: None

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	0	1	0	1	0	0

Response

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT

Command 0x20. FM_TUNE_FREQ

Sets the FM Receive to tune a frequency between 64 and 108 MHz in 10 kHz units. The CTS bit (and optional interrupt) is set when it is safe to send the next command. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The optional STC interrupt is set when the command completes. The STCINT bit is set only after the GET_INT_STATUS command is called. This command may only be sent when in powerup mode. The command clears the STC bit if it is already set. See Figure 29, “CTS and STC Timing Model,” on page 235 and Table 48, “Command Timing Parameters for the FM Receiver,” on page 237.

FM: LO frequency is 128 kHz above RF for RF frequencies ≤ 90 MHz and 128 kHz below RF for RF frequencies > 90 MHz. For example, LO frequency is 80.128 MHz when tuning to 80.00 MHz.

Note: For FMRX components 2.0 or earlier, tuning range is 76–108 MHz.

Note: Fast bit is supported in FMRX components 4.0 or later.

Note: Freeze bit is supported in FMRX components 4.0 or later.

Command arguments: Four

Response bytes: None

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	1	0	0	0	0	0
ARG1	0	0	0	0	0	0	FREEZE	FAST
ARG2	$FREQ_H$[7:0]							
ARG3	$FREQ_L$[7:0]							
ARG4	ANTCAP[7:0]							

ARG	Bit	Name	Function
1	7:1	Reserved	Always write to 0.
1	1	FREEZE	**Freeze Metrics During Alternate Frequency Jump.** If set will cause the blend, hicut, and softmute to transition as a function of the associated attack/release parameters rather than instantaneously when tuning to alternate station.
1	0	FAST	**FAST Tuning.** If set, executes fast and invalidated tune. The tune status will not be accurate.
2	7:0	$FREQ_H$[7:0]	**Tune Frequency High Byte.** This byte in combination with $FREQ_L$ selects the tune frequency in 10 kHz. In FM mode the valid range is from 6400 to 10800 (64–108 MHz).
3	7:0	$FREQ_L$[7:0]	**Tune Frequency Low Byte.** This byte in combination with $FREQ_H$ selects the tune frequency in 10 kHz. In FM mode the valid range is from 6400 to 10800 (64–108 MHz).
4	7:0	ANTCAP[7:0]	**Antenna Tuning Capacitor (valid only when using TXO/LPI pin as the antenna input).** This selects the value of the antenna tuning capacitor manually, or automatically if set to zero. The valid range is 0 to 191. Automatic capacitor tuning is recommended. **Note:** When tuned manually, the varactor is offset by four codes. For example, if the varactor is set to a value of 5 manually, when read back the value will be 1. The four codes (1pf) delta accounts for the capacitance at the chip.

Response

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT

Command 0x21. FM_SEEK_START

Begins searching for a valid frequency. Clears any pending STCINT or RSQINT interrupt status. The CTS bit (and optional interrupt) is set when it is safe to send the next command. RSQINT status is only cleared by the RSQ status command when the INTACK bit is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The optional STC interrupt is set when the command completes. The STCINT bit is set only after the GET_INT_STATUS command is called. This command may only be sent when in powerup mode. The command clears the STCINT bit if it is already set. See Figure 29, “CTS and STC Timing Model,” on page 235 and Table 48, “Command Timing Parameters for the FM Receiver,” on page 237.

Command arguments: One

Response bytes: None

Command

Bit	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
CMD	0	0	1	0	0	0	0	1
ARG1	0	0	0	0	SEEKUP	WRAP	0	0

ARG	**Bit**	**Name**	**Function**
1	7:4	Reserved	Always write to 0.
1	3	SEEKUP	**Seek Up/Down.** Determines the direction of the search, either UP = 1, or DOWN = 0.
1	2	WRAP	**Wrap/Halt.** Determines whether the seek should Wrap = 1, or Halt = 0 when it hits the band limit.
1	1:0	Reserved	Always write to 0.

Response

Bit	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT

Command 0x22. FM_TUNE_STATUS

Returns the status of FM_TUNE_FREQ or FM_SEEK_START commands. The command returns the current frequency, RSSI, SNR, multipath, and the antenna tuning capacitance value (0-191). The command clears the STCINT interrupt bit when INTACK bit of ARG1 is set. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This command may only be sent when in powerup mode.

Command arguments: One

Response bytes: Seven

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	1	0	0	0	1	0
ARG1	0	0	0	0	0	0	CANCEL	INTACK

ARG	Bit	Name	Function
1	7:2	Reserved	Always write to 0.
1	1	CANCEL	**Cancel seek.** If set, aborts a seek currently in progress.
1	0	INTACK	**Seek/Tune Interrupt Clear.** If set, clears the seek/tune complete interrupt status indicator.

Response

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT
RESP1	BLTF	X	X	X	X	X	AFCRL	VALID
RESP2	READFREQ$_H$[7:0]							
RESP3	READFREQ$_L$[7:0]							
RESP4	RSSI[7:0]							
RESP5	SNR[7:0]							
RESP6	MULT[7:0]							
RESP7	READANTCAP[7:0] (Si4704/05/06/2x only)							

RESP	Bit	Name	Function
1	7	BLTF	**Band Limit.** Reports if a seek hit the band limit (WRAP = 0 in FM_START_SEEK) or wrapped to the original frequency (WRAP = 1).
1	6:2	Reserved	Always returns 0.
1	1	AFCRL	**AFC Rail Indicator.** Set if the AFC rails.
1	0	VALID	**Valid Channel.** Set if the channel is currently valid as determined by the seek/tune properties (0x1403, 0x1404, 0x1108) and would have been found during a Seek.
2	7:0	READFREQ$_H$[7:0]	**Read Frequency High Byte.** This byte in combination with READFREQ$_L$ returns frequency being tuned (10 kHz).
3	7:0	READFREQ$_L$[7:0]	**Read Frequency Low Byte.** This byte in combination with READFREQ$_H$ returns frequency being tuned (10 kHz).
4	7:0	RSSI[7:0]	**Received Signal Strength Indicator.** This byte contains the receive signal strength when tune is complete (dBµV).
5	7:0	SNR[7:0]	**SNR.** This byte contains the SNR metric when tune is complete (dB).
6	7:0	MULT[7:0]	**Multipath.** This byte contains the multipath metric when tune is complete. Multipath indicator is available only for Si474x, Si4706-C30 and later and Si4704/05/30/31/34/35/84/85 -D50 and later.
7	7:0	READANTCAP [7:0]	**Read Antenna Tuning Capacitor (Si4704/05/06/2x only)**. This byte contains the current antenna tuning capacitor value.

Command 0x23. FM_RSQ_STATUS

Returns status information about the received signal quality. The commands returns the RSSI, SNR, frequency offset, and stereo blend percentage. It also indicates valid channel (VALID), soft mute engagement (SMUTE), and AFC rail status (AFCRL). This command can be used to check if the received signal is above the RSSI high threshold as reported by RSSIHINT, or below the RSSI low threshold as reported by RSSILINT. It can also be used to check if the signal is above the SNR high threshold as reported by SNRHINT, or below the SNR low threshold as reported by SNRLINT. For the Si4706/4x, it can be used to check if the detected multipath is above the multipath high threshold as reported by MULTHINT, or below the multipath low threshold as reported by MULTLINT. If the PILOT indicator is set, it can also check whether the blend has crossed a threshold as indicated by BLENDINT. The command clears the RSQINT, BLENDINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT, MULTHINT, and MULTLINT interrupt bits when INTACK bit of ARG1 is set. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This command may only be sent when in powerup mode.

Command arguments: One

Response bytes: Seven

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	1	0	0	0	1	1
ARG1	0	0	0	0	0	0	0	INTACK

ARG	Bit	Name	Function
1	0	INTACK	**Interrupt Acknowledge.** 0 = Interrupt status preserved. 1 = Clears RSQINT, BLENDINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT, MULTHINT, MULTLINT.

Response

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT
RESP1	BLENDINT	X	MULTHINT	MULTLINT	SNRHINT	SNRLINT	RSSIHINT	RSSIILINT
RESP2	X	X	X	X	SMUTE	X	AFCRL	VALID
RESP3	PILOT	STBLEND[6:0]						
RESP4	RSSI[7:0]							
RESP5	SNR[7:0]							
RESP6	MULT[7:0]							
RESP7	FREQOFF[7:0]							

RESP	Bit	Name	Function
1	7	BLENDINT	**Blend Detect Interrupt.** 0 = Blend is within the Blend threshold settings. 1 = Blend goes above or below the Blend threshold settings.
1	5	MULTHINT	**Multipath Detect High (Si474x, Si4706-C30 and later and Si4704/05/30/31/34/35/84/85-D50 and later only).** 0 = Detected multipath value has not exceeded above the Multipath high threshold. 1 = Detected multipath value has exceeded above the Multipath high threshold.
1	4	MULTLINT	**Multipath Detect Low (Si474x, Si4706-C30 and later and Si4704/05/30/31/34/35/84/85-D50 and later only).** 0 = Detected multipath value has not fallen below the Multipath low threshold. 1 = Detected multipath value has fallen below the Multipath low threshold.
1	3	SNRHINT	**SNR Detect High.** 0 = Received SNR has not exceeded above SNR high threshold. 1 = Received SNR has exceeded above SNR high threshold.
1	2	SNRLINT	**SNR Detect Low.** 0 = Received SNR has not fallen below SNR low threshold. 1 = Received SNR has fallen below SNR low threshold.
1	1	RSSIHINT	**RSSI Detect High.** 0 = RSSI has not exceeded above RSSI high threshold. 1 = RSSI has exceeded above RSSI high threshold.
1	0	RSSILINT	**RSSI Detect Low.** 0 = RSSI has not fallen below RSSI low threshold. 1 = RSSI has fallen below RSSI low threshold.
2	3	SMUTE	**Soft Mute Indicator.** Indicates soft mute is engaged.
2	1	AFCRL	**AFC Rail Indicator.** Set if the AFC rails.
2	0	VALID	**Valid Channel.** Set if the channel is currently valid and would have been found during a Seek.
3	7	PILOT	**Pilot Indicator.** Indicates stereo pilot presence.
3	6:0	STBLEND[6:0]	**Stereo Blend Indicator.** Indicates amount of stereo blend in% (100 = full stereo, 0 = full mono).
4	7:0	RSSI[7:0]	**Received Signal Strength Indicator.** Contains the current receive signal strength (0–127 dBµV).
5	7:0	SNR[7:0]	**SNR.** Contains the current SNR metric (0–127 dB).
6	7:0	MULT[7:0]	**Multipath (Si474x, Si4706-C30 and later and Si4704/05/30/31/34/35/84/85-D50 and later only).** Contains the current multipath metric. (0 = no multipath; 100 = full multipath)
7	7:0	FREQOFF[7:0]	**Frequency Offset.** Signed frequency offset (kHz).

Command 0x24. FM_RDS_STATUS

Returns RDS information for current channel and reads an entry from the RDS FIFO. RDS information includes synch status, FIFO status, group data (blocks A, B, C, and D), and block errors corrected. This command clears the RDSINT interrupt bit when INTACK bit in ARG1 is set and, if MTFIFO is set, the entire RDS receive FIFO is cleared (FIFO is always cleared during FM_TUNE_FREQ or FM_SEEK_START). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This command may only be sent when in power up mode. The FIFO size is 25 groups for FMRX component 2.0 or later, and 14 for FMRX component 1.0.

Notes:

1. FM_RDS_STATUS is supported in FMRX component 2.0 or later.
2. MTFIFO is not supported in FMRX component 2.0.

Command arguments: One

Response bytes: Twelve

Command

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	1	0	0	1	0	0
ARG1	0	0	0	0	0	STATUSONLY	MTFIFO	INTACK

ARG	Bit	Name	Function
1	2	STATUSONLY	**Status Only.** Determines if data should be removed from the RDS FIFO. 0 = Data in BLOCKA, BLOCKB, BLOCKC, BLOCKD, and BLE contain the oldest data in the RDS FIFO. 1 = Data in BLOCKA will contain the last valid block A data received for the current station. Data in BLOCKB will contain the last valid block B data received for the current station. Data in BLE will describe the bit errors for the data in BLOCKA and BLOCKB.
1	1	MTFIFO	**Empty FIFO** 0 = If FIFO not empty, read and remove oldest FIFO entry. 1 = Clear RDS Receive FIFO.
1	0	INTACK	**Interrupt Acknowledge** 0 = RDSINT status preserved. 1 = Clears RDSINT.

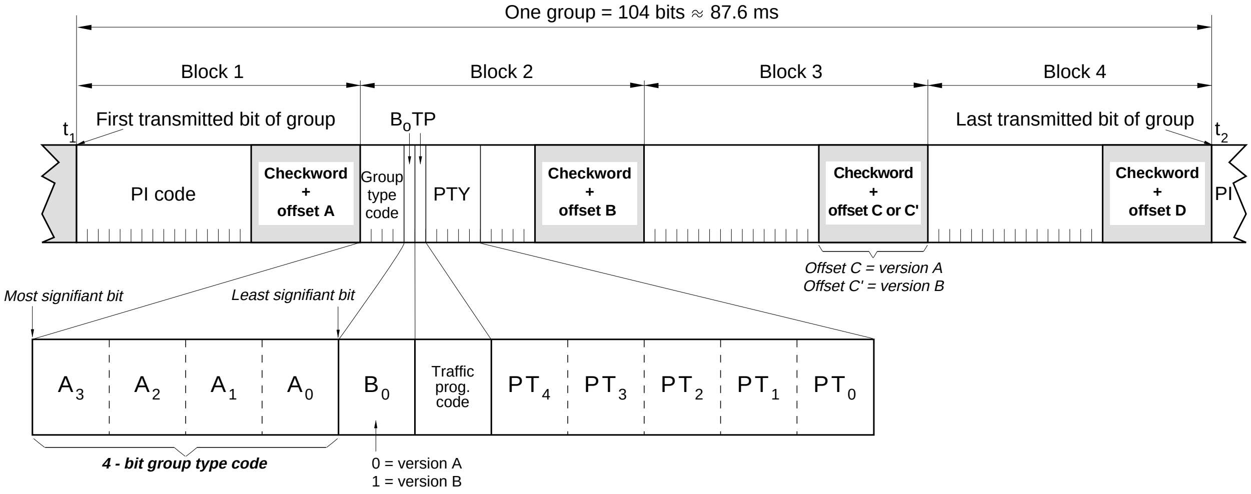
Response

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT
RESP1	X	X	RDSNEWBLOCKB	RDSNEWBLOCKA	X	RDSSYNCFOUND	RDSSYNCLOST	RDSRECV
RESP2	X	X	X	X	X	GRPLOST	X	RDSSYNC

Bit	D7	D6	D5	D4	D3	D2	D1	D0
RESP3	RDSFIFOUSED[7:0]							
RESP4	BLOCKA[15:8]							
RESP5	BLOCKA[7:0]							
RESP6	BLOCKB[15:8]							
RESP7	BLOCKB[7:0]							
RESP8	BLOCKC[15:8]							
RESP9	BLOCKC[7:0]							
RESP10	BLOCKD[15:8]							
RESP11	BLOCKD[7:0]							
RESP12	BLEA[1:0]		BLEB[1:0]		BLEC[1:0]		BLED[1:0]	

RESP	Bit	Name	Function
1	5	RDSNEWBLOCKB	**RDS New Block B.** 1 = Valid Block B data has been received.
1	4	RDSNEWBLOCKA	**RDS New Block A.** 1 = Valid Block A data has been received.
1	2	RDSSYNCFOUND	**RDS Sync Found.** 1 = Found RDS synchronization.
1	1	RDSSYNCLOST	**RDS Sync Lost.** 1 = Lost RDS synchronization.
1	0	RDSRECV	**RDS Received.** 1 = FIFO filled to minimum number of groups set by RDSFIFOCNT.
2	2	GRPLOST	**Group Lost.** 1 = One or more RDS groups discarded due to FIFO overrun.
2	0	RDSSYNC	**RDS Sync.** 1 = RDS currently synchronized.
3	7:0	RDSFIFOUSED	**RDS FIFO Used.** Number of groups remaining in the RDS FIFO (0 if empty). If non-zero, BLOCKA-BLOCKD contain the oldest FIFO entry and RDSFIFOUSED decrements by one on the next call to RDS_FIFO_STATUS (assuming no RDS data received in the interim).
4	7:0	BLOCKA[15:8]	**RDS Block A.** Block A group data from oldest FIFO entry if STATUSONLY is 0. Last valid Block A data if STATUSONLY is 1 (Si4749, Si4706-C30 and later and Si4705/31/35/85-D50 and later only).
5	7:0	BLOCKA[7:0]	
6	7:0	BLOCKB[15:8]	**RDS Block B.** Block B group data from oldest FIFO entry if STATUSONLY is 0. Last valid Block B data if STATUSONLY is 1 (Si4749, Si4706-C30 and later and Si4705/31/35/85-D50 and later only).
7	7:0	BLOCKB[7:0]	
8	7:0	BLOCKC[15:8]	**RDS Block C.** Block C group data from oldest FIFO entry.
9	7:0	BLOCKC[7:0]	

RESP	Bit	Name	Function
10	7:0	BLOCKD[15:8]	**RDS Block D.** Block D group data from oldest FIFO entry.
11	7:0	BLOCKD[7:0]	
12	7:6	BLEA[1:0]	**RDS Block A Corrected Errors.** 0 = No errors. 1 = 1–2 bit errors detected and corrected. 2 = 3–5 bit errors detected and corrected. 3 = Uncorrectable.
12	5:4	BLEB[1:0]	**RDS Block B Corrected Errors.** 0 = No errors. 1 = 1–2 bit errors detected and corrected. 2 = 3–5 bit errors detected and corrected. 3 = Uncorrectable.
12	3:2	BLEC[1:0]	**RDS Block C Corrected Errors.** 0 = No errors. 1 = 1–2 bit errors detected and corrected. 2 = 3–5 bit errors detected and corrected. 3 = Uncorrectable.
12	1:0	BLED[1:0]	**RDS Block D Corrected Errors.** 0 = No errors. 1 = 1–2 bit errors detected and corrected. 2 = 3–5 bit errors detected and corrected. 3 = Uncorrectable.



<u>*Notes to figure 9:*</u>

1. *Group type code = 4 bits (see 3.1)*
2. *B_o = version code = 1 bit (see 3.1)*
3. *PI code = Programme Identification code = 16 bits (see 3.2.1.1 and annex D)*
4. *TP = Traffic Programme Identification code = 1 bit (see 3.2.1.3)*
5. *PTY = Programme Type code = 5 bits (see 3.2.1.2 and annex F)*
6. *Checkword + offset "N" = 10 bits added to provide error protection and block and group synchronization information (see 2.3 and 2.4 and annexes A,B and C)*
7. *t_1 ‹ t_2 : Block 1 of any particular group is transmitted first and block 4 last*

Figure 9: Message format and addressing

2.3 Error protection

Each transmitted 26-bit block contains a 10-bit checkword which is primarily intended to enable the receiver/decoder to detect and correct errors which occur in transmission. This checkword (i.e. c'_9, c'_8, ... c'_o in figure 8) is the sum (modulo 2) of:

a) the remainder after multiplication by x^{10} and then division (modulo 2) by the generator polynomial g(x), of the 16-bit information word,

b) a 10-bit binary string d(x), called the "offset word",

where the generator polynomial, g(x) is given by:

$$g(x) = x^{10} + x^8 + x^7 + x^5 + x^4 + x^3 + 1$$

and where the offset values, d(x), which are different for each block within a group (see 2.4) are given in annex A.

The purpose of adding the offset word is to provide a group and block synchronisation system in the receiver/decoder (see 2.4). Because the addition of the offset is reversible in the decoder the normal additive error-correcting and detecting properties of the basic code are unaffected.

The checkword thus generated is transmitted m.s.b. (i.e. the coefficient of c'_9 in the checkword) first and is transmitted at the end of the block which it protects.

3.1.3 Group types

It was described above (see also figure 9) that the first five bits of the second block of every group are allocated to a five-bit code which specifies the application of the group and its version, as shown in table 3.

Table 3: Group types

Group type	Group type code/version					Flagged in type 1A groups	Description
	A_3	A_2	A_1	A_0	B_0		
0 A	0	0	0	0	0		Basic tuning and switching information only (see 3.1.5.1)
0 B	0	0	0	0	1		Basic tuning and switching information only (see 3.1.5.1)
1A	0	0	0	1	0		Programme Item Number and slow labelling codes only (see 3.1.5.2)
1B	0	0	0	1	1		Programme Item Number (see 3.1.5.2)
2 A	0	0	1	0	0		RadioText only (see 3.1.5.3)
2 B	0	0	1	0	1		RadioText only (see 3.1.5.3)
3 A	0	0	1	1	0		Applications Identification for ODA only (see 3.1.5.5)
3 B	0	0	1	1	1		Open Data Applications
4 A	0	1	0	0	0		Clock-time and date only (see 3.1.5.6)
4 B	0	1	0	0	1		Open Data Applications
5 A	0	1	0	1	0		Transparent Data Channels (32 channels) or ODA (see 3.1.5.8)
5 B	0	1	0	1	1		Transparent Data Channels (32 channels) or ODA (see 3.1.5.8)
6 A	0	1	1	0	0		In House applications or ODA (see 3.1.5.9)
6 B	0	1	1	0	1		In House applications or ODA (see 3.1.5.9)
7 A	0	1	1	1	0	Y	Radio Paging or ODA (see 3.1.5.10 and annex M)
7 B	0	1	1	1	1		Open Data Applications
8 A	1	0	0	0	0	Y	Traffic Message Channel or ODA (see 3.1.5.12)
8 B	1	0	0	0	1		Open Data Applications
9 A	1	0	0	1	0	Y	Emergency Warning System or ODA (see 3.1.5.13)
9 B	1	0	0	1	1		Open Data Applications
10 A	1	0	1	0	0		Programme Type Name
10 B	1	0	1	0	1		Open Data Applications
11 A	1	0	1	1	0		Open Data Applications
11 B	1	0	1	1	1		Open Data Applications
12 A	1	1	0	0	0		Open Data Applications
12 B	1	1	0	0	1		Open Data Applications
13 A	1	1	0	1	0	Y	Enhanced Radio Paging or ODA (see annex M)
13 B	1	1	0	1	1		Open Data Applications
14 A	1	1	1	0	0		Enhanced Other Networks information only (see 3.1.5.19)
14 B	1	1	1	0	1		Enhanced Other Networks information only (see 3.1.5.19)
15 A	1	1	1	1	0		Defined in RBDS only
15 B	1	1	1	1	1		Fast switching information only (see 3.1.5.20)

3.1.4.2 Open Data Applications - Group structure

Open Data Applications must use the format shown in figure 10 for ODA type A groups and in figure 11 for ODA type B groups.

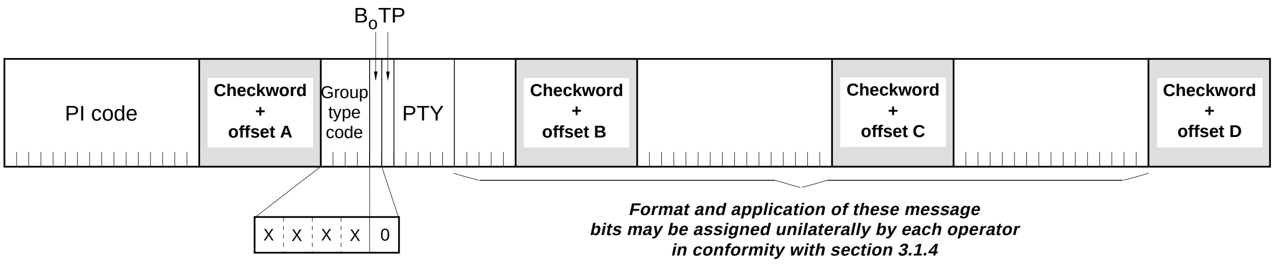

B0TP
PI code
Checkword + offset A
Group type code
PTY
Checkword + offset B
Checkword + offset C
Checkword + offset D
X X X X 0
Format and application of these message bits may be assigned unilaterally by each operator in conformity with section 3.1.4

Figure 10: ODA type A groups

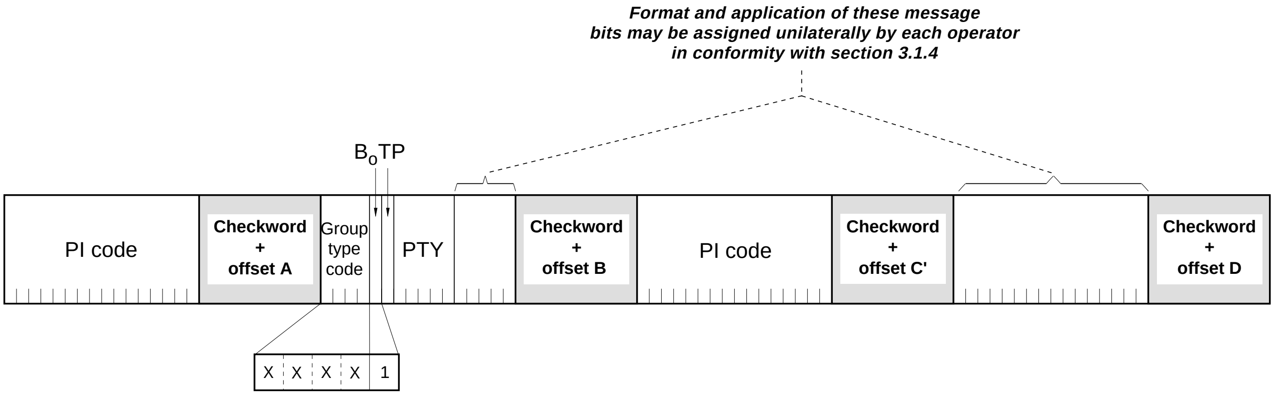


Figure 11: ODA type B groups

3.1.5 Coding of the Group types

3.1.5.1 Type 0 groups: Basic tuning and switching information

The repetition rates of type 0 groups must be chosen in compliance with 3.1.3.

Figure 12 shows the format of type 0A groups and figure 13 the format of type 0B groups.

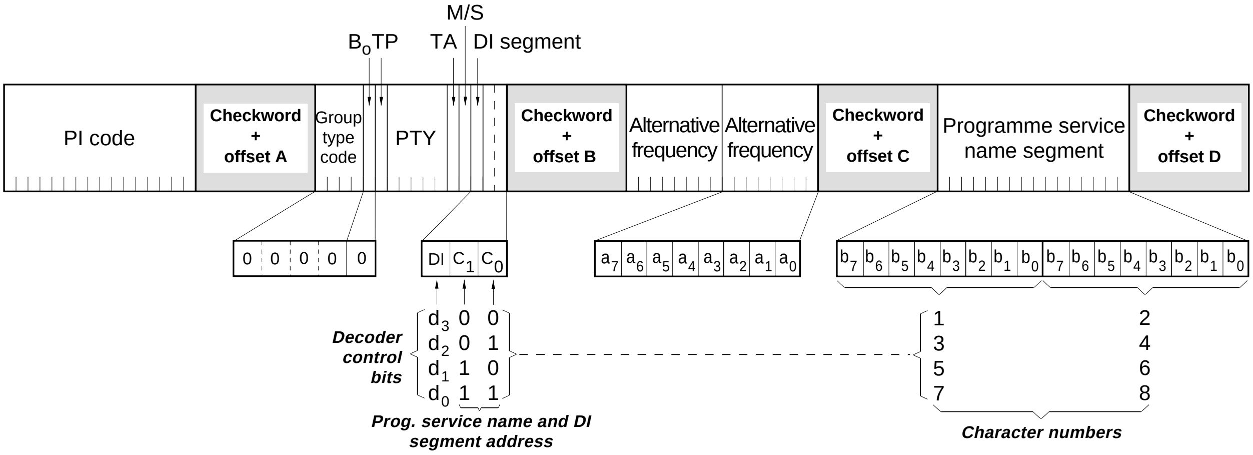


Figure 12: Basic tuning and switching information - Type 0A group

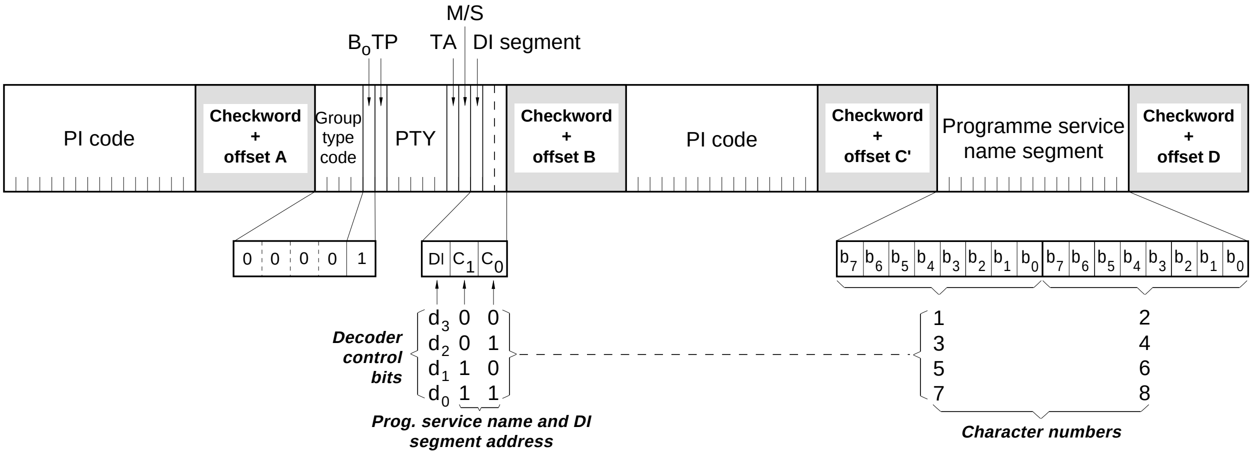


Figure 13: Basic tuning and switching information - Type 0B group

Type 0A groups are usually transmitted whenever alternative frequencies exist. Type 0B groups without any type 0A groups may be transmitted only when no alternative frequencies exist.

There are two methods (A and B) for transmission of alternative frequencies (see 3.2.1.6.2).

The Programme Service name comprises eight characters, intended for static display on a receiver. It is the primary aid to listeners in programme service identification and selection. The use of PS to transmit text other than a single eight character name is not permitted (see also 3.2.2). Transmission of a PS name usually takes four type 0A groups, but to allow an instant display of the PS when a receiver pre-set is selected, the PS name is often stored for subsequent recall from memory when a programme service is selected. For this reason PS should generally be invariant.

If a broadcaster wishes to transmit longer Programme Service names, programme-related information or any other text, then RadioText provides this feature.

Notes on Type 0 groups:

1. *Version B differs from version A only in the contents of block 3, the offset word in block 3, and, of course, the version code B_0*
2. *For details of Programme Identification (PI), Programme Type (PTY) and Traffic Programme (TP) code, see figure 9, 3.2.1 and annexes D and F.*
3. *TA = Traffic announcement code (1 bit) (see 3.2.1.3).*
4. *MS = Music Speech switch code (1 bit) (see 3.2.1.4).*
5. *DI= Decoder-identification control code (4 bits) (see 3.2.1.5). This code is transmitted as 1 bit in each type 0 group. The Programme Service name and DI segment address code (C_1 and C_0) serves to locate these bits in the DI codeword. Thus in a group with C_1C_0 = "00" the DI bit in that group is d_3 . These code bits are transmitted most significant bit (d_3) first.*
6. *Alternative frequency codes (2 x 8 bits) (see 3.2.1.6).*
7. *Programme Service name (for display) is transmitted as 8-bit character as defined in the 8-bit code-tables in annex E. Eight characters (including spaces) are allowed for each network and are transmitted as a 2-character segment in each type 0 group. These segments are located in the displayed name by the code bits C_1 and C_o in block 2. The addresses of the characters increase from left to right in the display. The most significant bit (b_7) of each character is transmitted first.*

3.1.5.3 Type 2 groups: RadioText

Figure 16 shows the format of type 2A groups and figure 17 the format of type 2B groups.

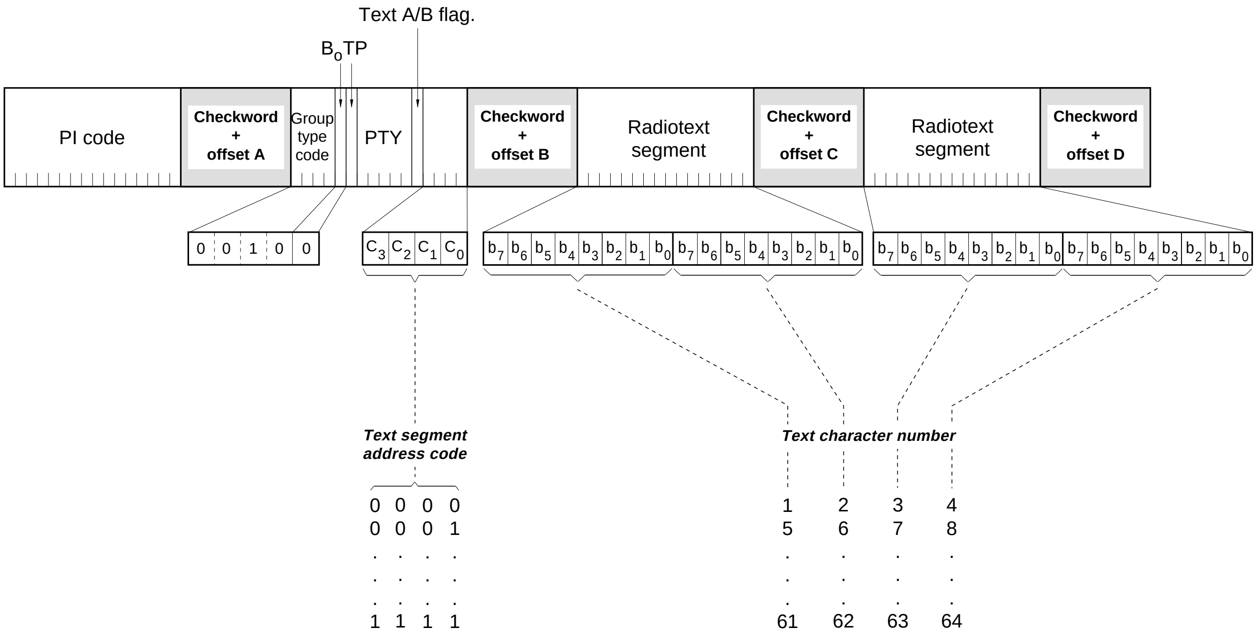


Figure 16: RadioText - Type 2A group

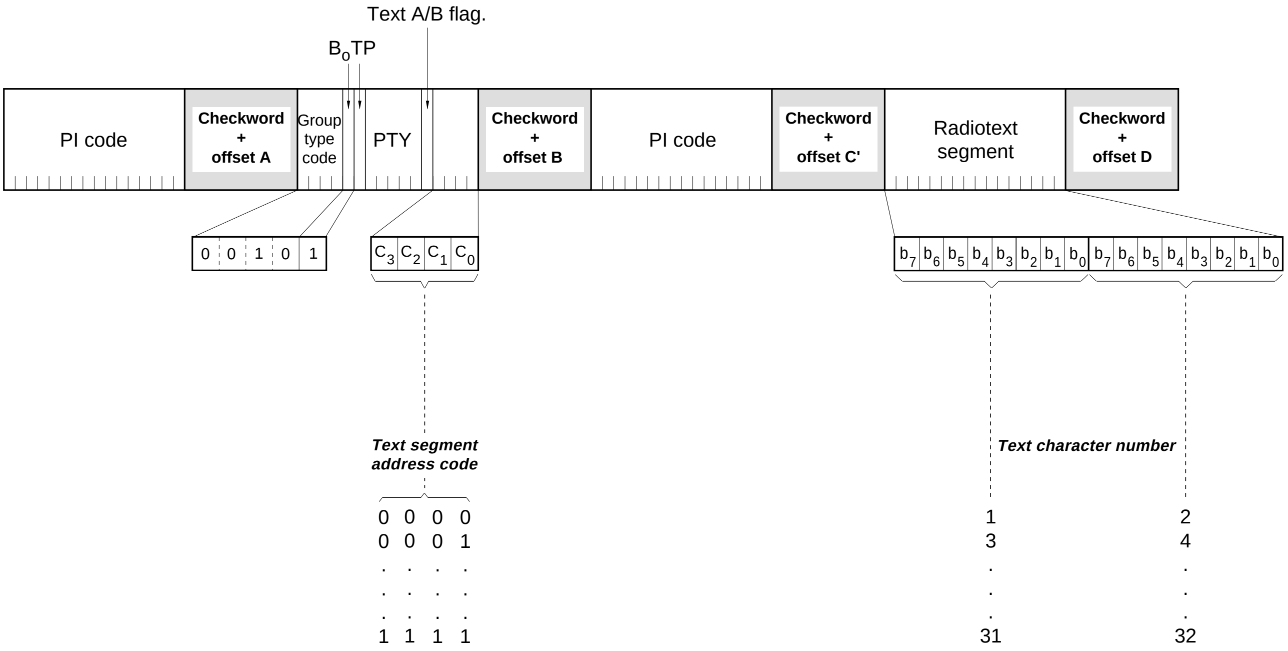


Figure 17: RadioText - Type 2B group

The 4-bit text segment address defines in the current text the position of the text segments contained in the third (version A only) and fourth blocks. Since each text segment in version 2A groups comprises four characters, messages of up to 64 characters in length can be sent using this version. In version 2B groups, each text segment comprises only two characters and therefore when using this version the maximum message length is 32 characters.

A new text must start with segment address “0000" and there must be no gaps up to the highest used segment address of the current message. The number of text segments is determined by the length of the message, and each message should be ended by the code 0D (Hex) - carriage return - if the current message requires less than 16 segment addresses.

If a display which has fewer than 64 characters is used to display the RadioText message then memory should be provided in the receiver/decoder so that elements of the message can be displayed sequentially. This may, for example, be done by displaying elements of text one at a time in sequence, or, alternatively by scrolling the displayed characters of the message from right to left.

Code 0A (Hex) - line feed - may be inserted to indicate a preferred line break.

It should be noted that because of the above considerations there is possible ambiguity between the addresses contained in version A and those contained in version B. For this reason a mixture of type 2A and type 2B groups must not be used when transmitting any one given message.

- An important feature of type 2 groups is the Text A/B flag contained in the second block. Two cases occur:

  If the receiver detects a change in the flag (from binary "0" to binary "1" or vice-versa), then the whole RadioText display should be cleared and the newly received RadioText message segments should be written into the display.

- If the receiver detects no change in the flag, then the received text segments or characters should be written into the existing displayed message and those segments or characters for which no update is received should be left unchanged.

When this application is used to transmit a 32-character message, at least three type 2A groups or at least six type 2B groups should be transmitted in every two seconds.

It may be found from experience that all RadioText messages should be transmitted at least twice to improve reception reliability.

Notes on Type 2 groups:

1. *RadioText is transmitted as 8-bit characters as defined in the 8-bit code-tables in annex E. The most significant bit (b_7) of each character is transmitted first.*
2. *The addresses of the characters increase from left to right in the display.*