SIEMENS

ICs for Consumer Electronics

Megatext Plus

SDA 5275

Delta Specification / Application Notes 01.97

SDA 5275 **Revision History:** **Current Version: 01.97**		
Previous Version:		
Page (in previous Version)	Page (in current Version)	Subjects (major changes since last revision)

Edition 01.97

This edition was realized using the software system FrameMaker®.

Published by Siemens AG,
Bereich Halbleiter, Marketing-
Kommunikation, Balanstraße 73,
81541 München

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

[1] A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

[2] Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

Table of Contents **Page**

1 Delta Specification

1.1 Overview

Megatext Plus (SDA 5275) has been developed based on the original Megatext (SDA 5273). To make it easy for the user to recognize the differences between these two versions, this delta specification is provided. It is assumed that the reader of this document is familiar with the documentation of the SDA 5273 (“Volume 1”).

Megatext Plus is completely hardware compatible to the SDA 5273. However its internal processing has been changed to enable reception of all level 2.5 related data with only little external software support.

Megatext Plus is able to request, acquire and display higher level pages automatically in realtime. The firmware processes objects, DRCS-characters, parallel attributes, CLUT and sidepanel information for 16:9 applications.

The only remaining task for the user is to reserve enough memory space for higher level pages like MOTs, POPs, DRCS-pages and related pseudopackets. For Level 2.5 transmission this number of additional pages should not exceed more than 500 packets. So the user should reserve appropriate memory for MOTs, POPs and DRCS - pages (refer to [3]). Because some part of high level information is transmitted in pseudopackets X/26, X/27, X/28 and X/29, enough memory should also be reserved in the P40 and P80 chains.

The Serial/Parallel Conversion (S/P-C) is able to handle all Level 2.5 Features which are described in the “World Standard Teletext Norm”. For the evaluation of all information stored in packets X/26, X/27, X/28, X/29 and in the linked pseudo pages the S/P-C needs additional temporary memory space in the external memory, the so-called hidden display memory.

The S/P-C first builds up a temporary page in the hidden display memory and than copies it into block 2 and block 3 of the internal DRAM of the Megatext Plus. So all 10 KBytes of the display memory are used by the S/P-C.

Because it is necessary to store a lot of additional information, Megatext Plus is **only** working with **external RAM**.

From the point of view of the user, the following changes have been made in comparison with the SDA 5273.

- CLUT of display generator
- Number and addressing of DRCS characters
- Megatext Command Interface (MCI)
- WSS support

These changes are described in the following paragraphs.

1.2 Changes in Megatext Plus Command Interface

Most of the Megatext commands are not touched, so that their functional behavior and parameters are not changed. These are commands with the number: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 17, 19, 23, 24, 31, 32, 37, 39, 40, 41, 45, 49, 50, 51 and 54. For the specification of these commands please refer to *command interface description for SDA 5273*.

The following commands are changed and described below:

- ACQ_CONTROL (0)
- SERIAL_PARALLEL_CONVERSION (47)
- WRITE_TOP_TITLE (53)

Two additional commands are included:

- INHIBIT_UPDATE (10)
- READ_TRACE_REQUEST_TABLE (52)

Most of the changes are done in the S/P-C (SERIAL_PARALLEL_CONVERSION). Because of the complex data processing involved with this command and because of details of the WST specification [3] the Level 2.5 firmware uses some resources of the Megatext Plus that can not be used by the external controller anymore. These are shown in the following table:

Resources	**Used by Command**	**Comment**
Chapter 0-3 of the external memory	Acquisition	Page Look Up Table
Chapter 4-9 of the external memory	S/P-C	Hidden display memory
Chapter 10-17 of the external memory	S/P-C	Reserved for internal usage
CLUT vector 16-31	S/P-C	These additional colors can now used from the transmitted level 2.5 data
DRCS-Address 0-23	S/P-C	Upto 24 DRCS characters are defined in the WST specification [3]
Byte 4 and 5 of block 2 and 3 in the internal memory	S/P-C	These enhanced attributes are now under control of Teletext (i.e. colors, flash modes etc.). Don’t forget to set the appropriate screen mask registers!

The Megatext Plus firmware automatically requests all DRCS, POP and MOT pages in realtime (if enabled), so that there is zero access time for higher level information, if a display page is changed. The controller has to make sure, that enough memory is available to store all this information. In addition to that, the storage of packets X/26, X/27, X/28, X/29 must be enabled.

1.2.1 ACQ_CONTROL (command no. 0d)

The changes concerning this command refer to the automatic request of POPs, DRCS, GPOPS, GDRCS pages and the reception of WSS, VPS and text data in line 16 and 23.

Input Parameters

PKT_BUF_ACQ_CONTROL = MCI0_0 (REG_13)

R_MPEX	R_MP	VPS_ENA	0	TTX_FC	ACQ_BUF_ON	R_AI	PAGE_TRACE

PAGE_TRACE Switches on/off page trace recording in the page trace memory.

1: Page trace recording is switched on.
0: Page trace recording is switched off.

R_AI 1: Automatic request of all Additional TOP Tables is enabled.
0: Automatic request of all Additional TOP Tables is disabled.

ACQ_BUF_ON Controls switching on/off of TTX data transfer to packet buffer and TTX data acquisition.

1: Data transfer to packet buffer and ACQ is switched on.
0: Data transfer to packet buffer and ACQ is switched off.

TTX_FC Teletext framing code redefinition.

1: Framing code of input parameter register TTX_FRAMING_CODE is taken to redefine framing code reference for acquisition.
0: Framing code remains the same as before giving this command, independent of register TTX_FRAMING_CODE.
VPS_ENA controls switching on/off VPS reception (*ACQ_BUF_ON* must be switched on).

VPS_ENA 1: VPS reception is switched on.
0: VPS reception is switched off.

R_MP 1: Automatic request of all Multipage TOP Tables is enabled.
0: Automatic request of Multipage TOP Tables is disabled.

R_MPEX 1: Automatic request of all extended Multipage TOP Tables is enabled.
0: Automatic request of extended Multipage TOP Tables is disabled.

TTX_FRAMING_CODE = MCI0_1(REG_12)

TTX_FC_7	TTX_FC_6	TTX_FC_5	TTX_FC_4	TTX_FC_3	TTX_FC_2	TTX_FC_1	TTX_FC_0

TTX_FC_(7:0) The framing code is compared by data acquisition with the received teletext framing code as a reference for byte synchronization.

HIGH-LEVEL-CONTROL_1 = MCI0_2 (REG_11)

E_D_27	E_GD_27	E_P_27	E_GP_27	0	E_D_MOT	0	E_P_MOT

E_P_MOT
1: Automatic request of all POPs (global and local) linked by the MOTs is enabled.
0: Automatic request of POPs linked by the MOTs is disabled.

E_D_MOT
1: Automatic request of all DRCS (global and local) linked by the MOTs is enabled.
0: Automatic request of DRCS linked by the MOTs is disabled.

E_GP_27
1: Automatic request of all Global POP tables linked by the X/27/4 is enabled.
0: Automatic request of Global POP tables linked by the X/27/4 is disabled.

E_P_27
1: Automatic request of all Normal POP tables linked by the X/27/4 is enabled.
0: Automatic request of Normal POP tables linked by the X/27/4 is disabled.

E_GD_27
1: Automatic request of all Global DRCS tables linked by the X/27/4 is enabled.
0: Automatic request of Global DRCS tables linked by the X/27/4 is disabled.

E_D_27
1: Automatic request of all Normal DRCS tables linked by the X/27/4 is enabled.
0: Automatic request of Normal DRCS tables linked by the X/27/4 is disabled.

HIGH-LEVEL-CONTROL_2 = MCI0_3 (REG_10)

0	0	0	0	0	0	ENA_LINE_16/23	E_MOT

E_MOT
1: Automatic request of all MOT - tables is enabled.
0: Automatic request of all MOT - tables is disabled.

ENA_LINE_16/23 (refer to **paragraph 1.8** *Wide Screen Signaling (WSS) and Video Program System (VPS))*

1: Automatic handling of the SINGLE_DATA_LINE_Register 98 is enabled;

At the start of the field the register 98 is set to 16 and after line 16 it is set to 23 (from the internal firmware), so that the reception of VPS and WSS is now possible in one field without support from the external controller.

0: Automatic handling is disabled. The SINGLE_DATA_LINE Register 98 is not touched by the firmware.

VPS_WAIT_CONTROL = MCI0_4 (REG_9)

VPS_ WAIT_ CTRL_7	VPS_ WAIT_ CTRL_6	VPS_ WAIT_ CTRL_5	VPS_ WAIT_ CTRL_4	VPS_ WAIT_ CTRL_3	VPS_ WAIT_ CTRL_2	VPS_ WAIT_ CTRL_1	VPS_ WAIT_ CTRL_0

VPS_WAIT_ CTRL (7:0) : defines the number of fields which have to be passed without the reception of VPS, before Megatext Plus automatically switches to TEXT reception in line_16. (see also bit *ENA_LINE16/23* which must be set to activate this function). This function can be reactivated every time when this command is executed.

NU_TE_FR = MCI0_5 (REG_8)

NU_TE_ FR_7	NU_TE_ FR_6	NU_TE_ FR_5	NU_TE_ FR_4	NU_TE_ FR_3	NU_TE_ FR_2	NUTE_ FR_1	NU_TE_ FR_0

NU_TE_FR (7:0) : defines the number of fields which have to be passed without text reception, before the flag *NO_TEXT* is set (see also indicator bits below).

Return Values

None

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1-255	Not defined

Comments

- Before using this command to switch on packet reception, the packet buffer must have been declared in the I²C packet buffer registers PB_ADR, PB_LENGTH and the I²C memory allocation registers IAT or XAT. **Don’t forget to reserve chapter 4 ... 17 in the XAT register for the temporary display memory.** These chapters cannot be used for normal page-acquisition. For good acquisition performance use search type 0 for normal page-acquisition. Search type 3 must be used if packet X/29 (magazine related data) should be evaluated.
- If packet reception is switched on, the packet buffer is initialized to 0, the ACQ is reset and then started.
- To ensure proper working of the ACQ, all ACQ-relevant input parameters must have been declared. For more details please refer to the document *ACQ Reference [2]*.
- To receive VPS data ensure that the I²C registers SINGLE_DATA_LINE and the EXTRA_FRAMING_WINDOW are set correctly or use the automatic option.
- If the automatic TOP_TABLE request option is used, **ACQ group 5** must be declared with a sub-code do care qualification. If memory overflow occurred during allocation of memory space, an interrupt is set.
- The check bits are automatically set so that additional information tables, multipage tables and multipage extension tables are checked in the way described in the TOP specification of the IRT.
- If one of the bits R_AI, R_MP or R_MPEX is set, after reception of the BTT the acquisition is searching for link information in the Basic-TOP-Table-List, allocates memory space and requests user-optionally the Additional-Information-TOP-Tables, Multipage-TOP-Tables or Multipage-Extension-TOP-Tables. The pages will always be requested in PRQ group 5. Before enabling this option, please ensure that the PRQ group record has been declared with search type 2. Use WRITE_GROUP for this command.
- The location of the PAGE_TRACE memory can be defined with the M3l registers 123, 124 and 125.
- If the automatic MOT_TABLE request option is used, **ACQ group 7** must be declared with search type 2 and a sub-code qualification (that means: 07, FF, 0F, 00, 00 hex). If a memory overflow occurred during allocation of memory space, an interrupt is set.
- If the automatic POP(DRCS)_TABLE request option is used, **ACQ group 4** must be declared with search type 2 and a sub-code qualification (that means: 07, FF, 0F, 00, 00 hex). If a memory overflow occurred during allocation of memory space, an interrupt is set.
- All POP and DRCS pages are requested in group 4.

Return Values

Following additional status informations are available in Megatext Plus. The status bits are reset by the firmware each time when this command is executed.

TEXT STATUS REGISTER Block0/Byte3/Row0/Col21

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TOP	FLOF	WSS	VPS	TEXT_16	NO_TEXT

NO_TEXT : 1 channel with no text in the given field_time
0 channel with text

TEXT_16 : 1 no VPS detection in the given field_time; switch to Text in Line_16
: 0 no reception of text in Line_16

VPS : 1 VPS information is available
0 VPS information is not available

WSS : 1 WSS information is available
0 WSS information is not available

FLOF : 1 packet_27/0000 is detected
0 packet_27/0000 is not detected

TOP : 1 Basic_TOP_Table is received
0 Basic_TOP_Table is not received

1.2.2 SERIAL_PARALLEL_CONVERSION (command no. 47d)

The additional features of the new S/P-C are the following:

- Automatic processing of all level 2.5 features described in the WST specification [3] (processing of X/26, X/27, X/28 and X/29 data)
- Flexible handling of all possible language combinations
- Non-latin character set support with national option and X/26 processing (cyrillic, greek, arabic, hebrew)
- Secondary language support (“twist”)
- Multiple OSD Windows in text with the help of the command INHIBIT_UPDATE

Input Parameters

SPC_MODE_2 = MCI2_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	STAT_ROW0	EN_STAT

EN_STAT This bit inhibits the update of row 24/0 by the S/P-C. If this bit is changed, the controller has to wait until the S/P-C is ready (see handshake bit below), before he executes the WRITE_ TOP_TITLE command.

1: Row 24/0 of the inner screen is disabled for the S/P-C.
0: Row 24/0 of the inner screen is enabled for the S/P-C.

STAT_ROW0 1: The status_line of the currently displayed chapter will be written by the S/P-C to the basic display memory row 0/column 0. Normal text starts on row 1/col 0.
0: The status_line of the currently displayed chapter will be written by the S/P-C to the basic display memory row 24/column 0. Normal text starts on row 0/column 0.

SPC_MODE_1 = MCI0_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ROLL_OFF	TIME_OFF	NEW_SUB	REVEAL	SEC_LA	PCS_CHAR	SPC_1	SPC_0

SPC_1, SPC_0 Controls S/P-C on/off switching for different page modes as shown in the following table:

SPC_1, SPC_0	Description of S/P-C Behavior
00	S/P-C is switched off.
01	Single page mode. Only the page which is selected by the basic display page registers will be converted. All higher level features are handled in the specified way. (including CLUTs and character designation codes).
10	S/P-C is switched off.

SPC_1, SPC_0	Description of S/P-C Behavior
11	S/P-C is switched on. Double page mode. Both pages which are selected by the basic and extended page registers will be converted. In the dual page mode there is an automatic fall-back to level 1.5.

PCS_CHAR This bit enables the S/P-C to use user-defined PCS characters instead of hardwired ROM characters of Megatext Plus. The user can exchange the complete G0 set and the G2 set or parts of them. These parts are defined by registers G0_WINDOW_START, G0_WINDOW_END, G2_WINDOW_START and G2_WINDOW_END. The equivalent PCS character set (or a part of it) of G0 (G2) must be downloaded between PCS addresses 20_H-$7F_H$ ($A0_H$-FF_H) before the S/P-C is started. The translation of national option characters is automatically switched off.

1: The PCS character set is enabled for the S/P-C
0: The PCS character set is disabled for the S/P-C. Only characters stored in the character_ROM are used.

REVEAL The REVEAL bit commands the S/P-C to reveal concealed characters. Concealed characters are shown as spaces.

1: Concealed characters are visible.
0: Concealed characters are not visible.

SEC_LA This bit decides about the source of the secondary language.

0: The G0-secondary language is selected by the S/P-C depending on the data of X/29 and X/28.
1: The secondary language must be initialized by the user (refer to *SLANG_6..0*).

NEW_SUB The NEW_SUB bit commands the S/P-C to either consider or ignore the page header control bits C5 (News Flash) and C6 (Sub-Title).

1: The S/P-C considers the control bits C5 and C6. If one or both are set to 1, the box mask register 1 (BOXMR_1 of display control words) and box display word 1 (BOXDW_1 of display control words) are used by the display generator. Otherwise BOXMR_0 and BOXDW_0 are used.
0: The S/P-C ignores the control bits C5 and C6. In this case BOXMR_0 and BOXDW_0 are used. In each case the S/P-C converts serial box on and box off control information into parallel for the affected characters in the basic display memory by setting the BX bit in the CDWs.

ROLL_OFF The ROLL_OFF bit commands the S/P-C to display the rolling header (Col. 8-31) or not to display it.

1: The S/P-C does not convert the page header to the display memory.
0: The S/P-C converts the page header to the display memory. Because the S/P-C refreshes the page header continuously, the result is a “rolling header”.

TIME_OFF The *TIME_OFF* bit commands the S/P-C to display the last 8 Bytes (Col 32-39) of the rolling header or not.

1: The S/P-C does not convert the time information to the display memory.
0: The S/P-C converts the time information to the display memory. Because the S/P-C refreshes the page header continuously, the result is a “rolling time”.

SPC_MODE_0 = MCI0_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	LEV_SEL_1	LEV_SEL_0	EN_CH_0_7	EN_16_9	EN_DRCS	PRIM_LA	EN_CLUT

EN_CLUT This bit forces the S/P-C to evaluate the CLUT features of level 2.5 of the WST specification [3].
CLUTS 2, 3 are updated by the S/P-C under control of the teletext-data.
CLUTS 0,1, 4, 5, 6 and 7 are user definable.

1: The S/P-C redefines the related CLUTs under control of the transmitted teletext data.
0: The S/P-C does not overwrite the CLUT.

PRIM_LA This bit decides about the source of the primary language.

1: The primary language is selected by the S/P-C, depending on the data of X/28 X/29 and the “c-Bit (C12, C13, C14)”.
0: The primary language must be initialized by the user (refer to *PLANG_6..0)*.

EN_DRCS Enables the automatic download of upto 24 DRCS characters under control of teletext data (Level 2.5). For allocation of DRCS memory see the comments.

1: The DRCS Feature is enabled.
0: The DRCS-Feature is disabled. All of the specified DRCS memory is free for the user.

EN_16_9 Enables the 16:9 feature of the WST specification [3]. That means the “side panels” are written by the S/P-C in the way specified in the WST [3]. The appropriate memory cannot be used for OSD menues, except if the INHIBIT UPDATE command is used. The S/P-C also influences the display position word according to the parameters given by packet X/28. The original display position is saved in a temporary register and restored, if a page without sidepanels should be displayed.

1: The 16:9 Feature is enabled.
0: The 16:9 Feature is disabled.

EN_CH_0_7 This bit enables the S/P-C to write bit 9-bit 40 of column 0-7 concerning to the information given by the broadcaster.

1: The attributes of the first 8 characters are updated by the S/P-C.
0: Bit 9 - Bit 40 are not updated by the S/P-C.

LEV_SEL_1:0 This bits selects the teletext level which should be processed by the internal S/P-C.

00:Level 1 is selected
01:Level 1.5 is selected
10:Level 2.5 is selected
11:Not defined

Window Definition for the PCS Character Set

(refer to the bit *PCS_CHAR* in spc_mode1)

G0_WINDOW_START = MCI0_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
G0_WIN_ S7	G0_WIN_ S6	G0_WIN_ S5	G0_WIN_ S4	G0_WIN_ S3	G0_WIN_ S2	G0_WIN_ S1	G0_WIN_ S0

G0_WIN_S_7 through G0_WIN_S_0 define the lowest address of the G0 set to be substituted by PCS characters. This value must be equal or bigger than 20_H.

G0_WINDOW_END = MCI0_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
G0_WIN_ E7	G0_WIN_ E6	G0_WIN_ E5	G0_WIN_ E4	G0_WIN_ E3	G0_WIN_ E2	G0_WIN_ E1	G0_WIN_ E0

G0_WIN_S_7 through G0_WIN_S_0 define the highest address of the G0 set to be substituted by PCS characters. This value must be equal or smaller than $7F_H$.

G2_WINDOW_START = MCI0_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
G2_WIN_ S7	G2_WIN_ S6	G2_WIN_ S5	G2_WIN_ S4	G2_WIN_ S3	G2_WIN_ S2	G2_WIN_ S1	G2_WIN_ S0

G2_WIN_S_7 through G2_WIN_S_0 define the lowest address of the G2 set to be substituted by PCS characters. This value must be equal or bigger than 20_H.

G2_WINDOW_END = MCI0_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
G2_WIN_ E7	G2_WIN_ E6	G2_WIN_ E5	G2_WIN_ E4	G2_WIN_ E3	G2_WIN_ E2	G2_WIN_ E1	G2_WIN_ E0

G2_WIN_S_7 through G2_WIN_S_0 define the highest address of the G2 set to be substituted by PCS characters. This value must be equal or smaller than $7F_H$.

Display Page Parameters

These registers define the address pointer of the display page which will be converted into the basic display memory (internal RAM block 2). Use the “SEARCH_PAGE” command to get the address pointer. The half of the screen can be selected with the bit *BES* of the page position word (PPW) display control register.

Address pointer to the basic display page (this is the pointer to the page to be displayed).

For detail information about the address format please refer to document *M3l-Bus Register* [2] **paragraph 2.4**.

AP_CHAP_2 = MCI1_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EX = 0 EX = 1	0	0 CHP_10	0 CHP_9	BYT5 CHP_8	BYT4 CHP_7	BYT3 CHP_6	BYT2 CHP_5

AP_CHAP_1 = MCI1_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BYT1 CHP_4	BYT0 CHP_3	BLK_2 CHP_2	BLK_1 CHP_1	BLK_0 CHP_0	0	0	0

AP_CHAP_0 = MCI1_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

Extended Display Page Registers

These registers define the address pointer of the display page which will be converted into the extended display memory (internal RAM block 3). Use the “SEARCH_PAGE” command to get the address pointer. Selection on which half of the screen the page appears can be done by bit *BES* of the page position word (PPW) display control register.

Address pointer to the extended display page (this is the pointer to the page to be displayed).

For detail information about the address format please refer to document *M3l-Bus Register* [2] **paragraph 2.4**.

AP_EXCHAP_2 = MCI1_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EX = 0 EX = 1	0	0 CHP_10	0 CHP_9	BYT5 CHP_8	BYT4 CHP_7	BYT3 CHP_6	BYT2 CHP_5

AP_EXCHAP_1 = MCI1_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BYT1 CHP_4	BYT0 CHP_3	BLK_2 CHP_2	BLK_1 CHP_1	BLK_0 CHP_0	0	0	0

AP_EXCHAP_0 = MCI1_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

Language Selection

PRIMARY_LANGUAGE_SELECTION = MCI2_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	LANG_6	LANG_5	LANG_4	LANG_3	LANG_2	LANG_1	LANG_0

LANG_6 – LANG_0 Defines the language_number which should be used for the G0 set. (Related bit *PRIM_LA*)

PRIMARY_LANGUAGE_SELECTION = MCI2_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SLANG_6	SLANG_5	SLANG_4	SLANG_3	SLANG_2	SLANG_1	SLANG_0

SLANG_6 – SLANG_0 Defines the secondary language_number which should be used for G0-set. (Related bit *SEC_LA*)

Return Values

None

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1-255	Not defined

Comments

- After invoking the command SERIAL_PARALLEL_CONVERSION with SPC_1, SPC_0 = 00 (S/P-C_off), there is a delay until the last S/P-C cycle will be finished (refer to *SPC_READY bit in SPC status register)*. If you want to stop the S/P-C by setting the *SPC_0/SPC_1* to “0”, it is also necessary to define the rest of the input parameters with valid data.
- The User Defined Characters (UDC), Row 0/Column 0-7/Bytes 0-5 of the display memory, are under control of the external controller. That means that the user has to write the current page number in the desired format directly to the display memory. Only if the bit *EN_CH_0_7* is set, the S/P-C will change the background_color if there is a level 2.5 attribute like full screen color or full row color.
- Primary and secondary language definition:
  If the bit *PRIM_LA* is set, the language is taken from the language information of the header (C14-C12), or if transmitted from the appropriate parameter of packet X/28 or

X/29. The assignment of the language to the bits C14-C12 in each header is defined in the last 8 entries of the *language definition table* (see second table below). This character set definition is used from the internal firmware if there is no additional information via packets_X/28, X/29.

The *language definition table* **must** be initialized by the external controller in the given memory location before the S/P-C is enabled. The *language definition table* is available as an “SDO” file (*language.sdo*). Please contact your Siemens representative.

The secondary language is always taken from packet X/28 or X/29. If none of these packets are transmitted the secondary language is identical to the primary language.

The following table shows the assignment of the supported languages to the internal language code. This code must be used to define a language in the language definition table. If a non latin language is selected, the appropriate character set must be downloaded into the PCS memory.

Internal Language Code Table

Language	Internal Language Code (hexadecimal)
English	00
German	01
Swedish/Finnish	02
Italian	03
French	04
Portuguese/Spanish	05
Czech/Slovak	06
Polish	07
Serbian/Croat/Sloven	08
Rumanian	09
Turkish	0A
Estonian	0B
Lithuanian/Lettish	0C
Hungarian	0D
Danish	0E
South African	0F
ASCII	10
Undefined latin language	11
Undefined latin language	12

Internal Language Code Table (cont'd)

Language	Internal Language Code (hexadecimal)
Undefined latin language	13
Undefined latin language	14
Undefined latin language	15
Undefined latin language	16
Undefined latin language	17
Undefined latin language	18
Undefined latin language	19
Russian/Byelorussian	20
Ukranian	21
Greek	22
Albanian	23
English G0 / Arabic G2	24
French G0 / Arabic G2	25
Arabic	26
Hebrew G0 / Arabic G2	27
Serbian/Croatian Cyrillic	28
Reserved	29-3F

Language Definition Table

External Memory Binary Address	External Memory Chap/ Row/Col	Data (hexadecimal) Internal Language Code	Language_# according to the Enhanced Teletext Spec (decimal)	Language
808808	17/00/04	00	0	English
80880A	17/00/05	01	1	German
80880C	17/00/06	02	2	Swedish/Finnish
80880E	17/00/07	03	3	Italian
808810	17/00/08	04	4	French
808812	17/00/09	05	5	Portuguese/Spanish
808814	17/00/10	06	6	Czech/Slovak
808816	17/00/11	12	7	Reserved
808818	17/00/12	07	8	Polish
80881A	17/00/13	01	9	German
80881C	17/00/14	02	10	Swedish/Finnish
80881E	17/00/15	03	11	Italian
808820	17/00/16	04	12	French
808822	17/00/17	12	13	Reserved
808824	17/00/18	06	14	Czech/Slovak
808826	17/00/19	12	15	Reserved
808828	17/00/20	00	16	English
80882A	17/00/21	01	17	German
80882C	17/00/22	02	18	Swedish/Finnish
80882E	17/00/23	03	19	Italian
808830	17/00/24	04	20	French
808832	17/00/25	05	21	Portuguese/Spanish
808834	17/00/26	0A	22	Turkish
808836	17/00/27	22	23	Greek
808838	17/00/28	12	24	Reserved
80883A	17/00/29	23	25	Albanian
80883C	17/00/30	0D	26	Hungarian

Language Definition Table (cont'd)

External Memory Binary Address	External Memory Chap/ Row/Col	Data (hexadecimal) Internal Language Code	Language_# according to the Enhanced Teletext Spec (decimal)	Language
80883E	17/00/31	12	27	Reserved
808840	17/01/00	06	28	Slovakian
808842	17/01/01	12	29	Reserved
808844	17/01/02	08	30	Serbian/Croat/Sloven
808846	17/01/03	09	31	Rumanian
808848	17/01/04	28	32	Serbian/Croatian Cyril
80884A	17/01/05	01	33	German
80884C	17/01/06	0B	34	Estonian
80884E	17/01/07	0C	35	Lithuanian/Lettish
808850	17/01/08	20	36	Russian/Byelorussian
808852	17/01/09	08	37	Serbian/Croatian latin
808854	17/01/10	06	38	Czech/Slovak
808856	17/01/11	21	39	Ukrainian
808858	17/01/12	12	40	Reserved
80885A	17/01/13	12	41	Reserved
80885C	17/01/14	12	42	Reserved
80885E	17/01/15	12	43	Reserved
808860	17/01/16	12	44	Reserved
808862	17/01/17	12	45	Reserved
808864	17/01/18	12	46	Reserved
808866	17/01/19	12	47	Reserved
808868	17/01/20	12	48	Reserved
80886A	17/01/21	12	49	Reserved
80886C	17/01/22	12	50	Reserved
80886E	17/01/23	12	51	Reserved
808870	17/01/24	12	52	Reserved
808872	17/01/25	12	53	Reserved

Language Definition Table (cont'd)

External Memory Binary Address	External Memory Chap/ Row/Col	Data (hexadecimal) Internal Language Code	Language_# according to the Enhanced Teletext Spec (decimal)	Language
808874	17/01/26	0A	54	Turkish
808876	17/01/27	22	55	Greek
808878	17/01/28	27	56	Hebrew
80887A	17/01/29	12	57	Reserved
80887C	17/01/30	12	58	Reserved
80887E	17/01/31	12	59	Reserved
808880	17/02/00	12	60	Reserved
808882	17/02/01	12	61	Reserved
808884	17/02/02	12	62	Reserved
808886	17/02/03	12	63	Reserved
			C14 C13 C12	
808888	17/02/04	00	0 0 0	English
80888A	17/02/05	04	0 0 1	French
80888C	17/02/06	02	0 1 0	Swedish/Finnish
80888E	17/02/07	06	0 1 1	Czech/Slovak
808890	17/02/08	01	1 0 0	German
808892	17/02/09	05	1 0 1	Portuguese/Spanish
808894	17/02/10	03	1 1 0	Italian
808896	17/02/11	12	1 1 1	Reserved

- The S/P-C uses chapter 4 to chapter 17 of the external memory as a temporary memory.
- If the dual page mode is selected, the sync and row attribute registers must be set in the appropriate way by the user. In this mode only Level_1.5 is supported.
- The following table explains the DRCS memory allocation:

12*10*1 Address	USAGE	Number of Characters
0_H-17_H	Transmitted level 2.5 characters	24

18_H-$1F_H$	OSD	8
20_H-$7F_H$	G0 DRCS character set	96
80_H-$9F_H$	OSD	32
$A0_H$-FF_H	G2 DRCS character set	96

- The following S/P-C status bits are available at address Block0/Byte5/Row6/Col16

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	0	0	0	0	SIDE_ PANEL	S/P-C_ Ready	Outer_ Screen

SIDE_PANEL This bit tells whether a side panel information for the current display page is received.

If a sidepanel is transmitted, the external controller has to increase the pixel speed using the *16_by_9* bit in I²C register 114. The position of the display has to be corrected with the *Sync_Delay_Word*.

0: No side_panel is activated for the current display page.

1: Side_panel is activated for the current display page.

S/P-C_Ready The *S/P-C_READY* bit is reset by the command SERIAL_PARALLEL_CONVERSION and set from the internal firmware if the current S/P-C cycle is finished. This bit is very helpful if you want to stop the S/P-C to use the display memory for other purpose. In this case you have to wait until the *SPC_READY* bit is set again before you start writing into the display memory.

Outer_Screen This bit can be used as an indicator, whether the outer-screen is under control of teletext or not. This might be useful if the user wants to define this area by the outer screen mask register.

0: Outer screen is not used by the S/P-C (only black blanks are stored in that area).

1: Some high level information (side panel or screen background color) is stored in that area. The outer screen mask register should not be used by the controller.

1.2.3 INHIBIT_UPDATE_S/P-C (command no. 10d)

This command defines display memory parts which can not be overwritten by the S/P-C. Any number of inhibit update windows can be defined. In between these windows bytes 0-4 of the character display word (CDW) are blocked for the S/P-C. The window can be closed again by using the bit *UPDATE*. Before starting the S/P-C the first time this command should be given with parameters *ALL* and *UPDATE* set to “1”.

Input Parameters

MODE_1 = MCI0_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	IG_WIN_DEF	ACT_WIN	BLOCK_2_3	ALL	UPDATE

IG_WIN_DEF This bit is only relevant if you have defined different inhibit windows in the background, which means the whole display memory is still refreshed by the S/P-C. After all settings are done they can be enabled/disabled with one command. In this case only the bits *IG_WIN_DEF* and *ACT_WIN* must be defined. All other input parameters will be ignored.

1: Only the bit *ACT_WIN* will be considered. All other input parameters are not relevant.

0: All input parameters will be considered as described.

ACT_WIN This bit forces the S/P-C to process all window definitions.

HINT: if no windows are defined, *ACT_WIN* should be set to “0”.

1: During the refresh of the display memory, the S/P-C considers the window definitions. The defined windows will not be updated from the S/P-C anymore.

0: The S/P-C does not process the window definitions any more. All window settings are still in the background but the whole display memory will be refreshed by the S/P-C again.

BLOCK_2_3 These bits select whether the command is valid for block 2 (inner screen area) or block 3 (outer screen area) of the display memory.

1: Block 3 is affected by this command.

0: Block 2 is affected by this command.

ALL This bit forces the S/P-C to update the specified window again with teletext data.

1: The total memory of block 2 / 3 is influenced by this command.

0: Only the window defined by the COORDINATE registers is influenced by this command.

UPDATE This bit forces the S/P-C to update the specified window again with teletext data.

1: The specified inhibit update window is canceled.

0: The specified window is not updated any more by the S/P-C.

Window Coordinate

COORDINATE_ROW_START = MCI0_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	S_ROW_4	S_ROW_3	S_ROW_2	S_ROW_1	S_ROW_0

COORDINATE_COLUMN_START = MCI0_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	S_COL_4	S_COL_3	S_COL_2	S_COL_1	S_COL_0

COORDINATE_ROW_END = MCI0_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	E_ROW_4	E_ROW_3	E_ROW_2	E_ROW_1	E_ROW_0

COORDINATE_COLUMN_END = MCI0_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	E_COL_4	E_COL_3	E_COL_2	E_COL_1	E_COL_0

The COORDINATE registers define the start and end row and column address of the inhibit update windows. The end address must always be bigger than the start address.

Return Values

None

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1-255	Not defined

Comments

None

1.2.4 WRITE_TOP_TITLE (command no. 53d)

Because all bits of the character display word are used in level 2.5 teletext, they must also be defined in the TOP title. That means, that now 5 bytes are available for defining the TOP title attributes.

Input Parameters

Display_memory_address (this is the display destination pointer to which the TOP title is to be written).

ROW_COL_POSITION_2 = MCI0_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	1	1	1

ROW_COL_POSITION_1 = MCI0_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	BLK_1	BLK_0	ROW_4	ROW_3	ROW_2

ROW_COL_POSITION_0 = MCI0_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0

COL_5 thru COL_0 Column address of first TOP title character to be written.

ROW_4 thru ROW_0 Row address of first TOP title character to be written.

BLK_1, BLK_0 Selects display memory to which TOP title is to be written.

MODE_1 = MCI0_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	APPEND_ CHAR	FILL_ BLANKS	READ	CUT_ BLANKS	PAGE_ NUM_ENA

PAGE_NUM_ENA

1: If no additional information to the page number specified in PAGE_NUMBER is present, this page number will be changed to ASCII format and written to the position specified in ROW_COL_POSITION.

0: If no additional information is present, nothing will be written to the display memory. If additional information is present it will be written into the position specified in ROW_COL_POSITION.

CUT_BLANKS

1: Blanks behind the last TOP title character to be written will be truncated.

0: Blanks behind the last TOP title character to be written will not be truncated.

READ

1: Only the additional information is returned. Nothing will be written in the display memory.

0: The TOP title will be written to the specified address and the additional information will be returned.

FILL_BLANKS

1: Instead of a page number or a TOP title, blanks will be written to the specified address. If this bit is set, all other mode bits are ignored. The number of blanks to be written is given in NUMBER_OF_BLANKS. The attributes of these blanks are given in TOP_TITLE_ATTRIBUTES.

0: FILL_BLANKS is switched off.

APPEND_CHAR

1: A character specified by TOP_TITLE_CHARACTER appended to the written page number or TOP title.

0: APPEND_CHARACTER is switched off.

Page Number

This is the page number for which the TOP title should be written.

PAGE_NUMBER_1 = MCI0_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	M2	M1	M0

PAGE_NUMBER_0 = MCI0_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0

All above listed page number bits have the same meaning as defined in the world system teletext specification.

Number of Blanks

This register defines the number of blanks to be written.

NUMBER_OF_BLANKS = MCI1_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	NU_5	NU_4	NU_3	NU_2	NU_1	NU_0

NU_5 thru NU_0 value must be between 0-39d

Top Title Attributes

These registers define the character attributes for the TOP title string to be written.

TOP_TITLE_ATTRIBUTE_1 = MCI1_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

TOP_TITLE_ATTRIBUTE_1 = MCI1_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TT_31	TT_30	TT_29	TT_28	0	0	0	0

TOP_TITLE_ATTRIBUTE_1 = MCI1_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TT_23	TT_22	TT_21	TT_20	TT_19	TT_18	TT_17	TT_16

TOP_TITLE_ATTRIBUTE_0 = MCI1_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TT_15	TT_14	TT_13	0	0	0	TT_9	0

All above listed TOP title attribute bits have the same meaning as defined for character display words.

TOP Title Characters

This register defines the character to be appended to a written TOP title.

TOP_TITLE_CHARACTER = MCI1_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

With *CH7 thru CH0* any character from the Megatext Plus character set can be selected.

Return Values

Additional Information

The first 8 bytes (magazine number, page number tens, page number units, link information, direct selection) of the given page number are returned. For details of these bits refer to the TOP specification [1]. The additional information consists of 8 bytes ADDI_BYTE_7 thru ADDI_BYTE_0 with ADDI_BYTE_0 as the least significant. The FALSE bit indicates whether the byte has been received correctly.

ADDI_BYTE_7 = MCI2_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_7_3	AI_7_2	AI_7_1	AI_7_0

ADDI_BYTE_6 = MCI2_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_6_3	AI_6_2	AI_6_1	AI_6_0

ADDI_BYTE_5= MCI2_3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_5_3	AI_5_2	AI_5_1	AI_5_0

ADDI_BYTE_4 = MCI2_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_4_3	AI_4_2	AI_4_1	AI_4_0

ADDI_BYTE_3 = MCI2_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_3_3	AI_3_2	AI_3_1	AI_3_0

ADDI_BYTE_2 = MCI2_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_2_3	AI_2_2	AI_2_1	AI_2_0

ADDI_BYTE_1 = MCI3_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_1_3	AI_1_2	AI_1_1	AI_1_0

ADDI_BYTE_0 = MCI3_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	FALSE	0	AI_0_3	AI_0_2	AI_0_1	AI_0_0

The *ADDI* bits have the same meaning as described in the document “TOP-System for Teletext” [1].

Next Column Address

This is the column address after the last written TOP title character.

ROW_COL_POSITION_2 = MCI0_2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	1

ROW_COL_POSITION_1 = MCI0_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	BLK_1	BLK_0	0	0	0

ROW_COL_POSITION_0 = MCI0_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0

BLK_1, BLK_0 Block address of the TOP title character last written.

ROW_4 thru ROW_0 Row address of the TOP title character last written.

COL_5 thru COL_0 Column address behind the TOP title character last written. If this address is equal to the column address in ROW_COL_POSITION_0 and CUT_BLANKS = 1, the TOP title string consists only of blanks.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Basic TOP table not received.
2	Additional information not found.
3-255	Not defined.

Comment

Before the additional information table is received for the first time, all bytes of this chapter are automatically set to 20_H.

1.2.5 READ_TRACE_REQUEST_TABLE (command no. 52d)

This command is introduced to optimize the performance of the complete system. In many cases it is necessary to know how many pages are in the transmitted cycle (page trace tab) or how many pages are in progress (to be in progress tab). In former versions this had to be programmed externally and therefore was very slow.

For further information about the tables refer to command *Read_Clear_Page_Trace* [2].

Input Parameters

TABLE_SELECT = MCI0_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	To_be_ intended	To_be_in_ progress	Sub_Page_T race	Page_ Trace

With the bits in this register one of the four tables is selected. The number of all set bits in the selected table is returned. Only one of these bits may be set at the same time.

Return Values

NUMBER_OF_BITS_1 = MCI1_1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	NUM_10	NUM_9	NUM_8

NUMBER_OF_BITS_0 = MCI1_0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NUM_7	NUM_6	NUM_5	NUM_4	NUM_3	NUM_2	NUM_1	NUM_0

NUM_10 ... NUM_0 The number of bits which are set in the selected table are returned.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1-255	Not defined.

Comments

None

1.3 Changes in the CLUT of the Megatext Plus

In the SDA 5273, CLUT 0 and 1 (vectors 0-15) are hardwired. In the SDA 5275 the values of these vectors must be programmed. **To be compatible with the SDA 5273 these values have to be initialized by the external controller.** The positioning of the CLUT vectors in the memory is already defined in document *Display Functions* [2].

Refer to **paragraph 2.5** Application Notes in this document.

1.4 Changes in the DRCS-Addressing of the Megatext Plus

The memory for DRCS characters has been essentially increased in the SDA 5275. To address these additional DRCS characters, the character display word is expanded to 43 bits. That means that also the screen mask registers (OSMR, OSDW, BOXMRO, BOXDW0, ISMR0, ISDW0, BOXMR1, BOXDW1, ISMR1, ISDW1) are expanded to 43 bits.

The new CDW looks like:

Byte Pos.	Bit	Name	Function	Remark
0	0 1 2 3 4 5 6 7	B0 B1 B2 B3 B4 B5 B6 B7	ROM character select	Each character is defined by a 12 × 10 pixel matrix
1	8	B8		
	9	US	Underline/Separate graphic	Function depends on special character
	10	UH	Upper Half double height	
	11	DH	Double Height	
	12	DW	Double Width	Marks left half of character
	13	CO	Conceal/Reveal	
	14	TRB	Transparent Background	Video picture visible
	15	TRF	Transparent Foreground	

Byte Pos.	Bit	Name	Function	Remark
2	16	BX	Box Mode	
	17 18 19	BC0 BC1 BC2	Selection out of 8 Background colors	Color vector
	20 21 22	FC0 FC1 FC2	Selection out of 8 Foreground colors	Color vector
	23	F0	Control of Flash modes	
3	24 25 26	F1 F2 F3		
	27	IC	Inverse colors	May be used as cursor
	28 29	BC3 BC4	CLUTselect for Background color	
	30 31	FC3 FC4	CLUT select for Foreground color	
4	32 33 34 35 36	DD0 DD1 DD2 DD3 DD4	Multimode bits Addressing of accents Addressing of PCS Memory Selection of the DRCS mode	
	37 38	DM0 DM1	Display Mode selection	
	39	UC	User CLUT select	Selects CLUT 0:3 or CLUT 4:7
5	40 41	B9 B10	Additional DRCS-Addresses	
	42	B11	Must be set to “0”	
	43 - 47		Reserved for future use	Not used for now

Bits 40 and 41 can be used for addressing the additional DRCS characters. Bit 42 in all display mask registers must be set to “0”. Not all of the possible 4096 addresses are defined. The following table shows how many DRCS characters can be defined depending on the selected mode:

PCS Resolution	Number of Possible DRCS Characters	Location in Memory
12 × 10 × 1	256	Block 0 / row 8-24
	400	Block 1 / row 0-24
	400	Block 3 / row 0-24
12 × 10 × 2	128	Block 0 / row 8-24
	200	Block 1 / row 0-24
	200	Block 3 / row 0-24
12 × 10 × 4	64	Block 0 / row 8-24
	100	Block 1 / row 0-24
	100	Block 3 / row 0-24

1.5 Reveal/Conceal

The function reveal/conceal can now be realized very simple by setting/resetting of the *REVEAL* bit in the input parameters of the command *SERIAL PARALLEL CONVERSION* (see **paragraph 1.2.2** in this document).

1.6 User Defined Characters (UDC)

UDC = The first 8 characters (column 0-7) in row 0 of the display memory (e.g. pagenumber).

The processing of the UDC has changed. With Megatext SDA 5273, the UDC must be initialized in the internal memory block 0. With Megatext Plus, the UDC must be initialized directly into block 2 (display memory). Refer to **paragraph 1.2.2** comments in this document.

1.7 Box Bit

The use of the box bit in the character display word is inverted by the S/P-C compared to the Megatext SDA 5273. Therefore it is necessary to exchange the contents of the following display registers to get a correct display for newsflash and subtitle pages:

Inner_Screen_Display_Register_1<---> Box_Display_Word_1

Inner_Screen_Mask_Register_1<---> Box_Mask_Register_1

SDA 5273	**SDA 5275**	
BX = 1	BX = 0	Inside box
BX = 0	BX = 1	Outside box

This change was necessary to handle the new WST [3] feature “transparent” (color 8).

1.8 Wide Screen Signaling (WSS) and Video Program System (VPS)

The SDA 5275 has an integrated single data line module for realtime WSS and VPS processing. If this module is enabled, the internal PU takes over the control of the M3l-Bus Register 98 SINGLE_DATA_LINE. The single data line module switches automatically between line 16 and line 23 in one field. If the SINGLE_DATA_LINE register is initialized to line 16 by the external controller before the command *ACQ_CONTROL*, the line 16 processing (text line or VPS line) is also part of this module (see text status register below). The WSS data are error checked and written to a buffer in the internal memory. Their actual values can be read at any time. The received data are biphase decoded and only written if the whole WSS line is received without error. Clock-run-in and framing-code are not stored. To enable the WSS/VPS module initialize the registers *High_Level_Control_2, VPS_WAIT_CONTROL, NU_TE_FR* of the command *ACQ_CONTROL* (refer to item 2.1 ACQ_CONTROL) and initialize the M3l-Register EXTRA FRAMINGCODE WINDOW (refer to the document *M3l-Bus Register* [2]).

WSS

Block0/Byte0/Row1/Col0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WSS_7	WSS_6	WSS_5	WSS_4	WSS_3	WSS_2	WSS_1	WSS_0

Block0/Byte0/Row1/Col

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	WSS_13	WSS_12	WSS_11	WSS_10	WSS_9	WSS_8

WSS(13:0) data bits of the transmitted WSS information. Refer to WSS Specification [4]

WSS(3:0)	-> aspect ratio
WSS(7:4)	-> enhanced service
WSS(10:8)	-> subtitles
WSS(13:11)	-> reserved

TEXT STATUS REGISTER Block0/Byte3/Row0/Col21

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TOP	FLOF	WSS	VPS	TEXT_16	NO_TEXT

The WSS status bit is an indicator for the reception of a valid WSS packet. Before reading of the WSS data set the WSS status bit in the text status register to “0”. Use a polling technic until the WSS status bit is “1”.
(for more information about the text status register refer to **paragraph 2.1** ACQ_CONTROL in this document)

VPS

Block0/Byte2/Row1/Col0-13

Column Position	VPS Word	VPS Data Word
0	3	Source identification
1	4	Source identification
2	5	**Sound data special identification**
3	6	Signal content idetification
4	7	ASCII plain text channel
5	8	Routing
6	9	Routing
7	10	Reports commands
8	11	**VPS extra information**
9	12	**VPS extra information**
10	13	**VPS extra information**
11	14	**VPS extra information**
12	15	Reserved for data protection
13		Status word

Bit-Resolution for the VPS Data Words

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	VPS_3	VPS_2	VPS_1	VPS_0

The VPS data are error checked and written to a buffer in the internal memory. Their current values can be read any time. The received data are biphase decoded and only written if the whole VPS line is received without error. The status word shows the number of detected errors. Clock-run-in and framing-code are not stored. For further information about VPS refre to the VPS Specification [5].

TEXT STATUS REGISTER Block0/Byte3/Row0/Col21

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TOP	FLOF	WSS	VPS	TEXT_16	NO_TEXT

The VPS status bit is an indicator for the reception of a valid VPS packet. Before reading of the VPS data set the VPS status bit in the text status register to “0” . Use a polling technic until the VPS status bit is “1”.
(for more information about the text status register refer to **paragraph 2.1** ACQ_CONTROL in this document)

2 Application Notes

2.1 CNN Page 7FE/0000

Problem

The CNN channel is using the page number 7FE/0000 for a special data service.

The page number xFE is now reserved in the new teletext specification for the magazine organisation table MOT. Therefore the SDA 5275 interpretes the page 7FE/0000 as a MOT. This can create rubbish displays and unneeded memory allocation for dummy links (POP, DRCS).

Workaround

We found out that CNN is only using the subcode 0000 for this data service. This subcode (0000) is a forbidden code for MOT as defined in the new teletext specification. To avoid the mentioned problem, it is necessary to block this page (for the MOT processing). This can be done in the following way :

- Use a search group with a higher priority as the MOTgroup_7 to collect the page 7FE/0000, e.g. use search group_0 and set the whole page number and subpage number in the do_care_mask of this group to do_care.
- Set a page request for the page 7FE/0000 in this group with the command add_page.

The internal firmware interpretes only the pages stored in search group_7 as MOT's. Now page 7FE/0000 will not be stored in this group and the MOT processing can run without problems.

2.2 Version Code Overview

Version code overview for

- Megatext SDA 5273
- Megatext Plus SDA 5275
- Compacttext SDA 5273C

In the internal memory one location is reserved for the version code. This version code can be used to distinguish the above mentioned ICs.

IC Differentiation

Memory Address	SDA 5273	SDA 5275	SDA 5273C
Block_0 **Byte_4** Row_7 Column_23	00_H	01_H	02_H

Version Differentiation

Memory Address	SDA 5275		SDA 5273C	
	Version	Version Code	Version	Version Code
Block_0	A23	22_H	C29	11_H
Byte_3	B11	22_H	C129	12_H
Row_7	B12	22_H	C229	12_H
Column_23				

Version Differentiation

Version	SDA 5273					
	Memory Address					
	mci0_5	mci0_4	mci0_3	mci0_2	mci0_1	mci0_0
C22	32_H	35_H	30_H	38_H	39_H	33_H
C24	31_H	34_H	30_H	33_H	39_H	34_H
C26	31_H	39_H	30_H	37_H	39_H	34_H
C134	33_H	30_H	30_H	35_H	39_H	35_H

Hint mci0_5 to mci0_0 are equal to M3l-Bus registers reg8 to reg13. The version code in these M3l-Bus registers is valid after Megatext is reset until the first mci command is given.

2.3 Termination Display Word (TDW)

Pages with sidepanels and a background colour unequal to black show a vertical black bar on the left or right side of the display. This area is defined by the TDW.

Proposal: Initialize the TDW with a space and background color 8. The borders will then automatically have the screen background color.

TDW Block0/Row3/Col5

Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
00_H	00_H	50_H	00_H	00_H	20_H

2.4 M3l-Register 113 Sync Source Selection

In the sync source selection register the most significant bit (bit 7) must be set to “1” otherwise the right edge of the display can be incorrect if the 16 by 9 mode is used.

This bit has no effect in the Megatext SDA 5273 and also in the 4 by 3 mode.

2.5 Colour Look Up Table

In the Megatext SDA 5273 the CLUTs 0 and 1 are hardwired. In the Megatext Plus SDA 5275 the CLUTs 0 and 1 must be initialized by the external controller. The following table shows the values for the CLUTs 0 and 1 which have to be initialized by the external controller :

MEMORY MAPPING

Memory Address	Byte 5		Byte 4		Byte 3		Byte 2		Byte 1		Byte 0	
	7654	3210	7654	3210	7654	3210	7654	3210	7654	3210	7654	3210
	CLUT 0 / Color 2			**CLUT 0 / Color 3**			**CLUT 0 / Color 4**			**CLUT 0 / Color 5**		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Block0/Row3/Col16	0000	1111	0000	0000	1111	1111	1111	0000	0000	1111	0000	1111
	CLUT 0 / Color 6			**CLUT 0 / Color 7**			**CLUT 1 / Color 8**			**CLUT 1 / Color 9**		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Block0/Row3/Col17	1111	1111	0000	1111	1111	1111	0000	0000	0000	0000	0000	0111
	CLUT 1 / Color 10			**CLUT 1 / Color 11**			**CLUT 1 / Color 12**			**CLUT 1 / Color 13**		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Block0/Row3/Col18	0000	0111	0000	0000	0111	0111	0111	0000	0000	0111	0000	0111
	CLUT 1 / Color 14			**CLUT 1 / Color 15**			**CLUT 0 / Color 0**			**CLUT 0 / Color 1**		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Block0/Row3/Col19	0111	0111	0000	0111	0111	0111	0000	0000	0000	0000	0000	1111

2.6 Multi Language Processing

2.6.1 Non-Latin Character Set Selection via Packets X/28, X/29

With the new level 2.5 teletext standard it is possible to define a new G0/G2 character set via packets X/28, X/29. This means that each page in the transmitted cycle can be defined with its own character set. Megatext Plus supports this feature in the following way:

All latin-based character sets are integrated in the character ROM and will be processed automatically by the internal firmware. For all non-latin-based character sets the internal PCS memory can be used. Supported PCS character sets are cyrillic, arabic, greek, hebrew. These character sets can be ordered from your Siemens representative and are available as an “SDO”-file. Due to the fact that it is only possible to download one PCS character set at a time into the Megatext Plus, it is necessary to know which character set will be needed for the actual display page. The register described below is the interface between the internal S/P-C, which processes the character set, and the external controller, which has to download the right PCS character set.

CHARACTER SET CONTROL REGISTER Block0/Byte2/Row6/Col16

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HAND_SHAKE	INT	LANG_5	LANG_4	LANG_3	LANG_2	LANG_1	LANG_0

HAND_SHAKE : Handshake bit
If the internal firmware detects that a non-latin character set should be used for the current display page, this bit will be set to “1”. The external controller can use a polling technique in the mainloop to detect when this bit goes into the “1” condition. If this bit is in the “1” condition, the bits *LANG (5:0)* define the language that is needed for the actual display page (refer to the *internal language code table* **paragraph 1.2.2** *Serial Parallel Conversion* in this document). If this character set is available and supported by the external controller, it must be downloaded into the PCS memory of Megatext Plus. After the download, the bit *HAND_SHAKE* must be reset by the external controller, to indicate to the S/P-C, that the download is finished and the new character set can be used.

LANG(5:0) : Internal language code (refer to the *internal language code table* **paragraph 1.2.2** *Serial Parallel Conversion*).
The S/P-C returns in these bits the requested language via packet X/28 and X/29 for the current display page.

INT : This bit is for internal use only.

2.6.2 Example for Russian Market

This example shows the decisive differents in the initialisation of Megatext Plus for the russian market.

2.6.2.1 Input Parameter for the Command *Serial_Parallel_Conversion*

SPC_MODE_0 = MCI0_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	x	x	x	x	x	1	x

SPC_MODE_1 = MCI0_5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	x	x	1	x	x	x

The bit *SEC_LA* should be set and the secondary language should be defined to english because currently, no russian broadcaster transmits packet X/28 or X/29.

SECONDARY LANGUAGE_SELECTION = MCI2_4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

2.6.2.2 Initialisation of the Character Set, Related to the Header Control Bits C14-C12 (*refer to file language.sdo*)

LANGUAGE DEFINITION TABLE

External Memory Binary Address	External Memory Chap/Row/Col	Data (hexadecimal) Internal Language Code	Language_# according to the Enhanced Teletext Spec			Language
			C14	C13	C12	
808888	17/02/04	**00**	0	0	0	English
80888A	17/02/05	**20**	0	0	1	Russian
80888C	17/02/06	**0B**	0	1	0	Estonian
80888E	17/02/07	**06**	0	1	1	Czech/Slovak
808890	17/02/08	**01**	1	0	0	German
808892	17/02/09	**08**	1	0	1	Serbian/ Croatian
808894	17/02/10	**0C**	1	1	0	Lithuanian/ Lettish
808896	17/02/11	**21**	1	1	1	Ukranian

2.6.2.3 Initialisation of the Additional Table for the Packet X/26 Character Processing

This table is necessary for the cyrillic and baltic languages. The table is available as an “SDO”-file and can be requested from your Siemens representative (*g2dm_tab.sdo*).

2.6.2.4 Initialisation of the Cyrillic Character Set in the Internal PCS Memory of the Megatext Plus

This character set is available as an “SDO”-file and can be requested from your Siemens representative (*ukr_russ.sdo*).

2.7 Memory Initialisation

The following memory definition must be done in the initialisation sequence of the external controller software:

- Mark the external chapters 0..17 in the M3l registers 91, 92 and 93 as used. These chapters are not available for page acquisition.
- Enable the refresh for the external DRAM (*refer to command interface description [2]*)
- Initialize the CLUT 0 and 1 (*refer to* ***paragraph 1.3*** *of this document*).
- Put the packet buffer into the external DRAM (e.g. **chapter 18**).
(*refer to M3l register 3, 4 and 5 in the document m3l register description [2]*)
- Put the page trace into the external DRAM (e.g. **chapter 19**).
(refer to M3l register 123, 124 and 125 in the document m3l register description [2])
- Define the search groups as described under **paragraph** ***1.2.1*** *ACQ_CONTROL comments* of this document.
- Request the packets 29/0, 1 and 4 for each magazine.
(refer to command interface description [2])
- Initialize the *language definition table (file language.sdo).*
(refer to ***paragraph 1.2.2*** *SERIAL PARALLEL CONVERSION comments of this document)*
- Initialize the additional table for X/26 character processing to NIL:
external memory chap16/row8/col0 = 00_H
- Use the command *INHIBIT_UPDATE* with *ALL* and *UPDATE* set to “1”, to initialize the inhibit table for block2 and block3.
- Use the command *MOVE_MEMORY_SEG* to initialize the external chapters 4-17 to “0”.
- Enable the packets 26, 27 and 28 in the M3l register 106 (*pseudo packet enable*)

2.8 Example for Megatext Plus Initialisation

```
'*************************************************************************************************
Reset Megatext Plus
'*************************************************************************************************
WR_I2C Megatext &d255 0 : delay 100
'*************************************************************************************************
'*************************************************************************************************
Initialisation of the M3L_Register
'*************************************************************************************************
WR_I2C Megatext 108 &h00                   '  acquisition_timing_1
WR_I2C Megatext 109 &h10                   '  acquisition_timing_0
WR_I2C Megatext 115 &h82                   '  black_level_clamp
WR_I2C Megatext 116 &h00                   ' display_timing
WR_I2C Megatext 117 &h00                   '  v_delay_setting
WR_I2C Megatext 081 &h26                   '  set slicer_control
WR_I2C Megatext 082 &h07                   '  output_pin_control
WR_I2C Megatext 083 &hC1                   '  rgb_control
WR_I2C Megatext 085 &h04                   '  display_vco
WR_I2C Megatext 114 &h18                   '  display_pll_control
WR_I2C Megatext 113 &h03                   '  sync_source_selection
WR_I2C Megatext 112 &h00                   '  system_clock_ontrol
WR_I2C Megatext 001 &d00                   '  pb_length
WR_I2C Megatext 002 &d17                   '  pb_length
WR_I2C Megatext 003 &h80                   '  pb_adr_2 Chap_18
WR_I2C Megatext 004 &h90                   '  pb_adr_1
WR_I2C Megatext 088 &hFF                   '  iat_2 Internal Memory allocation
                                           register
WR_I2C Megatext 089 &hFF                   '  iat_1
WR_I2C Megatext 090 &hFF                   '  iat_0
WR_I2C Megatext 091 &h03                   ' xat_2 external Memory allocation
                                           register
WR_I2C Megatext 092 &hFF                   '  xat_1
WR_I2C Megatext 093 &hFF                   '  xat_0
WR_I2C Megatext 096 &d06                   '  dew_start_line
```

```
WR_I$^2$C Megatext 097 &d23              ' dew_end_line
WR_I$^2$C Megatext 098 &d16              ' single_data_line
WR_I$^2$C Megatext 099 &h62              ' ttx_framing_window
WR_I$^2$C Megatext 100 &hB4              ' extra_framing_window
WR_I$^2$C Megatext 105 &hCF              ' reception_threshold
WR_I$^2$C Megatext 106 &h1F              ' pseudo_packet_enable set to all
WR_I$^2$C Megatext 123 &h80              ' pt_adr_2 Page Trace = Chap_19
WR_I$^2$C Megatext 124 &h98              ' pt_adr_1
WR_I$^2$C Megatext 125 &h00              ' pt_adr_0
'*************************************************************************************************
'*************************************************************************************************
```

Init Row attributes from 2/0 ... 2/24 = 47 00 00 00 00 80

```
'*************************************************************************************************
For Co = 0 to 24
  WRRC    48 1  0 2 co &h80
  WRRC    48 2  0 2 co 0
  WRRC    48 4  0 2 co 0
  WRRC    48 8  0 2 co 0
  WRRC    48 16 0 2 co 0
  WRRC    48 32 0 2 co &h47
NEXT
'*************************************************************************************************
'*************************************************************************************************
```

Init display registers

```
'*************************************************************************************************
'Set Sync_Delay_Word
 BRC$="0 03 00" : left=&h000000 : right=&h000400 : GOSUB WRITE_double_word
' Set Display_Position_Word
 BRC$="0 03 01" : left=&h000010 : right=&hEFEA00 : GOSUB WRITE_double_word
' Set Page_Position_Word for single_page_mode"
 BRC$="0 03 02" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Termination_Display_Word"
 BRC$="0 03 05" : left=&h000050 : right=&h000020 : GOSUB WRITE_double_word
' Set Outer_Screen_Mask_Register"
```

```
 BRC$="0 03 06" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Outer_Screen_Display_Word"
 BRC$="0 03 07" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Inner_Screen_Mask_Register_0 ISMR_0"
 BRC$="0 03 10" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Inner_Screen_Display_Word_0  ISDW_0"
 BRC$="0 03 11" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Box_Mask_Register_1 BOXMR_1
 BRC$="0 03 12" : left=&h000000 : right=&h00C000 : GOSUB WRITE_double_word
' Set Box_Display_Word_1 BOXDW_1
 BRC$="0 03 13" : left=&h000000 : right=&h00C000 : GOSUB WRITE_double_word
' Set Inner_Screen_Mask_Register_1 ISMR_1
 BRC$="0 03 14" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
' Set Inner_Screen_Display_Word_1 ISDW_1
 BRC$="0 03 15" : left=&h000000 : right=&h000000 : GOSUB WRITE_double_word
'************************************************************************************************
'************************************************************************************************
```

Define clut 0/1

```
'************************************************************************************************
$include ".\include\clut_ini"
'************************************************************************************************
'************************************************************************************************
```

' switch on refresh of external DRAM

```
'************************************************************************************************
WRRC    48 01 0 6 31 &d64                        ' byte 0 block 0 row 6 column 31
'************************************************************************************************
'************************************************************************************************
```

' set XRAM_SIZE to 512kbyte

```
'************************************************************************************************
WRRC    48 01 0 0 31 &hFF
WRRC    48 02 0 0 31 &h01
'************************************************************************************************
```

```
'************************************************************************************************
CHANNEL_CHANGE
'************************************************************************************************
' ACQ_OFF
'************************************************************************************************
gosub ACQ_OFF
'************************************************************************************************

'************************************************************************************************
' SPC_OFF , set the source_pointer (D1,D0) to a chapter which is used for
' page memory (e.g. chapter 20). Rolling Header handling!
'************************************************************************************************
spc_mode1=0 : spc_mode0 = 0 : D1=&h80 : D0=&hA0 : gosub SPC_OFF
'************************************************************************************************
'************************************************************************************************
' init external chapters 4-17 with 0000 0000 with command Move_Memory_Seg
'************************************************************************************************
for i=4 to 17
chp=i*8
chp_1=chp & &hFF
chp_2=((chp & &hFF00)/256) | &h80
WR_I²C Megatext  mci0_5  chp_2                 '  source_start_2
WR_I²C Megatext  mci0_4  chp_1                 '  source_start_1
WR_I²C Megatext  mci0_3  &h00                  '  source_start_0
WR_I²C Megatext  mci0_2  &h00                  '  source_end_2
WR_I²C Megatext  mci0_1  &h07                  '  source_end_1
WR_I²C Megatext  mci0_0  &hFE                  '  source_end_0
WR_I²C Megatext  mci1_5  chp_2                 '  destination_start_2
WR_I²C Megatext  mci1_4  chp_1                 '  destination_start_1
WR_I²C Megatext  mci1_3  &h00                  '  destination_start_0
WR_I²C Megatext  mci1_1  &h00                  '  substitution_pattern_d
WR_I²C Megatext  mci1_0  &h00                  '  substitution_pattern
WR_I²C Megatext  mci3_3  &h48                  '  find_control
WR_I²C Megatext  mci_com 17                    '  command
Command$ = " MOVE_MEMORY_SEG " : GOSUB watch_command_run
next
'************************************************************************************************
```

```
'************************************************************************************************
' init language_definition_table
'************************************************************************************************
DOWNLOAD ".\sdo\language.sdo" mem=e
'************************************************************************************************
'************************************************************************************************
' Set cursor_control_word (SPC_INTERCHANGE, TRANSFER)
'************************************************************************************************
WRRC    48 32 0 6 20 &h0
'************************************************************************************************
'************************************************************************************************
' init G2/DiacriticalMarks_table (Chap_16/row_8/col_0 = 0)
'************************************************************************************************
WRRC    48 128 16 8 0 &h00
' G2/DiacriticalMarks_table
'DOWNLOAD ".\sdo\g2dm_tab.sdo" mem=e
'************************************************************************************************
'************************************************************************************************
' use the command INHIBIT_UPDATE to initialize the inhibit table
'************************************************************************************************
WR_I²C Megatext  mci0_0  &h03                 '  mode_1 update block2
WR_I²C Megatext  mci_com 10                   '  command
Command$=" INHIBIT_UPDATE " : GOSUB watch_command_run
WR_I²C Megatext  mci0_0  &h07                 '  mode_1 update block3
WR_I²C Megatext  mci_com 10'  command
Command$=" INHIBIT_UPDATE " : GOSUB watch_command_run
'************************************************************************************************
'************************************************************************************************
' init all groups to not_used
'************************************************************************************************
for PRQ_GROUP=&h80 to &h87
WR_I²C Megatext mci0_5  PRQ_GROUP             ' set GR_UNUSE and group no.
WR_I²C Megatext mci3_1  1                     '  set command_run_bit
WR_I²C Megatext mci_com 23                    '  command READ_GROUP
Command$=" READ_GROUP " : GOSUB watch_command_run
next
'************************************************************************************************
```

```
'**********************************************************************************************
' init the activ header ids to NIL
'**********************************************************************************************
BRC$="0 00 22" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 23" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 24" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 25" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 26" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 27" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 28" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
BRC$="0 00 29" : left=&h000000 : right=&h000001 : GOSUB WRITE_double_word
'**********************************************************************************************
'**********************************************************************************************
'  Start of commands
'**********************************************************************************************
' Create free Chap chain
'**********************************************************************************************
WR_I2C Megatext 040 &h80                 '  ffp_chap_2 Chap_20
WR_I2C Megatext 041 &hA0                 '  ffp_chap_1
WR_I2C Megatext 042 &h00                 '  ffp_chap_0
WR_I2C Megatext 044 &h00                 '  nf_chap_1
WR_I2C Megatext 045 &d250                '  nf_chap_0
WR_I2C Megatext mci3_1  1                '  set command_run_bit
WR_I2C Megatext mci_com 7                '  mci command execute
Command$ = " Create free Chap chain " : GOSUB watch_command_run
'**********************************************************************************************
'**********************************************************************************************
' Create free P40 chain
'**********************************************************************************************
WR_I2C Megatext 072 &h89                 '  ffp_p40_2Chap_300
WR_I2C Megatext 073 &h60                 '  ffp_p40_1
WR_I2C Megatext 074 &h00                 '  ffp_p40_0
WR_I2C Megatext 076 &h04                 '  nf_p40_1 1024
WR_I2C Megatext 077 &h00                 '  nf_p40_0
WR_I2C Megatext mci3_1  1                '  set command_run_bit
```

```
WR_I2C Megatext mci_com 8                    '  mci command execute
Command$=" Create free P40 chain " : GOSUB watch_command_run
'*********************************************************************************************
'*********************************************************************************************
' Create free P80 chain
'*********************************************************************************************
WR_I2C Megatext 064 &h8                      '  ffp_p80_2Chap_410
WR_I2C Megatext 065 &hD0                     '  ffp_p80_1
WR_I2C Megatext 066 &h00                     '  ffp_p80_0
WR_I2C Megatext 068 &h00                     '  nf_p80_1 25
WR_I2C Megatext 069 &h25                     '  nf_p80_0
WR_I2C Megatext mci3_1   1                   '  set command_run_bit
WR_I2C Megatext mci_com 9                    '  mci command execute
Command$=" Create free P80 chain " : GOSUB watch_command_run
'*********************************************************************************************
'*********************************************************************************************
' Clear page trace
'*********************************************************************************************
WR_I2C Megatext mci3_1   1                   '  set command_run_bit
WR_I2C Megatext mci_com 6                    '  mci command execute
Command$=" Clear page trace " : GOSUB watch_command_run
'*********************************************************************************************
'*********************************************************************************************
' Init group_0 with type_0"
'*********************************************************************************************
WR_I2C Megatext mci0_2 &h0                   ' type 0
WR_I2C Megatext mci0_1 &h0                   ' group 0
WR_I2C Megatext mci1_5 &h0                   ' do care mask
WR_I2C Megatext mci1_4 &h0                   ' do care mask
WR_I2C Megatext mci1_3 &h0                   ' do care mask
WR_I2C Megatext mci1_2 &h0                   ' do care mask
WR_I2C Megatext mci1_1 &h0                   ' do care mask
WR_I2C Megatext mci3_1 1                     ' set command_run_bit
WR_I2C Megatext mci_com 50                   ' command WRITE_GROUP
Command$=" WRITE_GROUP " : GOSUB watch_command_run
'*********************************************************************************************
```

```
'*******************************************************************************************
' Init group_3 with type_3"
'*******************************************************************************************
WR_I2C Megatext mci0_2 &h03                    ' type 3
WR_I2C Megatext mci0_1 &h03                    ' group 3
WR_I2C Megatext mci1_5 &hFF                    ' do care mask
WR_I2C Megatext mci1_4 &h0F                    ' do care mask
WR_I2C Megatext mci1_3 &h00                    ' do care mask
WR_I2C Megatext mci1_2 &h00                    ' do care mask
WR_I2C Megatext mci1_1 &h00                    ' do care mask
WR_I2C Megatext mci3_1 1                       ' set command_run_bit
WR_I2C Megatext mci_com 50                     ' command WRITE_GROUP
Command$= " WRITE_GROUP " : GOSUB watch_command_run
'*******************************************************************************************
'*******************************************************************************************
' Init group_4 with type_2 for POP_ and DRCS_Pages"
'*******************************************************************************************
WR_I2C Megatext mci0_2 &h02                    ' type 2
WR_I2C Megatext mci0_1 &h04                    ' group 4
WR_I2C Megatext mci1_5 &h07                    ' do care mask
WR_I2C Megatext mci1_4 &hFF                    ' do care mask
WR_I2C Megatext mci1_3 &h0F                    ' do care mask
WR_I2C Megatext mci1_2 &h00                    ' do care mask
WR_I2C Megatext mci1_1 &h00                    ' do care mask
WR_I2C Megatext mci3_1 1                       ' set command_run_bit
WR_I2C Megatext mci_com 50                     ' command WRITE_GROUP
Command$=" WRITE_GROUP " : GOSUB watch_command_run
'*******************************************************************************************
'*******************************************************************************************
' Init group_5 with type_2 for AIT's, MpT's, MpET's"
'*******************************************************************************************
WR_I2C Megatext mci0_2 &h02                    ' type 2
WR_I2C Megatext mci0_1 &h05                    ' group 5
WR_I2C Megatext mci1_5 &h07                    ' do care mask
```

```
WR_I2C Megatext mci1_4 &hFF                 ' do care mask
WR_I2C Megatext mci1_3 &h7F                 ' do care mask
WR_I2C Megatext mci1_2 &h3F                 ' do care mask
WR_I2C Megatext mci1_1 &h00                 ' do care mask
WR_I2C Megatext mci3_1 1                    ' set command_run_bit
WR_I2C Megatext mci_com 50                  ' command WRITE_GROUP
Command$=" WRITE_GROUP " : GOSUB watch_command_run
'*******************************************************************************************
'*******************************************************************************************
' Init group_6 with type_2 for BTT"
'*******************************************************************************************
WR_I2C Megatext mci0_2 &h02                 ' type 2
WR_I2C Megatext mci0_1 &h06                 ' group 6
WR_I2C Megatext mci1_5 &h07                 ' do care mask
WR_I2C Megatext mci1_4 &hFF                 ' do care mask
WR_I2C Megatext mci1_3 &h00                 ' do care mask
WR_I2C Megatext mci1_2 &h3F                 ' do care mask
WR_I2C Megatext mci1_1 &h00                 ' do care mask
WR_I2C Megatext mci3_1 1                    ' set command_run_bit
WR_I2C Megatext mci_com 50                  ' command WRITE_GROUP
Command$= " WRITE_GROUP " : GOSUB watch_command_run
'*******************************************************************************************
'*******************************************************************************************
' Init group_7 with type_2 for MOT's"
'*******************************************************************************************
WR_I2C Megatext mci0_2 &h02                 ' type 2
WR_I2C Megatext mci0_1 &h07                 ' group 7
WR_I2C Megatext mci1_5 &h07                 ' do care mask
WR_I2C Megatext mci1_4 &hFF                 ' do care mask
WR_I2C Megatext mci1_3 &h0F                 ' do care mask
WR_I2C Megatext mci1_2 &h00                 ' do care mask
WR_I2C Megatext mci1_1 &h0                  ' do care mask
WR_I2C Megatext mci3_1 1                    ' set command_run_bit
WR_I2C Megatext mci_com &d50                ' command WRITE_GROUP
Command$= " WRITE_GROUP " : GOSUB watch_command_run
'*******************************************************************************************
```

```
'*********************************************************************************************
' add pseudopackets x29/0,1,4 for all magazines
'*********************************************************************************************
for mag_nr =232 to 239                      ' E8 to EF
WR_I2C Megatext mci0_5 &h04                 '  packet_control pckblf = 1
WR_I2C Megatext mci0_1 &h03                 '  prq_group = 3
WR_I2C Megatext mci1_5 mag_nr               '  row + magazine
WR_I2C Megatext mci1_4 &h00                 '  designationcode
WR_I2C Megatext mci3_1 1                    '  set command_run_bit
WR_I2C Megatext mci_com &d02                '  command ADD_PACKET_29_30
Command$= " ADD_PACKET_29_30 " : GOSUB watch_command_run
WR_I2C Megatext mci0_5 &h04                 '  packet_control pckblf = 1
WR_I2C Megatext mci0_1 &h03                 '  prq_group = 3
WR_I2C Megatext mci1_5 mag_nr               '  row + magazine
WR_I2C Megatext mci1_4 &h01                 '  designationcode
WR_I2C Megatext mci3_1 1                    '  set command_run_bit
WR_I2C Megatext mci_com &d02                '  command ADD_PACKET_29_30
Command$= " ADD_PACKET_29_30 " : GOSUB watch_command_run
WR_I2C Megatext mci0_5 &h04                 '  packet_control pckblf = 1
WR_I2C Megatext mci0_1 &h03                 '  prq_group = 3
WR_I2C Megatext mci1_5 mag_nr               '  row + magazine
WR_I2C Megatext mci1_4 &h04                 '  designationcode
WR_I2C Megatext mci3_1 1                    '  set command_run_bit
WR_I2C Megatext mci_com &d02                '  command ADD_PACKET_29_30
Command$= " ADD_PACKET_29_30 " : GOSUB watch_command_run
next
'*********************************************************************************************
'*********************************************************************************************
' add all pages with search type0
'*********************************************************************************************
request_mode=&h51 : check=0 : page=&h100 : GOSUB ADD_ALL_PAGES
'*********************************************************************************************
```

```
'***********************************************************************************************
' Add Basic_TOP_Table 1F0_3Fxx
'***********************************************************************************************
acq_control_1=&h14 : group=6 : check=2 : page=&h1F0
sub_minutes=&h00 : sub_hours=&h3F : GOSUB ADD_PAGE
'***********************************************************************************************
'***********************************************************************************************
' ACQ_ON
'***********************************************************************************************
gosub ACQ_ON
'***********************************************************************************************
'***********************************************************************************************
' SPC_ON, search dislay page, write UDC, call spc
'***********************************************************************************************
page=&h100 : Sub_Code=0 : GOSUB Search_page
block=2 : GOSUB WRITE_page_number
E1=D1 : E0=D0 : GOSUB SPC
'***********************************************************************************************
'***********************************************************************************************
Key_loop:
kb=inkey()
IF kb=Esc                     GOTO  EXIT
IF kb=CursorUp_key            GOSUB NEXT_PAGE
IF kb=CursorDown_key          GOSUB PREVIOUS_PAGE
IF kb=CursorLeft_key          GOSUB MOVE_SCREEN_LEFT
IF kb=CursorRight_key         GOSUB MOVE_SCREEN_RIGHT
IF kb=F1_key                  GOSUB MIX_TV_MODE
IF kb=F2_key                  GOSUB TOGGLE_CONCEAL
IF kb=F3_key                  GOSUB TOGGLE_ACQ
IF kb=F4_key                  GOSUB TOGGLE_DUAL_SINGLE
IF kb=F5_key                  GOSUB WINDOW
IF kb=F7_key                  GOSUB LEVEL
IF kb=F8_key                  GOSUB AUTO_NEXT
IF kb=F9_key                  GOSUB WRITE_TOP_TITLE
IF kb=F10_key                     GOSUB SET_TO_PG_TR_MODE
IF kb=shift_F10_key               GOTO  CHANNEL_CHANGE
IF kb = CR_key                    GOTO  START
IF kb>47                          GOSUB FIX_PAGE
goto Key_LOOP
'************************************************************************************************************
```

2.9 Time/Rolling Header Processing

In Megatext Plus SDA 5275 the processing of time and rolling header has been changed compared to the SDA 5273. The time (Bytes 32-39) and the rolling header (Bytes 8-31) are now written directly into the current display chapter. This chapter is defined through the *Address Pointer to Basic Display Page* in the command *SERIAL_PARALLEL_ CONVERSION*. Therefore it is necessary to send the command *SERIAL_ PARALLEL_CONVERSION* once with a valid input parameter *Address Pointer to Basic Display Page*. The rest of the input parameters are not relevant in this case. This should be done before giving the *TIME_DISPLAY* command or before using the time and rolling header data for other purpose.

2.10 Display Position Word

The column value is always modified relatively to the initialisation value by the firmware. Depending on the position (left, right or split), the display is automatically shifted.

Recommended Startvalue:

DPW Block0/Row3/Col1

Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
00_H	00_H	10_H	EF_H	EA_H	00_H

3 Abbreviations

TOP	Table Of Pages
BTT	Basic Top Table
AI	Additional Information Table
MP	Multi Page table
MPEX	Multi Page Extention table
FLOF	Full Level One Feature
OSD	On Screen Display
IRT	Institut für Rundfunk Technik (in Munich,Germany)
WST	World Standard Teletext specification
EBU	European Broadcasting Union
ACQ	Acquisition firmware routine
S/P-C	Serial Parallel Conversion
MOT	Magazine Organisation Table
POP	Public Object Page
GPOP	Global Public Object Page
DRCS	Dynamic Redefinable Character Set
GDRCS	Global Dynamic Redefinable Character Set
PCS	Programmable Character Set
MCI	Megatext Command Interface
WSS	Wide Screen Signaling
VPS	Video Programm System
CDW	Character Display Word
UDC	User Definable Characters
VBI	Vertical Blanking Interval

4 References

[1] IRT-Institut für Rundfunk Technik “TOP System for Teletext”

[2] Megatext Documentation Volumn 1

[3] Enhanced Teletext Specification, European Telecommunications Standards Institute ETSI

[4] Television Systems; 625-Line Television Wide Screen Signaling, European Telecommunications Standards Institute ETSI

[5] IRT-Institut für Rundfunk Technik “Video-Programm-System”, Germany.