

MOST

Media Oriented Systems Transport

Multimedia and Control
Networking Technology

MOST FunctionBlock AmFmTuner

Rev 2.4.2

09/2003



Legal Notice

COPYRIGHT

© Copyright 1999 - 2003 MOST Cooperation. All rights reserved.

LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

MOST Cooperation

Administration
Bannwaldallee 48
D-76185 Karlsruhe
Germany

Tel: (+49) (0) 721 966 50 00

Fax: (+49) (0) 721 966 50 01

E-mail: contact@mostcooperation.com

Web: www.mostcooperation.com



© Copyright 1999 - 2003 MOST Cooperation
All rights reserved

MOST is a registered trademark

1	INTRODUCTION	7
2	FUNCTIONBLOCK DEFINITION	7
2.1	AmFmTuner (FBlockID=0x40)	7
2.1.1	FktIDs (0x000)	7
2.1.2	Notification (0x001)	7
2.1.3	NotificationCheck (0x002)	9
2.1.4	SourceInfo (0x100)	10
2.1.5	Allocate (0x101)	12
2.1.6	DeAllocate (0x102)	13
2.1.7	SourceActivity (0x103)	13
2.1.8	SourceName (0x104)	14
2.1.9	SourceConnect (0x105)	14
2.1.10	SourceDisConnect (0x106)	16
2.1.11	SourceRouting (0x107)	16
2.1.12	SyncDataInfo (0x116)	17
2.1.13	ATWaveband (0x200)	18
2.1.14	ATStationInfo (0x201)	19
2.1.15	ATSeek (0x204)	22
2.1.16	ATPI (0x205)	23
2.1.17	ATFrequency (0x206)	23
2.1.18	ATPresetSave (0x400)	24
2.1.19	ATPresetList1 (0x401)	25
2.1.20	ATPresetList2 (0x402)	27
2.1.21	ATPresetShortList1 (0x403)	29
2.1.22	ATPresetShortList2 (0x404)	31
2.1.23	ATPresetShortList3 (0x405)	32
2.1.24	ATPresetShortList4 (0x406)	34
2.1.25	ATPresetShortList5 (0x407)	35
2.1.26	ATPresetShortList6 (0x408)	37
2.1.27	ATPresetShortList7 (0x409)	38
2.1.28	ATPresetShortList8 (0x40A)	40
2.1.29	DTStations (0x410)	41
2.1.30	DTPStations (0x412)	44
2.1.31	TPSwitch (0x450)	45
2.1.32	RDSSwitch (0x451)	45
2.1.33	REG (0x452)	46
2.1.34	TAEscape (0x453)	47
2.1.35	Autostore (0x454)	47
2.1.36	TAInfo (0x455)	47
2.1.37	TAMessage (0x456)	48
2.1.38	PTY31Message (0x459)	48
2.1.39	Radiotext (0x45A)	49
2.1.40	AnnouncementEscape (0x45B)	50
2.1.41	JpTraf (0x45C)	50
3	FUNCTIONBLOCK DYNAMIC SPECIFICATION	52

Bibliography MOST Function Catalog

This is a list of released FunctionBlocks at the release time of this specification. FBlocks which are released later are not reflected in this list.

FBlockID	FunctionBlock
0x00	GeneralFBlock
0x00	GeneralPlayer
0x01	NetBlock
0x02	NetworkMaster
0x03	ConnectionMaster
0x06	Diagnosis
0x0F	Enhanced Testability
0x22	AudioAmplifier
0x26	MicrophoneInput
0x30	AudioTapePlayer
0x31	AudioDiskPlayer
0x34	DVDVideoPlayer
0x40	AmFmTuner
0x41	TMCTuner
0x42	TVTuner
0x50	Telephone
0x51	GeneralPhoneBook
0x60	GraphicDisplay
0xFF	UniqueFunctions

AmFmTuner FBlock (0x40) Change History

Changes AmFmTuner FBlock 2.4.1 to AmFmTuner FBlock 2.4.2

Change Ref.	FktID	Changes
2.4.2-001	0x205	- Changed valid range for parameter NSteps
2.4.2-002	0x206	- Changed valid range for parameter NSteps
2.4.2-003	0x400	- Changed description of parameter PresetNumber

Changes AmFmTuner FBlock 2.4 to AmFmTuner FBlock 2.4.1

Change Ref.	FktID	Changes
2.4.1-001	0x002	- Changed description of parameter FktIDList.
		-
		-

1 Introduction

A MOST Function Catalog is a collection of MOST FunctionBlocks.

This document contains the specification of a FunctionBlock. MOST FunctionBlocks are standardized and maintained by MOST workgroup Device Architecture (WG_DA). In order to speed up the process of making new Function Blocks available, every Function Block will be updated individually as required.

2 FunctionBlock Definition

2.1 AmFmTuner (FBlockID=0x40)

This function block makes the functions for AmFmTuner applications available.

2.1.1 FktIDs (0x000)

With the property FktIDs the functions of a function block may be inquired.

2.1.1.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	FktIDs (0x000)	Get	
		Status	BitField
		Error	ErrorCode, ErrorInfo

2.1.1.2 Parameter

BitField

RLE-coded bitfield of available functions Remark: FktIDs are 12 Bit encoded !

Basis datatype	Length	Description
Stream		FktID1, FktID2, ...

2.1.2 Notification (0x001)

This property administrates the Notification Matrix of a function block.

2.1.2.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	Notification (0x001)	Set	Control, DeviceID, FktIDList
		Get	FktID
		Status	FktID, DeviceIDList
		Error	ErrorCode, ErrorInfo

2.1.2.2 Parameter

Control

The parameter Control determines, where the entry has to be done, or the deletion respectively. SetAll = Entry of DeviceID in all properties that support Notification
SetFunction = Entry of DeviceID for the specified functions in the Notification-Matrix
ClearAll = Deletion of DeviceID at all functions of the Notification-Matrix ClearFunction = Deletion of DeviceID for the specified functions in the Notification-Matrix

Basis datatype	Range of values	Code	Description
Enum	0x00..0x03	0x00	SetAll
		0x01	SetFunction
		0x02	ClearAll
		0x03	ClearFunction

DeviceID

Rx/TxLog of a device or group address

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

FktID

Function

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

DeviceIDList

List of Devices

Basis datatype	Length	Description
Stream		DeviceID {, DeviceID}

FktIDList

List of functions with a maximum of 4.

Basis datatype	Length	Description
Stream	8	FktID {, FktID}

2.1.3 NotificationCheck (0x002)

Under certain system conditions it can be helpfull if a device can check whether its entries are still existent in the notification matrix or not. In case of error, a device is able to renew its entries.

2.1.3.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	NotificationCheck (0x002)	Get	DeviceID
		Status	DeviceID, FktIDList
		Error	ErrorCode, ErrorInfo

2.1.3.2 Parameter

FktIDList

List of functions.

Basis datatype	Length	Description
Stream	-	FktID {, FktID}

FktID

Function

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

DeviceID

Rx/TxLog of a device or groupaddress

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

ErrorCode

ErrorInfo

2.1.4 SourceInfo (0x100)

This property gives particulars about the type of synchronous source data.

2.1.4.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceInfo (0x100)	Get	SourceNr
		Status	SourceNr, DataType, DataDescription
		Error	ErrorCode, ErrorInfo

2.1.4.2 Parameter

DataType

Type of synchronous data stream.

Basis datatype	Range of values	Code	Description
Enum	0x00..0xFF	0x00	PCM
		0x01	CDROM
		0x02	SPDIF
		0x20	MPEG1 System Stream
		0x21	MPEG2 Program Stream
		0x22	MPEG2 Transport Stream
		0x40	MPEG1 DTCP System Stream
		0x41	MPEG2 DTCP Program Stream
		0x42	MPEG2 DTCP Transport Stream
		0xFF	Unknown

DataDescription

Depending on DataType, additional information will be transported in DataDescription.

Basis datatype	Length	Description	
Stream	-	DataType	Description
		0x00	Resolution, AudioChannels, SrcDelay, ChannelList
		0x01	Blockwidth, ChannelList
		0x02	ChannelList
		0x20	Blockwidth, ChannelList

	0x21	Blockwidth, ChannelList
	0x22	Blockwidth, ChannelList
	0x40	Blockwidth, ChannelList
	0x41	Blockwidth, ChannelList
	0x42	Blockwidth, ChannelList

Resolution

Resolution of the AudioSamples in byte.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

AudioChannels

Number of audio channels.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

SrcDelay

Delay of synchronous Ddata related to the Timing Master. Remark: The parameter SrcDelay represents the register NDR.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

ChannelList

List of particular channels.

Basis datatype	Length	Description
Stream	60	Channel {, Channel}

Channel

Number of a channel

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..59	1	not_defined

BlockWidth

Number of transferred byte per MOST frame.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

SourceNr

Number of data source.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.5 Allocate (0x101)

With this method Allocate the source will be caused to occupy synchronous channels.

2.1.5.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	Allocate (0x101)	Processing	
		Result	SourceNr, SrcDelay, ChannelList
		StartResult	SourceNr
		Error	ErrorCode, ErrorInfo

2.1.5.2 Parameter

SourceNr

Number of data source (within one function block there can be more than one), e.g. 0x01 for the first source.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

SrcDelay

Delay of synchronous data related to the Timing Master. Remark: The parameter SrcDelay represents the register NDR.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

ChannelList

List of particular Channels.

Basis datatype	Length	Description
Stream	60	Channel {, Channel}

Channel

Number of a channel

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..59	1	not_defined

2.1.6 DeAllocate (0x102)

The method DeAllocate causes the source to free occupied synchronous channels.

2.1.6.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	DeAllocate (0x102)	Processing	
		Result	SourceNr
		StartResult	SourceNr
		Error	ErrorCode, ErrorInfo

2.1.6.2 Parameter

SourceNr

Number of the data source (there can be several sources in one function block), e.g. 0x01 for the first source

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.7 SourceActivity (0x103)

This methode controls the activity of an audio source.

2.1.7.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceActivity (0x103)	Processing	
		Result	SourceNr, Activity
		StartResult	SourceNr, Activity
		Error	ErrorCode, ErrorInfo

2.1.7.2 Parameter

Activity

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	Off
		0x01	Pause
		0x02	On

SourceNr

Number of data source.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.8 SourceName (0x104)

By property SourceName, an identifier of the synchronous source data can be requested.

2.1.8.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceName (0x104)	Get	SourceNr
		Status	SourceNr, SourceName
		Error	ErrorCode, ErrorInfo

2.1.8.2 Parameter

SourceName

Basis datatype	MaxSize
String	11

SourceNr

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.9 SourceConnect (0x105)

By use of the method SourceConnect a source will connect their data to the given synchronous MOST channels. NOTE: In systems without a connection master, the methods

Allocate/Deallocate must be used to route synchronous data to the MOST bus! In systems with a connection master, it is up to such master to decide whether allocation or source routing is used throughout the system.

2.1.9.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceConnect (0x105)	Processing	
		Result	SourceNr, SrcDelay
		StartResult	SourceNr, ChannelList
		Error	ErrorCode, ErrorInfo

2.1.9.2 Parameter

SourceNr

Number of data source (within one function block there can be more than one), e.g. 0x01 for the first source.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

SrcDelay

Delay of synchronous data related to the Timing Master. Remark: The parameter SrcDelay represents the register NDR.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

ChannelList

List of particular Channels.

Basis datatype	Length	Description
Stream	60	Channel {, Channel}

Channel

Number of a channel

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..59	1	none

2.1.10 SourceDisConnect (0x106)

By use of the method SourceDisConnect the synchronous channels of a source will be disconnected. This is for use with the method SourceConnect only.

2.1.10.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceDisConnect (0x106)	Processing	
		Result	SourceNr
		StartResult	SourceNr
		Error	ErrorCode, ErrorInfo

2.1.10.2 Parameter

SourceNr

Number of data source (within one function block there can be more than one), e.g. 0x01 for the first source.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

2.1.11 SourceRouting (0x107)

This property describes the relation between the source numbers of the function block and the physically existing synchronous data sources. Use this property to determine which source numbers are mutually exclusive.

2.1.11.1 Format of Function

Function classes: Array of { Number }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SourceRouting (0x107)	Get	Pos
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.11.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Since this property has only one dimension, y is unused. Valid range: x=1..number of sources (like given in SyncDataInfo), y=0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

The content depends on the parameter pos.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0 }	PhysicalSource[1], PhysicalSource[2],...,PhysicalSource[NMax]
		{ x>0 }	PhysicalSource[x]

PhysicalSource

Number to identify the physical source this logical source number is related to. The physical source numbers are tested on equality by the connection master.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

2.1.12 SyncDataInfo (0x116)

This property SyncDataInfo can be used to query the function block on how many connections it may serve as sink or source.

2.1.12.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	SyncDataInfo (0x116)	Get	
		Status	SourceCount, SinkCount
		Error	ErrorCode, ErrorInfo

2.1.12.2 Parameter

SinkCount

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

SourceCount

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.13 ATWaveband (0x200)

Audio tuner Waveband list Datatype: Record of { Waveband, SubWavebandFM, SubWavebandAM }

2.1.13.1 Format of Function

Function classes: Record of { Enumeration Enumeration Enumeration }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATWaveband (0x200)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.13.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Since this is an unidimensional construction, the second Byte y is unused (y=0=const) and the simplified notation Pos={x} is valid. Valid range: x=0..3, y=0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0 }	Waveband, SubWavebandFM, SubWavebandAM
		{ x=1 }	Waveband
		{ x=2 }	SubWavebandFM
		{ x=3 }	SubWavebandAM

Waveband

FM, AM, TRF or WB

Basis datatype	Range of values	Code	Description
----------------	-----------------	------	-------------

Enum	0x01..0x04	0x01	FM
		0x02	AM
		0x03	TRF
		0x04	WB

SubWavebandFM

Selection of FMA/FM for single tuner or FMD/FM for double tuner. FM = list of the manually stored senders FMA = list of senders found by autostore FMD = list of senders found through the double tuner

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	do not care
		0x01	FMA/FMD
		0x02	FM/FMS

SubWavebandAM

LW(A), MW(A) oder SW(A) The „...A“-extension refers to senders found by autostore, e.g. LW = list of the manually stored senders LWA = list of senders found by Autostore

Basis datatype	Range of values	Code	Description
Enum	0x00..0x06	0x00	do not care
		0x01	LW
		0x02	LWA
		0x03	MW
		0x04	MWA
		0x05	SW
		0x06	SWA

2.1.14 ATStationInfo (0x201)

Audio tuner station information Datatype: Record of { Sendername, SendernameInfo, Frequency, PTY, PI, ATStationSwitches, ReceptionQuality }

2.1.14.1 Format of Function

Function classes: Record of { String Enumeration Number Number Number BitField Number }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATStationInfo (0x201)	Get	Pos
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.14.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Since this is an unidimensional construction, the second Byte y is unused (y=0=const) and the simplified notation Pos={x} is valid. Valid range: x=0..7, y=0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..0..7	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0 }	Sendername, SendernameInfo, Frequency, PTY, PI, ATStationInfoSwitches, ReceptionQuality
		{ x=1 }	Sendername
		{ x=2 }	SendernameInfo
		{ x=3 }	Frequency
		{ x=4 }	PTY
		{ x=5 }	PI
		{ x=6 }	ATStationInfoSwitches
		{ x=7 }	ReceptionQuality

Sendername

PS or frequency

Basis datatype	MaxSize
String	8

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

Frequency

Frequency

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	kHz

PTY

Programme Type of station as code according to RDS specification.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..31	1	none

PI

ProgramIndex

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

ATStationInfoSwitches

No Description

Basis datatype	Bit #	Code	Description
Boolean	Bit 0 ... 7	False	
		True	reserved
	Bit 0	False	TP off
		True	TP on
	Bit 1	False	TA off
		True	TA on
	Bit 2	False	D0 off
		True	D0 on
	Bit 3	False	D1 off
		True	D1 on
	Bit 4	False	D2 off
		True	D2 on
	Bit 5	False	D3 off
		True	D3 on
	Bit 6	False	MS off

	Bit 7	True	MS on
		False	TMC not available
		True	TMC available

ReceptionQuality

debug information Bit0..3 = Quality, Bit4..7 = FieldStrength

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

2.1.15 ATSeek (0x204)

2.1.15.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATSeek (0x204)	Set	ATSeekMode
		Get	
		SetGet	ATSeekMode
		Status	ATSeekMode
		Error	ErrorCode, ErrorInfo

2.1.15.2 Parameter

ATSeekMode

Off = deactivate the seek function Up/Down Auto = start frequency search starting at the current frequency Up/Down Scan Preset = start to scan the preset memories Up/Down Scan Frequency = start to scan frequencies starting at the current frequency Up/Down Continuous = start continuous frequency change Up/Down Preset = change to next preset Up/Down DTStationList = change to next list entry

Basis datatype	Range of values	Code	Description
Enum	0x00..0x10	0x00	off
		0x01	Up Auto
		0x02	Up Scan Preset
		0x03	Up Scan Frequ
		0x04	Up Continous
		0x05	Up Preset
		0x06	Up DTStationList
		0x07	Down Auto
		0x08	Down Scan Preset

		0x09	Down Scan Frequ
		0x0A	Down Continous
		0x0B	Down Preset
		0x0C	Down DTStationList
		0x0D	AutoTP Up
		0x0E	AudoTP Down
		0x0F	PTY Up
		0x10	PTY Down

2.1.16 ATPI (0x205)

This Property is for adjusting the ProgramIndex PI manually.

2.1.16.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPI (0x205)	Set	PI
		Get	
		SetGet	PI
		Increment	NSteps
		Decrement	NSteps
		Status	PI
		Error	ErrorCode, ErrorInfo

2.1.16.2 Parameter

NSteps

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..255	1	not_defined

PI

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

2.1.17 ATFrequency (0x206)

This Property is for adjusting the frequency manually in the current waveband .

2.1.17.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATFrequency (0x206)	Set	Frequency
		Get	
		SetGet	Frequency
		Increment	NSteps
		Decrement	NSteps
		Status	Frequency
		Error	ErrorCode, ErrorInfo

2.1.17.2 Parameter

NSteps

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..255	1	not_defined

Frequency

Frequency

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	kHz

2.1.18 ATPresetSave (0x400)

Saving of the current station on that station selector pushbutton, specified by PresetNumber. Save to that list specified by FktID. Please note: Saving is also possible to such a list, which is currently not active. There will be no changing of the currently active list.

2.1.18.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetSave (0x400)	Start	FktID, PresetNumber
		Error	ErrorCode, ErrorInfo

2.1.18.2 Parameter

PresetNumber

Position where the current station will be stored.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..12	1	none

FktID

function

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

2.1.19 ATPresetList1 (0x401)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo, PTY, PI, TpTa}

2.1.19.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration Number Number Boolean } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetList1 (0x401)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.19.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..6

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..0..6	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], PTY[1], PI[1], TpTa[1], ..., PresetSelection[Nmax], Sendername[NMax], SendernameInfo[NMax], PTY[NMax], PI[NMax], TpTa[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x], PTY[x], PI[x], TpTa[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]
		{ x>0, y=4 }	PTY[x]
		{ x>0, y=5 }	PI[x]
		{ x>0, y=6 }	TpTa[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

PTY

Programme Type of station as code according to RDS specification.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..31	1	none

PI

ProgramIndex

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

TpTa

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	Tp Off
		True	Tp On
	Bit 1	False	Ta Off
		True	Ta On
	Bit 2 ... 7	-	reserved

2.1.20 ATPresetList2 (0x402)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo, PTY, PI, TpTa}

2.1.20.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration Number Number Boolean } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetList2 (0x402)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.20.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..6

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], PTY[1], PI[1], TpTa[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax], PTY[NMax], PI[NMax], TpTa[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x], PTY[x], PI[x], TpTa[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]
		{ x>0, y=4 }	PTY[x]
		{ x>0, y=5 }	PI[x]
		{ x>0, y=6 }	TpTa[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

PTY

ProgrammeType of station as code according to RDS specification.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..31	1	none

PI

ProgramIndex

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

TpTa

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	Tp Off
		True	Tp On
	Bit 1	False	Ta Off
		True	Ta On
	Bit 2 ... 7	-	reserved

2.1.21 ATPresetShortList1 (0x403)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.21.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList1 (0x403)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.21.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

	0x03	by the radio automatically given name (several equal PS)
	0x04	by customer fixed name
	0x05	by customer newly given name

2.1.22 ATPresetShortList2 (0x404)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.22.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList2 (0x404)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.22.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]

		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

2.1.23 ATPresetShortList3 (0x405)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.23.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList3 (0x405)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.23.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

	0x03	by the radio automatically given name (several equal PS)
	0x04	by customer fixed name
	0x05	by customer newly given name

2.1.24 ATPresetShortList4 (0x406)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.24.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList4 (0x406)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.24.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]

		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

2.1.25 ATPresetShortList5 (0x407)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.25.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList5 (0x407)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.25.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

	0x03	by the radio automatically given name (several equal PS)
	0x04	by customer fixed name
	0x05	by customer newly given name

2.1.26 ATPresetShortList6 (0x408)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.26.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList6 (0x408)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.26.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]

		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

2.1.27 ATPresetShortList7 (0x409)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.27.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList7 (0x409)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.27.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)

		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

2.1.28 ATPresetShortList8 (0x40A)

station keys Information to the on the station keys stored stations. Via parameter PresetSelection a stored station may be selected. A complex tuner can make several Preset lists available, where however a simple tuner implements only one Preset list (ATPresetList1). Datatype: Array [1..NMax] of Record of {PresetSelection, Sendername, SendernameInfo}

2.1.28.1 Format of Function

Function classes: Array of { Record of { Enumeration String Enumeration } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	ATPresetShortList8 (0x40A)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.28.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	PresetSelection[1], Sendername[1], SendernameInfo[1], ..., PresetSelection[NMax], Sendername[NMax], SendernameInfo[NMax]
		{ x>0, y=0 }	PresetSelection[x], Sendername[x], SendernameInfo[x]
		{ x>0, y=1 }	PresetSelection[x]

		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]

PresetSelection

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	not active
		0x01	active

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	15

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

2.1.29 DTStations (0x410)

Informations about stations that may be received at the moment. Datatype: Array [1..NMax] of Record of {Selected, Sendername, SendernameInfo, PTY, PI, TpTa, TMC, ReceptionQuality}

2.1.29.1 Format of Function

Function classes: Array of { Record of { Boolean String Enumeration Number Number Boolean Number Number } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	DTStations (0x410)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.29.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..8

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	Selected[1], Sendername[1], SendernameInfo[1], PTY[1], PI[1], TpTa[1], ReceptionQuality[1], TMC[1], ..., Selected[NMax], Sendername[NMax], SendernameInfo[NMax], PTY[NMax], PI[NMax], TpTa[NMax], ReceptionQuality[NMax], TMC[NMax]
		{ x>0, y=0 }	Selected[x], Sendername[x], SendernameInfo[x], PTY[x], PI[x], TpTa[x], ReceptionQuality[x], TMC[x]
		{ x>0, y=1 }	Selected[x]
		{ x>0, y=2 }	Sendername[x]
		{ x>0, y=3 }	SendernameInfo[x]
		{ x>0, y=4 }	PTY[x]
		{ x>0, y=5 }	PI[x]
		{ x>0, y=6 }	TpTa[x]
		{ x>0, y=7 }	ReceptionQuality[x]
		{ x>0, y=8 }	TMC[x]

Selected

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	Not selected
		True	Selected
	Bit 1 ... 7	-	reserved

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	8

SendernameInfo

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	Station without RDS
		0x01	via RDS received name (original)
		0x02	by the radio automatically given name (PS changes)
		0x03	by the radio automatically given name (several equal PS)
		0x04	by customer fixed name
		0x05	by customer newly given name

PTY

Programme Type of station as code according to RDS specification.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..31	1	none

PI

ProgramIndex

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

TpTa

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	Tp Off
		True	Tp On
	Bit 1	False	Ta Off
		True	Ta On
	Bit 2 ... 7	-	reserved

ReceptionQuality

debug information Bit0..3 = Quality, Bit4..7 = FieldStrength

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

TMC

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	not_defined

2.1.30 DTPStations (0x412)

Traffic program list By Selected[x]=True an array-item will be marked as active. With Selected[1]=True the automatic Traffic program selection will be active. Datatype: Array [1..30] of Record of {Selected, Sendername}

2.1.30.1 Format of Function

Function classes: Array of { Record of { Boolean String } }

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	DTPStations (0x412)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.30.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Valid range: x=0..NMax, y=0..2

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..NMax	1	not_defined

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Stream	-	Pos	Data
		{ x=0, y=0 }	Selected[1], Sendername[1], ..., Selected[NMax], Sendername[NMax]
		{ x>0, y=0 }	Selected[x], Sendername[x]
		{ x>0, y=1 }	Selected[x]
		{ x>0, y=2 }	Sendername[x]

Selected

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	Not selected
		True	selected
	Bit 1 ... 7	-	reserved

Sendername

frequency without unit or RDS-name

Basis datatype	MaxSize
String	8

2.1.31 TPSwitch (0x450)

TrafficProgram

2.1.31.1 Format of Function

Function classes: Switch

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	TPSwitch (0x450)	Set	TPOnOff
		Get	
		SetGet	TPOnOff
		Status	TPOnOff
		Error	ErrorCode, ErrorInfo

2.1.31.2 Parameter

TPOnOff

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	TP Off
		True	TP On
	Bit 1 ... 7	-	reserved

2.1.32 RDSSwitch (0x451)

RadioDataSystem

2.1.32.1 Format of Function

Function classes: Switch

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	RDSSwitch (0x451)	Set	RDSOnOff
		Get	
		SetGet	RDSOnOff
		Status	RDSOnOff
		Error	ErrorCode, ErrorInfo

2.1.32.2 Parameter

RDSOnOff

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	RDS Off
		True	RDS On
	Bit 1 ... 7	-	reserved

2.1.33 REG (0x452)

Regionalization

2.1.33.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	REG (0x452)	Set	REGMode
		Get	
		SetGet	REGMode
		Status	REGMode
		Error	ErrorCode, ErrorInfo

2.1.33.2 Parameter

REGMode

Basis datatype	Range of values	Code	Description
Enum	0x00..0x04	0x00	OFF
		0x01	AUTO FAST
		0x02	AUTO NORMAL

		0x03	AUTO SLOW
		0x04	ON

2.1.34 TAEscape (0x453)

By triggering TAEscape the current VF-Message will be canceled. This function is available exclusively during VF-Messages.

2.1.34.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	TAEscape (0x453)	Processing	
		Result	
		Start	
		StartResult	
		Error	ErrorCode, ErrorInfo

2.1.34.2 Parameter

2.1.35 Autostore (0x454)

Autostore in the current Waveband.

2.1.35.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	Autostore (0x454)	Processing	Flags, Class, OPTypes, Name
		Result	Flags, Class, OPTypes, Name
		Start	
		StartResult	
		Error	ErrorCode, ErrorInfo

2.1.35.2 Parameter

2.1.36 TAInfo (0x455)

Name of the current Traffic program (during an announcement the station gets the prefix "TA=").

2.1.36.1 Format of Function

Function classes: Text

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	TAInfo (0x455)	Get	
		Status	TASendername
		Error	ErrorCode, ErrorInfo

2.1.36.2 Parameter

TASendername

Basis datatype	MaxSize
String	13

2.1.37 TAMessage (0x456)

Not to be controlled by the customer, active state flags a VF-Message.

2.1.37.1 Format of Function

Function classes: Switch

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	TAMessage (0x456)	Get	
		Status	TAMessageOnOff
		Error	ErrorCode, ErrorInfo

2.1.37.2 Parameter

TAMessageOnOff

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	VF-message not active
		True	VF-message active
	Bit 1 ... 7	-	reserved

2.1.38 PTY31Message (0x459)

Used to signal PTY31.

2.1.38.1 Format of Function

Function classes: Switch

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	PTY31Message (0x459)	Get	
		Status	PTY31
		Error	ErrorCode, ErrorInfo

2.1.38.2 Parameter

PTY31

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	False	PTY31 Off
		True	PTY31 On
	Bit 1 ... 7	-	reserved

2.1.39 Radiotext (0x45A)

This property contains the Radiotext (RT) service of an RDS / RBDS Tuner

2.1.39.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	Radiotext (0x45A)	Get	
		Status	TextA, TextB
		Error	ErrorCode, ErrorInfo

2.1.39.2 Parameter

TextB

32 Characters of Radiotext with Text B Flag

Basis datatype	MaxSize
String	#NULL#

TextA

32 Characters of Radiotext with Text A Flag

Basis datatype	MaxSize
----------------	---------

String	32
--------	----

2.1.40 AnnouncementEscape (0x45B)

Interrupts the current traffic announcement or PTY message. This function is available exclusively during these messages.

2.1.40.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	AnnouncementEscape (0x45B)	Processing	
		Result	
		Start	
		StartResult	
		Error	ErrorCode, ErrorInfo

2.1.40.2 Parameter

ErrorCode

ErrorInfo

2.1.41 JpTraf (0x45C)

To receive Japanese AM traffic information. Triggered by user interface and it shall automatically go back to previous mode when done.

2.1.41.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
AmFmTuner (0x40)	JpTraf (0x45C)	Processing	
		Result	
		Start	
		StartResult	
		Error	ErrorCode, ErrorInfo

2.1.41.2 Parameter

ErrorCode

ErrorInfo

3 FunctionBlock Dynamic Specification

TBD

