



**WESTERN AVIONICS**

**STANAG 3910 PC/AT  
LabWindows/CVI Drivers**

**P/N 1L01716H01 Rev 2.0**

**User Manual  
UM 01716 Rev B**

**© Western Avionics Ltd.  
13/14 Shannon Free Zone  
Co. Clare  
Ireland**

21 October 1998

# Western Avionics 3910/3838 PC/AT

## Introduction:

The Western Avionics Model 4510 PC-card is a standard plug-in PC/AT compatible card designed to meet the requirements of STANAG 3910. The card provides a powerful and intelligent interface between the PC host equipment and the STANAG 3910 data bus.

Bus Controller, Multiple-Remote Terminal with Monitoring can operate both independently and simultaneously. Chronological Bus Monitor mode is also provided, as an independent function.

The card provides comprehensive test and simulation functions for both STANAG 3910 and STANAG 3838 (MIL-STD-1553) systems.

## WA4510 Driver.

This instrument driver provides programming support for the Western Avionics 3910/3838 PC/AT.

It contains functions for opening, configuring, taking measurements from, and closing the instrument.

These drivers utilise modules provided by the standard "C" driver library, Part Number 1L01616H01, and the User Manual for these drivers should be consulted for detailed information on the operation of the various modules concerned.

## How To Use This Document:

This document is intended to be used as a programming reference manual.

It describes each function in the Western Avionics 3910/3838 PC/AT instrument.

The functions are listed in alphabetical order with a description of the function, C syntax of the function, and description of each parameter. Error codes for each function are common, and detailed in appendix A (see pages 53 to 54).

## Function Tree Layout:

Western Avionics 3910/3838 PC/AT	Function Name:
Initialise	WA4510_init
<i>Configuration Functions</i>	
<i>BC Functions</i>	
Create BC message	WA4510_crMsg
Create BC cycle	WA4510_crCycle
Create BC frame	WA4510_crFrame
<i>BC Modifiers</i>	
Modify BC message	WA4510_modMsg
Modify BC cycle	WA4510_modCycle
Modify BC frame	WA4510_modFrame
Modify BC RT table	WA4510_modBcMrt
<i>MRT Functions</i>	
Create MRT sub-address	WA4510_crMrtRtSa
<i>MRT Modifiers</i>	
Modify MRT table	WA4510_modMrtRt
Modify MRT sub-address	WA4510_modMrtRtSa
Modify MRT Mode Code	WA4510_modMrtRtMd
<i>MON Functions</i>	
Define MON triggers	WA4510_defTrigs
Define MON sequence	WA4510_defSeq
<i>Action / Status Functions</i>	
Run BC frame	WA4510_runFrame
Halt BC	WA4510_haltBC
Run MRT	WA4510_runMRT
Halt MRT	WA4510_haltMRT
Run MON	WA4510_runMON
Get MON range	WA4510_getRange
Halt MON	WA4510_haltMON
<i>Data Functions</i>	
Read BC message	WA4510_rdMsg
Read BC cycle	WA4510_rdCycle
Read BC frame	WA4510_rdFrame
Read BC RT table	WA4510_rdBcMrt
Read MRT table	WA4510_rdMrtRt
Read MRT sub-address	WA4510_rdMrtRtSa
Read MRT Mode Code	WA4510_rdMrtRtMd
Find MON messages	WA4510_findMsgs
Get MON message	WA4510_getMsg
<i>Utility Functions</i>	
Read word from card	WA4510_rWord
Write word to card	WA4510_wWord
Execute command	WA4510_exeCommand
Execute self-test	WA4510_selfTest
Get revision info	WA4510_revision
Read clock	WA4510_readCLK
Close	WA4510_close

---

The following functions are in alphabetical order.

---

## WA4510\_close

### Function

ViUInt16 WA4510\_close (void);

### Purpose

Function Name: WA4510\_close

Description: Closes down the driver operation and releases any memory resources opened by the drivers.

### Return Value

Control Name error

Description Returned error codes.

Return Type ViUInt16

Valid Range 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_crCycle

ViUInt16 WA4510\_crCycle (ViUInt32 \*cycleID, ViUInt16 msgCount, ViUInt32 msgList[ ]);

### Purpose

Function Name: WA4510\_crCycle

Description: Create a minor BC cycle.

### Parameter List

#### cycleID

Variable Type ViUInt32 (passed by reference)

Control Name : cycleID

Description : Cycle handle for created cycle. This is a single 32 bit output that returns the handle for the created cycle.

Return Type : ViUInt32

#### msgCount

Variable Type ViUInt16

Control Name : msgCount

Description : Number of messages in cycle. This should be the number of messages in the msgList[ ].

Variable Type : ViUInt16

#### msgList

Variable Type ViUInt32[ ]

Control Name : msgList

Description : This is an array of message IDs defining the order of the cycle to be created. The number of messages in the list should equal the value of msgCount.

Variable Type : ViUInt32

### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_crFrame

### Function

ViUInt16 WA4510\_crFrame (ViUInt32 \*frameID, ViUInt16 cycCount, ViUInt32 cycList[ ]);

### Purpose

Function Name: WA4510\_crFrame  
Description: Create a BC major frame.

### Parameter List

#### frameID

Variable Type ViUInt32 (passed by reference)  
Control Name : frameID  
Description : Frame handle for created frame. This is a single 32 bit output that returns the handle for the created frame.  
Return Type : ViUInt32

#### cycCount

Variable Type ViUInt16  
Control Name : cycCount  
Description : Number of cycles in frame. This should be the number of cycles in the cycList[ ].  
Variable Type: ViUInt16

#### cycList

Variable Type ViUInt32[ ]  
Control Name : cycList  
Description : This is an array of cycle IDs defining the order of the frame to be created. The number of cycles in the list should equal the value of cycCount.  
Variable Type: ViUInt32

### Return Value

Control Name: error  
Description: Returned error codes.  
Return Type: ViUInt16  
Valid Range: 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_crMrtRtSa

### Function

ViUInt16 WA4510\_crMrtRtSa (ViUInt16 rtAddr, ViUInt16 subAdd, ViUInt16 saType, ViUInt16 wrap,  
ViUInt16 rxWCnt, ViUInt16 rxErrs[ ], ViUInt16 txWCnt, ViUInt16 txData[ ],  
ViUInt16 txErrs[ ], ViUInt16 hsRiTm, ViUInt16 hsTiTm);

### Purpose

Function Name: WA4510\_crMrtRtSa  
Description: Defines and creates a RT sub-address in MRT mode.

### Parameter List

#### rtAddr

Variable Type : ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### subAdd

Variable Type : ViUInt16  
Control Name : subAdd  
Description : RT sub-address.  
Variable Type : ViUInt16  
Valid values : 0 to 30 for Low speed  
1 to 127 for High speed  
Default : 1

#### saType

Variable Type : ViUInt16  
Control Name : saType  
Description : Type of sub-address (LS or HS).  
Variable Type : ViUInt16  
Values:

Driver constant	Value	Description
LS_SA	1	LS sub-address
HS_SA	2	HS sub-address

Default: LS\_SA 1 LS sub-address

#### wrap

Variable Type : ViUInt16  
Control Name : wrap  
Description: Mode for data buffer. If wrap enabled then the TX and RX buffers will be the same. If not enabled then separate buffers will be created.  
Variable Type : ViUInt16  
Values:

Driver constant	Value	Description
NO_WRAP	1	Wrap disabled
WRAP	2	Wrap enabled

Default: NO\_WRAP 1 Wrap disabled

#### rxWCnt

Variable Type : ViUInt16  
Control Name : rxWCnt  
Description : Number of RX data words expected for this RT sub-address  
Variable Type : ViUInt16  
Default : 0

rxErrs

Variable Type ViUInt16[ ]  
 Control Name : rxErrs  
 Description : LS and HS error injection definition for RX messages. This is an array of 7 x 16 bit input values defining the LS and HS RX errors.  
 Variable Type: ViUInt16

rxErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

rxErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection. For further information see software driver library manual.

rxErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection. For further information see software driver library manual.

rxErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

rxErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection. For further information see software driver library manual.

rxErrs[5]: HS gate error position.

rxErrs[6]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.

Default : 0

txWCnt

Variable Type ViUInt16  
 Control Name : txWCnt  
 Description : Number of TX data words expected for this RT sub-address  
 Variable Type : ViUInt16  
 Default : 0

txData

Variable Type ViUInt16[ ]  
 Control Name : txData  
 Description : This is an input array of txWCnt 16 bit words for initialising the TX data buffer.  
 Variable Type : ViUInt16  
 Default : 0

txErrs

Variable Type ViUInt16[ ]  
 Control Name : txErrs  
 Description : LS and HS error injection definition for TX messages. This is an array of 7 x 16 bit input values defining the LS and HS TX errors.  
 Variable Type: ViUInt16

txErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

txErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
 For further information see software driver library manual.

txErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
 For further information see software driver library manual.

txErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

txErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection.  
 For further information see software driver library manual.

txErrs[5]: HS gate error position.

txErrs[6]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.  
 Default : 0



#### hsRiTm

Variable Type : ViUInt16  
Control Name : hsRiTm  
Description : This is the HS RI timeout value in uS.  
Variable Type : ViUInt16  
Valid Values : 30 to 255 (uS)  
Default : 0

#### hsTiTm

Variable Type : ViUInt16  
Control Name : hsTiTm  
Description : This is the HS TI time value in uS.  
Variable Type : ViUInt16  
Valid Values : 10 to 255 (uS)  
Default : 0

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_crMsg

### Function

ViUInt16 WA4510\_crMsg (ViUInt32 \*msgID, ViUInt16 msgInfo[ ], ViUInt16 dataCount, ViUInt16 dataWords[ ]);

### Purpose

Function Name: WA4510\_crMsg  
Description: Creates a BC message.

### Parameter List

#### msgID

Variable Type ViUInt32 (passed by reference)  
Control Name : msgID  
Description : Message handle for created message.  
This is a single 32 bit output that returns the handle for the created message.  
Return Type : ViUInt32

#### msgInfo

Variable Type ViUInt16[ ]  
Control Name : msgInfo  
Description : Information describing the message.  
This is an array of 20 x 16 bit input values.  
Variable Type : ViUInt16

#### msgInfo[0]: Message type.

Values:	Driver constant	Value	Description
	TXMD_WODA_3838	0x8000	TX mode no data
	RXMD_WIDA_3838	0x0001	RX mode + data
	TXMD_WIDA_3838	0x8001	TX mode + data
	RT_RT_3838	0x0002	3838 RT to RT
	BC_RT_3838	0x0003	3838 BC to RT
	RT_BC_3838	0x8003	3838 RT to BC
	BC_RT_3910	0x0004	3910 BC to RT
	RT_BC_3910	0x8005	3910 RT to BC
	RT_RT_3910	0x0006	3910 RT to RT
	MODE_3910	0x0007	3910 Mode code
	TX_MSG_3910	0x8007	3910 TX message
	B_MODE_WODA_3838	0x8008	Bcast mode no data
	B_RXMD_WIDA_3838	0x0009	Bcast mode + data
	B_RT_RT_3838	0x000A	Bcast 3838 RT to RT
	B_BC_RT_3838	0x000B	Bcast 3838 BC to RT
	B_BC_RT_3910	0x000C	Bcast 3910 BC to RT
	B_RT_RT_3910	0x000E	Bcast 3910 RT to RT
	B_MODE_3910	0x000F	Bcast 3910 mode code

#### msgInfo[1]: LsMsgGap - LS inter-message gap-time

Values: 4 to 65535 uS.

#### msgInfo[2]: LsBus - LS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 1553 bus
	SECONDARY_BUS	1	Secondary 1553 bus

msgInfo[3]: RT1 - 1st RT number.

Values: 0 to 31.

msgInfo[4]: Sub-address for RT1

Values: 1 to 30 for 3838.  
1 to 127 for 3910.

msgInfo[5]: RT2 - 2nd RT number.

Values: 0 to 31.

msgInfo[6]: Sub-address for RT2

Values: 1 to 30 for 3838  
1 to 127 for 3910.

msgInfo[7]: Message word-count.

msgInfo[8]: 3838 mode code number (if message type is LS mode code).

msgInfo[9]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

msgInfo[10]: LS error injection info.

This contains extra information to describe the 3838 error injection.  
For further information see software driver library manual.

msgInfo[11]: LS error position info.

This contains extra information defining the word number in the phase of the message for the error injection. For further information see software driver library manual.

msgInfo[12]: LS error phase info.

This defines the phase of the message for the error.

Values:	Driver constant	Value	Description
	ERR_DISABLED	0	No error injection
	ERR_FIR_BC_TX	1	Inject in 1st BC TX.
	ERR_SEC_BC_TX	2	Inject in 2nd BC TX.
	ERR_FIR_RT_TX	3	Inject in 1st RT TX.
	ERR_SEC_RT_TX	4	Inject in 2nd RT TX.

For further information see software driver library manual.

msgInfo[13] HsRtRtIMsgGap - 3910 RT to RT inter-message gap-time.

Values: 10 to 65535 uS.

msgInfo[14]: HsBus - HS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 3910 bus
	SECONDARY_BUS	1	Secondary 3910 bus

msgInfo[15]: 3910 error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

msgInfo[16]: HS error injection info.

This contains extra information to describe the 3910 error injection.

For further information see software driver library manual.

msgInfo[17]: HS gate error position.

msgInfo[18]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.

msgInfo[19]: HsRiTmOut - HS RT/TI time in uS.

dataCount

Variable Type : ViUInt16  
Control Name : dataCount  
Description : Size of data buffer required for message.  
Variable Type : ViUInt16

dataWords

Variable Type : ViUInt16[ ]  
Control Name : dataWords  
Description : Array of 16 bit values of size 'dataCount' to be copied to the data buffer.  
Variable Type : ViUInt16

Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_defSeq

### Function

ViUInt16 WA4510\_defSeq (ViChar sequence[ ], ViUInt16 \*errInfo);

### Purpose

Function Name: WA4510\_defSeq

Description: This function defines the Bus Monitor trigger sequence.

### Parameter List

sequence

Variable Type ViChar[ ]

Control Name : sequence

Description : This text input string defines the trigger sequence for bus monitor capture.  
The format of this text is described in the drivers manual.

Variable Type : ViChar

errInfo

Variable Type ViUInt16 (passed by reference)

Control Name : errInfo

Description : In the event of a syntax error in the sequence[ ] string, this output value will define the position  
in sequence[ ] where the syntax error occurred.

Return Type : ViUInt16

### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_defTrigs

### Function

ViUInt16 WA4510\_defTrigs (ViChar trigger[ ], ViUInt16 \*errInfo);

### Purpose

Function Name: WA4510\_defTrigs

Description: This function defines the Bus Monitor trigger words.

### Parameter List

#### trigger

Variable Type ViChar[ ]

Control Name : trigger

Description : This text input string defines up to 4 triggers.  
The format of this text is described in the drivers manual.

Variable Type : ViChar

#### errInfo

Variable Type ViUInt16 (passed by reference)

Control Name : errInfo

Description : In the event of a syntax error in the trigger[ ] string, this output value will define the position in trigger[ ] where the syntax error occurred.

Return Type : ViUInt16

### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_exeCommand

### Function

ViUInt16 WA4510\_exeCommand (ViUInt16 command);

### Purpose

Function Name: WA4510\_exeCommand  
Description: Execute a card command.

### Parameter List

command

Variable Type : ViUInt16  
Control Name : command  
Description : Execute defined command.  
Variable Type : ViUInt16  
Values :  
0x001 Go to BC mode  
0x002 Go to MRT mode  
0x003 Go to MON mode  
0x004 BC cold start  
0x005 BC warm start  
0x006 BC stop  
0x007 MRT cold start  
0x008 MRT warm start  
0x009 MRT stop  
0x00A Pause  
0x00B Unpause  
0x00C Load clock  
0x00D Self-test  
0x00E Run Monitor  
0x00F Stop Monitor  
0x010 Synchronize Clock

Note: 0x00A (Pause) and 0x00B (Unpause) are used to STOP and RESTART the Local Clock

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_findMsgs

### Function

ViUInt16 WA4510\_findMsgs (ViChar pattern[ ], ViUInt32 maxList, ViInt32 msgNums[ ], ViUInt16 \*errInfo);

### Purpose

Function Name: WA4510\_findMsgs

Description: This function searches the Bus Monitor stack for the occurrences of a defined message.

### Parameter List

#### pattern

Variable Type: ViChar[ ]

Control Name: pattern

Description: This text input string defines word to search for on the Bus Monitor stack. The format of this texts described in the drivers manual.

Variable Type: ViChar

#### maxList

Variable Type: ViUInt32

Control Name: maxList

Description: This input shall define the maximum allowable message numbers to be stored in msgNums[ ].

Variable Type: ViUInt32

#### msgNums

Variable Type: ViInt32[ ]

Control Name: msgNums

Description: This output array shall return with the message numbers on the stack that contained the search pattern defined by pattern[ ].

Return Type: ViInt32

#### errInfo

Variable Type: ViUInt16 (passed by reference)

Control Name: errInfo

Description: In the event of a syntax error in the pattern[ ] string, this output value will define the position in pattern[ ] where the syntax error occurred.

Return Type: ViUInt16

#### Return Value

Control Name: error

Description: Returned error codes.

Return Type: ViUInt16

Valid Range: 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).



## WA4510\_getMsg

### Function

ViUInt16 WA4510\_getMsg (ViUInt32 msgNum, ViUInt16 maxData, ViUInt16 msgInfo[ ]);

### Purpose

Function Name: WA4510\_getMsg

Description: This function gets the defined message data from the Bus Monitor stack.

### Parameter List

#### msgNum

Variable Type ViUInt32

Control Name: msgNum

Description : This is the required message number from the stack

Variable Type : ViUInt32

#### maxData

Variable Type ViUInt16

Control Name: maxData

Description : This input shall be set to the maximum allowable words to be read into msgInfo[ ].

Variable Type : ViUInt16

#### msgInfo

Variable Type ViUInt16[ ]

Control Name: msgInfo

Description : This output is an array of 16 bit words for returning the information and data related to the message. The format of this data is described in the drivers manual.

Return Type : ViUInt16

#### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_getRange

### Function

ViUInt16 WA4510\_getRange (ViInt32 \*total, ViInt32 \*negative, ViInt32 \*positive);

### Purpose

Function Name: WA4510\_getRange

Description: This function gets the message number range on the Bus Monitor stack.

### Parameter List

total

Variable Type ViInt32 (passed by reference)

Control Name : total

Description : This output value shall return the total number of messages on the stack.

Return Type : ViUInt32

negative

Variable Type ViInt32 (passed by reference)

Control Name : negative

Description : This output value shall return the most negative message number on the stack.

Return Type : ViUInt32

positive

Variable Type ViInt32 (passed by reference)

Control Name : positive

Description : This output value shall return the most positive message number on the stack.

Return Type : ViUInt32

Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_haltBC

### Function

ViUInt16 WA4510\_haltBC (void);

### Purpose

Function Name: WA4510\_haltBC  
Description: Halts the BC transmission.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_haltMON

ViUInt16 WA4510\_haltMON (void);

### Purpose

Function Name: WA4510\_haltMON  
Description: This function halts the card in MON mode.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_haltMRT

ViUInt16 WA4510\_haltMRT (void);

### Purpose

Function Name: WA4510\_haltMRT  
Description: This function halts the card in MRT mode.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_init

### Function

ViUInt16 WA4510\_init (ViUInt32 memAddress, ViUInt16 iopAddress, ViUInt16 coupling1553, ViUInt16 cardMode, ViUInt16 hsSubaddress, ViUInt16 rtRespTm, ViUInt16 rtRespTout, ViUInt16 bcCycleTm);

### Purpose

Function Name: WA4510\_init  
Description: Initialises the card with default data and puts the card into the desired mode of operation.

### Parameter List

#### memAddress

Variable Type ViUInt32  
Control Name : memAddress  
Description : Base address in PC for 8KByte PCAT card interface buffer.  
Variable Type : ViUInt32  
Valid Range : 0xA0000, 0xB0000, 0xC0000, 0xD0000, 0xE0000, 0xF0000  
Default Value : 0xD0000

#### iopAddress

Variable Type ViUInt16  
Control Name : iopAddress  
Description : Base address of PCAT I/O ports.  
Variable Type : ViUInt16  
Valid Range : 0x0100 to 0xFFE0 in 0x10 steps  
Default Value : 0x0500

#### coupling1553

Variable Type ViUInt16  
Control Name : coupling1553  
Description : Coupling to 1553 bus.  
Variable Type : ViUInt16  
Valid Range :

Driver constant	Value	Description
DIRECT_COUPLING	0	1553 direct coupling
STUB_COUPLING	1	1553 stub coupling

  
Default Value : DIRECT\_COUPLING

#### cardMode

Variable Type ViUInt16  
Control Name : cardMode  
Description : Sets card mode of operation.  
Variable Type : ViUInt16  
Valid Range : BCMRT\_MODE (for BcMrt mode)  
MRT\_MODE (for Mrt mode)  
CM\_MODE (for Chron. Mon. mode)  
Default Value : BCMRT\_MODE (bus controller)

#### hsSubaddress

Variable Type ViUInt16  
Control Name : hsSubaddress  
Description : Sets high-speed sub-address  
Variable Type : ViUInt16  
Valid Range : 1 (Rafale)  
26 (EFA)  
0xFFFF (1553 only)  
Default Value : 0xFFFF (1553)

#### rtRespTm

Variable Type : ViUInt16  
Control Name : rtRespTm  
Description : Sets RT response time.  
Variable Type : ViUInt16  
Valid Range : 4 to 30 (in microseconds)  
Default Value : 6

#### rtRespTout

Variable Type : ViUInt16  
Control Name : rtRespTout  
Description : Sets RT response timeout value.  
Variable Type : ViUInt16  
Valid Range : 10 to 255 (in microseconds)  
Default Value : 14

#### bcCycleTm

Variable Type : ViUInt16  
Control Name : bcCycleTm  
Description : Sets BC cycle time value.  
Variable Type : ViUInt16  
Valid Range : 100 to 65535 (in microseconds)  
Default Value : 20000

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modBcMrt

### Function

ViUInt16 WA4510\_modBcMrt (ViUInt16 rtAddr, ViUInt16 info[ ]);

### Purpose

Function Name: WA4510\_modBcMrt  
Description: Modifies the selected MRT table when the card is in BC mode.

### Parameter List

#### rtAddr

Variable Type : ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid Range : 0 to 31  
Default Value : 0

#### info

Variable Type : ViUInt16[ ]  
Control Name : info  
Description : Values for modification. This is an array of 5 x 16-bit input values  
Variable Type : ViUInt16  
Valid Range :

info[0]: 0 = RT monitored, 1 = RT simulated.  
info[1]: RT status word to be transmitted.  
info[2]: RT vector word to be transmitted for TX VECTOR mode code.  
info[3]: RT BIT word to be transmitted for 1553 TX BIT WORD mode.  
info[4]: RT BIT word to be transmitted for 3910 TX BIT WORD mode.

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modCycle

### Function

ViUInt16 WA4510\_modCycle (ViUInt32 cycleID, ViUInt16 msgCount, ViUInt32 msgList[ ]);

### Purpose

Function Name: WA4510\_modCycle  
Description: Modify a previously created BC cycle.

### Parameter List

#### cycleID

Variable Type : ViUInt32  
Control Name : cycleID  
Description : Cycle handle for previously created cycle.  
Variable Type : ViUInt32

#### msgCount

Variable Type : ViUInt16  
Control Name : msgCount  
Description : Number of messages in cycle. This should be the number of messages in the msgList[ ].  
Variable Type : ViUInt16

#### msgList

Variable Type : ViUInt32[ ]  
Control Name : msgList  
Description : This is an array of message IDs defining the order of the cycle to be created. The number of messages in the list should equal the value of msgCount.  
Return Type : ViUInt32

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modFrame

### Function

ViUInt16 WA4510\_modFrame (ViUInt32 frameID, ViUInt16 cycCount, ViUInt32 cycList[ ]);

### Purpose

Function Name: WA4510\_modFrame  
Description: Modify a BC major frame.

### Parameter List

#### frameID

Variable Type ViUInt32  
Control Name : frameID  
Description : Frame handle for previously created frame.  
Variable Type : ViUInt32

#### cycCount

Variable Type ViUInt16  
Control Name : cycCount  
Description : Number of cycles in frame. This should be the number of cycles in the cycList[ ].  
Variable Type: ViUInt16

#### cycList

Variable Type ViUInt32[ ]  
Control Name : cycList  
Description : This is an array of cycle IDs defining the order of the frame to be modified. The number of cycles in the list should equal the value of cycCount.  
Return Type : ViUInt32

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003  
Possible error codes and descriptions are as in appendix A (see pages 53 to 54).



## WA4510\_modMrtRt

### Function

ViUInt16 WA4510\_modMrtRt (ViUInt16 rtAddr, ViUInt16 info[ ]);

### Purpose

Function Name: WA4510\_modMrtRt

Description: Modifies the selected MRT table when the card is in MRT mode.

### Parameter List

rtAddr

Variable Type ViUInt16

Control Name : rtAddr

Description : RT address.

Variable Type : ViUInt16

Valid values : 0 to 31

Default : 0

info

Variable Type ViUInt16[ ]

Control Name : info

Description : Values for modification. This is an array of 10 x 16 bit input values.

Variable Type : ViUInt16

info[0]: 0 = RT monitored, 1 = RT simulated.

info[1]: RT status word to be transmitted.

info[2]: RT vector word to be transmitted for TX VECTOR mode code.

info[3]: RT BIT word to be transmitted for 1553 TX BIT WORD mode.

info[4]: RT BIT word to be transmitted for 3910 TX BIT WORD mode.

info[5]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error
	NO_3838_RESP_ERR	0x85	No response error

info[6]: LS error injection info.

This contains extra information to describe the 3838 error injection.

For further information see software driver library manual.

info[7]: LS error position info.

This contains extra information defining the word number of the message for the error injection.

For further information see software driver library manual.

info[8]: Disable Error Bus definition.

This defines which 3838 bus the errors are allowed to be injected on.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	No errors on PRIMARY
	SECONDARY_BUS	1	No errors on SECONDARY
	BOTH_BUSES	2	No errors on any BUS
	NO_BUSES	3	Enable on any BUS

info[9]: Disable HS Mode Codes.

This defines which HS mode codes are disabled for the RT.

Values:	Driver constant	Value	Description
	HS_MODES_OFF	1	Disable all mode codes
	TXTYPE	2	Disable TX mode codes
	RXTYPE	3	Disable RX mode codes
	HS_MODES_ON	4	All mode codes enabled

#### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modMrtRtMd

### Function

ViUInt16 WA4510\_modMrtRtMd (ViUInt16 rtAddr, ViUInt16 modeCode, ViUInt16 txErrs[ ] );

### Purpose

Function Name: WA4510\_modMrtRtMd  
Description: This function modifies the error injection for a LS mode code.

### Parameter List

#### rtAddr

Variable Type : ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### modeCode

Variable Type : ViUInt16  
Control Name : modeCode  
Description : Mode code number.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### txErrs

Variable Type : ViUInt16[ ]  
Control Name : txErrs  
Description : LS and HS error injection definition for TX messages.  
This is an array of 3 x 16 bit input values defining the LS errors.  
Variable Type: ViUInt16

txErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

txErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
For further information see software driver library manual.

txErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003  
Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modMrtRtSa

### Function

ViUInt16 WA4510\_modMrtRtSa (ViUInt16 rtAddr, ViUInt16 subAdd, ViUInt16 saType,  
ViUInt16 wrap, ViUInt16 rxWCnt, ViUInt16 rxErrs[ ], ViUInt16 txWCnt,  
ViUInt16 txData[ ], ViUInt16 txErrs[ ], ViUInt16 hsRiTm, ViUInt16 hsTiTm);

### Purpose

Function Name: WA4510\_modMrtRtSa  
Description: Modifies a previously created RT sub-address in MRT mode.

### Parameter List

#### rtAddr

Variable Type : ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### subAdd

Variable Type : ViUInt16  
Control Name : subAdd  
Description : RT sub-address.  
Variable Type : ViUInt16  
Valid values : 0 to 30 for Low speed  
1 to 127 for High speed  
Default : 1

#### saType

Variable Type : ViUInt16  
Control Name : saType  
Description : Type of sub-address (LS or HS).  
Variable Type : ViUInt16  
Values:

Driver constant	Value	Description
LS_SA	1	LS sub-address
HS_SA	2	HS sub-address

Default: LS\_SA 1 LS sub-address

#### wrap

Variable Type : ViUInt16  
Control Name: wrap  
Description: Mode for data buffer. If wrap enabled then the TX and RX buffers will be the same.  
If not enabled then separate buffers will be created.  
Variable Type : ViUInt16  
Values:

Driver constant	Value	Description
NO_WRAP	1	Wrap disabled
WRAP	2	Wrap enabled

Default: NO\_WRAP 1 Wrap disabled

#### rxWCnt

Variable Type : ViUInt16  
Control Name : rxWCnt  
Description : Number of RX data words expected for this RT sub-address  
Variable Type : ViUInt16  
Default : 0

rxErrs

Variable Type ViUInt16[ ]  
Control Name : rxErrs  
Description : LS and HS error injection definition for RX messages. This is an array of 7 x 16 bit input values defining the LS and HS RX errors.  
Variable Type: ViUInt16

rxErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

rxErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
For further information see software driver library manual.

rxErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

rxErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

rxErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection.  
For further information see software driver library manual.

rxErrs[5]: HS gate error position.

rxErrs[6]: This double word defines the bit position for a 3910 bit error.  
For further information see user manual.

Default : 0

txWCnt

Variable Type ViUInt16  
Control Name : txWCnt  
Description : Number of TX data words expected for this RT sub-address  
Variable Type : ViUInt16  
Default : 0

txData

Variable Type ViUInt16[ ]  
Control Name : txData  
Description : This is an input array of txWCnt 16 bit words for initialising the TX data buffer.  
Variable Type : ViUInt16  
Default : 0

txErrs

Variable Type ViUInt16[ ]  
Control Name : txErrs  
Description : LS and HS error injection definition for TX messages. This is an array of 7 x 16 bit input values defining the LS and HS TX errors.  
Variable Type: ViUInt16

txErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

txErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
For further information see software driver library manual.

txErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

txErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

txErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection.  
For further information see software driver library manual.

txErrs[5]: HS gate error position.

txErrs[6]: This double word defines the bit position for a 3910 bit error.  
For further information see user manual.

Default : 0

hsRiTm

Variable Type : ViUInt16  
Control Name : :hsRiTm  
Description : This is the HS RI timeout value in uS.  
Variable Type : ViUInt16  
Valid Values : 30 to 255 (uS)  
Default : 0

hsTiTm

Variable Type : ViUInt16  
Control Name : hsTiTm  
Description : This is the HS TI time value in uS.  
Variable Type : ViUInt16  
Valid Values : 10 to 255 (uS)  
Default : 0

Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_modMsg

### Function

ViUInt16 WA4510\_modMsg (ViUInt32 msgID, ViUInt16 msgInfo[ ], ViUInt16 dataCount, ViUInt16 dataWords[ ]);

### Purpose

Function Name: WA4510\_modMsg  
Description: Modifies a BC message.

### Parameter List

#### msgID

Variable Type ViUInt32  
Control Name : msgID  
Description : Message handle of previously created message. This is a single 32 bit input.  
Variable Type : ViUInt32

#### msgInfo

Variable Type ViUInt16[ ]  
Control Name : msgInfo  
Description : Information describing the parts of the message to be modified.  
This is an array of 20 x 16 bit input values.  
Variable Type : ViUInt16

msgInfo[0]: Message type.

Values:	Driver constant	Value	Description
	TXMD_WODA_3838	0x8000	TX mode no data
	RXMD_WIDA_3838	0x0001	RX mode + data
	TXMD_WIDA_3838	0x8001	TX mode + data
	RT_RT_3838	0x0002	3838 RT to RT
	BC_RT_3838	0x0003	3838 BC to RT
	RT_BC_3838	0x8003	3838 RT to BC
	BC_RT_3910	0x0004	3910 BC to RT
	RT_BC_3910	0x8005	3910 RT to BC
	RT_RT_3910	0x0006	3910 RT to RT
	MODE_3910	0x0007	3910 Mode code
	TX_MSG_3910	0x8007	3910 TX message
	B_MODE_WODA_3838	0x8008	Bcast mode no data
	B_RXMD_WIDA_3838	0x0009	Bcast mode + data
	B_RT_RT_3838	0x000A	Bcast 3838 RT to RT
	B_BC_RT_3838	0x000B	Bcast 3838 BC to RT
	B_BC_RT_3910	0x000C	Bcast 3910 BC to RT
	B_RT_RT_3910	0x000E	Bcast 3910 RT to RT
	B_MODE_3910	0x000F	Bcast 3910 mode code

msgInfo[1]: LsIMsgGap - LS inter-message gap-time

Values: 4 to 65535 uS.

msgInfo[2]: LsBus - LS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 1553 bus
	SECONDARY_BUS	1	Secondary 1553 bus

msgInfo[3]: RT1 - 1st RT number.

Values: 0 to 31.



msgInfo[4]: Sub-address for RT1  
Values: 1 to 30 for 3838.  
1 to 127 for 3910.

msgInfo[5]: RT2 - 2nd RT number.  
Values: 0 to 31.

msgInfo[6]: Sub-address for RT2  
Values: 1 to 30 for 3838.  
1 to 127 for 3910.

msgInfo[7]: Message word-count.

msgInfo[8]: 3838 mode code number (if message type is LS mode code).

msgInfo[9]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

msgInfo[10]: LS error injection info.

This contains extra information to describe the 3838 error injection.  
For further information see software driver library manual.

msgInfo[11]: LS error position info.

This contains extra information defining the word number in the phase of the message for the error injection.  
For further information see software driver library manual.

msgInfo[12]: LS error phase info.

This defines the phase of the message for the error.

Values:	Driver constant	Value	Description
	ERR_DISABLED	0	No error injection
	ERR_FIR_BC_TX	1	Inject in 1st BC TX.
	ERR_SEC_BC_TX	2	Inject in 2nd BC TX.
	ERR_FIR_RT_TX	3	Inject in 1st RT TX.
	ERR_SEC_RT_TX	4	Inject in 2nd RT TX

For further information see software driver library manual.

msgInfo[13] HsRtRtIMsgGap - 3910 RT to RT inter-message gap-time.  
Values: 10 to 65535 uS.

msgInfo[14]: HsBus - HS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 3910 bus
	SECONDARY_BUS	1	Secondary 3910 bus

msgInfo[15]: 3910 error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

msgInfo[16]: HS error injection info.

This contains extra information to describe the 3910 error injection.

For further information see software driver library manual.

msgInfo[17]: HS gate error position.

msgInfo[18]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.

msgInfo[19]: HsRiTmOut - HS RT/TI time in uS.

#### dataCount

Variable Type : ViUInt16  
Control Name : dataCount  
Description : Size of data buffer required for message.  
Variable Type : ViUInt16

#### dataWords

Variable Type : ViUInt16[ ]  
Control Name : dataWords  
Description : Array of 16 bit values of size 'dataCount' to be copied to the data buffer.  
Variable Type : ViUInt16

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdBcMrt

### Function

ViUInt16 WA4510\_rdBcMrt (ViUInt16 rtAddr, ViUInt16 info[ ]);

### Purpose

Function Name: WA4510\_rdBcMrt  
Description: Reads the selected MRT table when the card is in BC mode.

### Parameter List

#### rtAddr

Variable Type : ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid Range : 0 to 31  
Default Value : 0

#### info

Variable Type : ViUInt16[ ]  
Control Name : info  
Description : Values for read from table. This is an array of 5 x 16-bit output values  
Return Type : ViUInt16  
Valid Range : info[0]: 0 = RT monitored, 1 = RT simulated.  
info[1]: RT status word from table.  
info[2]: RT vector word from table.  
info[3]: RT BIT word for 1553 TX BIT WORD from table.  
info[4]: RT BIT word for 3910 TX BIT WORD from table.

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdCycle

### Function

ViUInt16 WA4510\_rdCycle (ViUInt32 cycleID, ViUInt16 msgCount, ViUInt32 msgList[ ]);

### Purpose

Function Name: WA4510\_rdCycle  
Description: Read a previously created BC cycle.

### Parameter List

#### cycleID

Variable Type: ViUInt32  
Control Name: cycleID  
Description: Cycle handle for previously created cycle.  
Variable Type: ViUInt32

#### msgCount

Variable Type: ViUInt16  
Control Name: msgCount  
Description: Maximum number of message handles to be saved in msgList[ ].  
Variable Type: ViUInt16

#### msgList

Variable Type: ViUInt32[ ]  
Control Name: msgList  
Description: This is an output array of minimum size 'msgCount'.  
The cycle message IDs will be returned in this array.  
Return Type: ViUInt32

### Return Value

Control Name: error  
Description: Returned error codes.  
Return Type: ViUInt16  
Valid Range: 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdFrame

### Function

ViUInt16 WA4510\_rdFrame (ViUInt32 frameID, ViUInt16 cycCount, ViUInt32 cycList[ ]);

### Purpose

Function Name: WA4510\_rdFrame  
Description: Read a previously created BC frame.

### Parameter List

#### frameID

Variable Type: ViUInt32  
Control Name: frameID  
Description: Frame handle for previously created frame.  
Variable Type: ViUInt32

#### cycCount

Variable Type: ViUInt16  
Control Name: cycCount  
Description: Maximum number of cycles handles to be saved in cycList.  
Variable Type: ViUInt16

#### cycList

Variable Type: ViUInt32[ ]  
Control Name: cycList  
Description: This is an output array of minimum size 'cycCount'.  
The frame cycle IDs will be returned in this array.  
Return Type: ViUInt32

### Return Value

Control Name: error  
Description: Returned error codes.  
Return Type: ViUInt16  
Valid Range: 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdMrtRt

### Function

ViUInt16 WA4510\_rdMrtRt (ViUInt16 rtAddr, ViUInt16 info[ ]);

### Purpose

Function Name: WA4510\_rdMrtRt  
Description: Reads the selected MRT table when the card is in MRT mode.

### Parameter List

rtAddr

Variable Type ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

info

Variable Type ViUInt16[ ]  
Control Name : info  
Description : Values for modification. This is an array of 10 x 16 bit input values.  
Return Type : ViUInt16

info[0]: 0 = RT monitored, 1 = RT simulated.

info[1]: RT status word to be transmitted.

info[2]: RT vector word for a TX VECTOR mode code.

info[3]: RT BIT word to be transmitted for 1553 TX BIT WORD mode.

info[4]: RT BIT word to be transmitted for 3910 TX BIT WORD mode.

info[5]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error
	NO_3838_RESP_ERR	0x85	No response error

info[6]: LS error injection info.

This contains extra information to describe the 3838 error injection.

For further information see software driver library manual.

info[7]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

info[8]: Disable Error Bus definition.

This defines which 3838 bus the errors are allowed to be injected on.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	No errors on PRIMARY
	SECONDARY_BUS	1	No errors on SECONDARY
	BOTH_BUSES	2	No errors on any BUS
	NO_BUSES	3	Enable on any BUS

info[9]: Disable HS Mode Codes.

This defines which HS mode codes are disabled for the RT.

Values:	Driver constant	Value	Description
	HS_MODES_OFF	1	Disable all mode codes
	TXTYPE	2	Disable TX mode codes
	RXTYPE	3	Disable RX mode codes
	HS_MODES_ON	4	All mode codes enabled

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdMrtRtMd

### Function

ViUInt16 WA4510\_rdMrtRtMd (ViUInt16 rtAddr, ViUInt16 modeCode, ViUInt16 txErrs[ ] );

### Purpose

Function Name: WA4510\_rdMrtRtMd  
Description: This function reads the error injection for a LS mode code.

### Parameter List

#### rtAddr

Variable Type ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### modeCode

Variable Type ViUInt16  
Control Name : modeCode  
Description : Mode code number.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### txErrs

Variable Type ViUInt16[ ]  
Control Name : txErrs  
Description : LS and HS error injection definition for TX messages.  
This is an array of 3 x 16 bit input values defining the LS errors.  
Return Type : ViUInt16

txErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

txErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
For further information see software driver library manual.

txErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003  
Possible error codes and descriptions are as in appendix A (see pages 53 to 54).



## WA4510\_rdMrtRtSa

### Function

ViUInt16 WA4510\_rdMrtRtSa (ViUInt16 rtAddr, ViUInt16 subAdd, ViUInt16 saType, ViUInt16 \*wrap,  
ViUInt16 rxWCnt, ViUInt16 rxData[ ], ViUInt16 rxErrs[ ], ViUInt16 txWCnt,  
ViUInt16 txData[ ], ViUInt16 txErrs[ ], ViUInt16 \*hsRiTm, ViUInt16 \*hsTiTm);

### Purpose

Function Name: WA4510\_rdMrtRtSa  
Description: Reads a previously created RT sub-address in MRT mode.

### Parameter List

#### rtAddr

Variable Type ViUInt16  
Control Name : rtAddr  
Description : RT address.  
Variable Type : ViUInt16  
Valid values : 0 to 31  
Default : 0

#### subAdd

Variable Type ViUInt16  
Control Name : subAdd  
Description : RT sub-address.  
Variable Type : ViUInt16  
Valid values : 0 to 30 for Low speed  
1 to 127 for High speed  
Default : 1

#### saType

Variable Type ViUInt16  
Control Name : saType  
Description : Type of sub-address (LS or HS).  
Variable Type : ViUInt16  
Values:  

Driver constant	Value	Description
LS_SA	1	LS sub-address
HS_SA	2	HS sub-address

  
Default: LS\_SA 1 LS sub-address

#### wrap

Variable Type ViUInt16 (passed by reference)  
Control Name: wrap  
Description: This output returns the wrap mode of the RT sub-address  
Return Type: ViUInt16  
Values:  

Driver constant	Value	Description
NO_WRAP	1	Wrap disabled
WRAP	2	Wrap enabled

#### rxWCnt

Variable Type ViUInt16  
Control Name : rxWCnt  
Description : Number of RX data words to be read into rxData.  
Variable Type : ViUInt16

rxData

Variable Type ViUInt16[ ]  
 Control Name : rxData  
 Description : This is an output array of minimum length rxWCnt.  
 The RX data buffer will be copied into this array.  
 Variable Type : ViUInt16  
 Default : 0

rxErrs

Variable Type ViUInt16[ ]  
 Control Name : rxErrs  
 Description : LS and HS error injection definition for RX messages.  
 This is an array of 7 x 16 bit input values defining the LS and HS RX errors.  
 Variable Type: ViUInt16

rxErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	ESP_TM_ERR	0x84	Response time error

rxErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
 For further information see software driver library manual.

rxErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
 For further information see software driver library manual.

rxErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

rxErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection.  
 For further information see software driver library manual.

rxErrs[5]: HS gate error position.

rxErrs[6]: This double word defines the bit position for a 3910 bit error.  
 For further information see user manual.

Default : 0

txWCnt

Variable Type ViUInt16  
Control Name : txWCnt  
Description : Number of TX data words to be read into txData.  
Variable Type : ViUInt16  
Default : 0

txData

Variable Type ViUInt16[ ]  
Control Name: txData  
Description : This is an output array of minimum length txWCnt.  
The TX data buffer will be copied into this array.  
Return Type : ViUInt16

txErrs

Variable Type ViUInt16[ ]  
Control Name : txErrs  
Description : LS and HS error injection definition for TX messages.  
This is an array of 7 x 16 bit input values defining the LS and HS TX errors.  
Variable Type: ViUInt16

txErrs[0]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

txErrs[1]: LS error injection info.

This contains extra information describing the 3838 error injection.  
For further information see software driver library manual.

txErrs[2]: LS error position info.

This contains extra information defining the word number of the message for the error injection.  
For further information see software driver library manual.

txErrs[3]: HS error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

txErrs[4]: HS error injection info.

This contains extra information to describe the 3910 error injection.  
For further information see software driver library manual.

txErrs[5]: HS gate error position.

txErrs[6]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.

#### hsRiTm

Variable Type ViUInt16 (passed by reference)

Control Name : hsRiTm

Description : This is the HS RI timeout value in uS.

Return Type : ViUInt16

#### hsTiTm

Variable Type ViUInt16 (passed by reference)

Control Name : hsTiTm

Description : This is the HS TI time value in uS.

Return Type : ViUInt16

#### Return Value

Control Name : error

Description : Returned error codes.

Return Type : ViUInt16

Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_rdMsg

### Function

ViUInt16 WA4510\_rdMsg (ViUInt32 msgID, ViUInt16 msgInfo[ ], ViUInt16 dataWords[ ], ViUInt16 dataCount);

### Purpose

Function Name: WA4510\_rdMsg  
Description: Read an existing BC message.

### Parameter List

#### msgID

Variable Type ViUInt32  
Control Name : msgID  
Description : Message handle of previously created message. This is a single 32 bit input.  
Variable Type : ViUInt32

#### msgInfo

Variable Type ViUInt16[ ]  
Control Name : msgInfo  
Description : Information describing the message. The information describing the message will be returned in this array of 20 x 16 bit output values.  
Return Type : ViUInt16

msgInfo[0]: Message type.

Values:	Driver constant	Value	Description
	TXMD_WODA_3838	0x8000	TX mode no data
	RXMD_WIDA_3838	0x0001	RX mode + data
	TXMD_WIDA_3838	0x8001	TX mode + data
	RT_RT_3838	0x0002	3838 RT to RT
	BC_RT_3838	0x0003	3838 BC to RT
	RT_BC_3838	0x8003	3838 RT to BC
	BC_RT_3910	0x0004	3910 BC to RT
	RT_BC_3910	0x8005	3910 RT to BC
	RT_RT_3910	0x0006	3910 RT to RT
	MODE_3910	0x0007	3910 Mode code
	TX_MSG_3910	0x8007	3910 TX message
	B_MODE_WODA_3838	0x8008	Bcast mode no data
	B_RXMD_WIDA_3838	0x0009	Bcast mode + data
	B_RT_RT_3838	0x000A	Bcast 3838 RT to RT
	B_BC_RT_3838	0x000B	Bcast 3838 BC to RT
	B_BC_RT_3910	0x000C	Bcast 3910 BC to RT
	B_RT_RT_3910	0x000E	Bcast 3910 RT to RT
	B_MODE_3910	0x000F	Bcast 3910 mode code

msgInfo[1]: LsIMsgGap - LS inter-message gap-time

Values: 4 to 65535 uS.

msgInfo[2]: LsBus - LS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 1553 bus
	SECONDARY_BUS	1	Secondary 1553 bus

msgInfo[3]: RT1 - 1st RT number.

Values: 0 to 31.

msgInfo[4]: Sub-address for RT1  
Values: 1 to 30 for 3838.  
1 to 127 for 3910.

msgInfo[5]: RT2 - 2nd RT number.  
Values: 0 to 31.

msgInfo[6]: Sub-address for RT2  
Values: 1 to 30 for 3838.  
1 to 127 for 3910.

msgInfo[7]: Message word-count.

msgInfo[8]: 3838 mode code number (if message type is LS mode code).

msgInfo[9]: 3838 error injection.

Values:	Driver constant	Value	Description
	NO_LS_ERRS	0x00	No errors
	PARITY_ERR	0x01	Parity error
	MANCHESTER_ERR	0x02	Manchester error
	SYNCHRO_ERR	0x03	Sync pattern error
	WRD_LEN_ERR	0x04	Word-length error
	WRONG_BUS_ERR	0x80	Wrong bus error
	BOTH_BUS_ERR	0x81	Both buses error
	POS_WRD_CNT_ERR	0x82	+ve word-count error
	NEG_WRD_CNT_ERR	0x83	-ve word-count error
	RESP_TM_ERR	0x84	Response time error

msgInfo[10]: LS error injection info.

This contains extra information to describe the 3838 error injection.  
For further information see software driver library manual.

msgInfo[11]: LS error position info.

This contains extra information defining the word number in the phase of the message for the error injection.  
For further information see software driver library manual.

msgInfo[12]: LS error phase info.

This defines the phase of the message for the error.

Values:	Driver constant	Value	Description
	ERR_DISABLED	0	No error injection
	ERR_FIR_BC_TX	1	Inject in 1st BC TX.
	ERR_SEC_BC_TX	2	Inject in 2nd BC TX.
	ERR_FIR_RT_TX	3	Inject in 1st RT TX.
	ERR_SEC_RT_TX	4	Inject in 2nd RT TX

For further information see software driver library manual.

msgInfo[13] HsRtRtIMsgGap - 3910 RT to RT inter-message gap-time.

Values: 10 to 65535 uS.

msgInfo[14]: HsBus - HS bus definition.

Values:	Driver constant	Value	Description
	PRIMARY_BUS	0	Primary 3910 bus
	SECONDARY_BUS	1	Secondary 3910 bus

msgInfo[15]: 3910 error injection.

Values:	Driver constant	Value	Description
	NO_HS_ERRS	0x00	No HS error injection.
	PRE_BIT_CNERR	0x01	Preamble bit count.
	NO_3910_RE_ERR	0x02	No 3910 response.
	NEG_3910_W_CNT_ERR	0x04	-ve word-count error.
	POS_3910_W_CNT_ERR	0x08	+ve word-count error.
	FCS_ERR	0x10	FCS error.
	GATE_ERR	0x20	Bit error.

msgInfo[16]: HS error injection info.

This contains extra information to describe the 3910 error injection.

For further information see software driver library manual.

msgInfo[17]: HS gate error position.

msgInfo[18]: This double word defines the bit position for a 3910 bit error.

For further information see user manual.

msgInfo[19]: HsRiTmOut - HS RT/TI time in uS.

#### dataWords

Variable Type : ViUInt16[ ]  
Control Name : dataWords  
Description : Array of 16 bit values of size 'dataCount' to be copied to the data buffer.  
Return Type : ViUInt16

#### dataCount

Variable Type : ViUInt16  
Control Name : dataCount  
Description : Size of data buffer required for message.  
Variable Type : ViUInt16

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_readCLK

### Function

ViUInt16 WA4510\_readCLK (ViUInt32 \*result);

### Purpose

Function Name: WA4510\_readCLK  
Description: Reads the current value of the on-board 10uS resolution clock.

### Parameter List

result

Variable Type ViUInt32 (passed by reference)  
Control Name : result  
Description : Reports the 32 bit value of the clock.  
Return Type : ViUInt32

*Note: Clock resolution is 10 microseconds.*

Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003  
Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_revision

ViUInt16 WA4510\_revision (ViChar driver\_rev[ ], ViChar cvi\_rev[ ]);

### Purpose

Function Name: WA4510\_revision  
Description: Enables revision levels of "C" drivers and CVI drivers to be interrogated and displayed.

### Parameter List

driver\_rev

Variable Type ViChar[ ]  
Control Name : driver\_rev  
Description : Revision level of "C" drivers.  
Return Type : ViChar[ ]

cvi\_rev

Variable Type ViChar[ ]  
Control Name : cvi\_rev  
Description : Revision level of CVI drivers.  
Return Type : ViChar[ ]

Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003  
Possible error codes and descriptions are as in appendix A (see pages 53 to 54).



## WA4510\_runFrame

### Function

ViUInt16 WA4510\_runFrame (ViUInt32 frameID, ViUInt16 count);

### Purpose

Function Name: WA4510\_runFrame  
Description: Runs a previously created BC frame.

### Parameter List

#### frameID

Variable Type ViUInt32  
Control Name : frameID  
Description : Handle for frame to be transmitted.  
Variable Type : ViUInt32  
Default :

#### count

Variable Type ViUInt16  
Control Name : count  
Description : Number of frames to be transmitted.  
Variable Type: ViUInt16  
Valid Values : 1 to 65535      Special case : 0 = transmit forever  
Default :

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_runMON

ViUInt16 WA4510\_runMON (void);

### Purpose

Function Name: WA4510\_runMON  
Description: This function runs the card in MON mode.

#### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_runMRT

### Function

ViUInt16 WA4510\_runMRT (void);

### Purpose

Function Name: WA4510\_runMRT  
Description: This function runs the card in MRT mode.

### Return Value

Control Name : error  
Description : Returned error codes.  
Return Type : ViUInt16  
Valid Range : 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

---

## WA4510\_rWord

ViInt16 WA4510\_rWord (ViUInt32 address);

### Purpose

Function Name: WA4510\_rWord  
Description: Reads a word from the card. The value is returned as a 16 bit value

### Parameter List

address

Variable Type ViUInt32  
Control Name : address  
Description : 32 bit address offset from start of card.  
Variable Type : ViUInt32

### Return Value

Control Name : value  
Description : 16 bit returned value of read function.  
Return Type : ViInt16

## WA4510\_selfTest

### Function

ViUInt16 WA4510\_selfTest (ViUInt16 \*result);

### Purpose

Function Name: WA4510\_selfTest  
Description: Executes self-test and reports the result.

### Parameter List

result

Variable Type: ViUInt16 (passed by reference)  
Control Name: result  
Description: Reports result of self-test. This is a single 16 bit output describing the result of the self-test. If the self-test has passed the value 0x8008 is returned. If the value is not 0x8008 self-test failed, see below for meaning of result.  
Return Type: ViUInt16  
Valid Range: 0x8008 to 0x9FF8  
Possible values: 0x8008 self-test passed  
Single failures: 0x9008 3838 interface test failed  
0x8808 3910 interface test failed  
0x8408 Frame counter test failed  
0x8208 Local clock test failed  
0x8108 Memory test 5 failed  
0x8088 Memory test 4 failed  
0x8048 Memory test 3 failed  
0x8028 Memory test 2 failed  
0x8018 Memory test 1 failed

Note: Several bits may be set simultaneously, if multiple failures are found.

Register contents will be as follows:

D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00  
1 0 0 LS HS FR LC M5 M4 M3 M2 M1 1 0 0 0

LS = 1 3838 interface test failed  
HS = 1 3910 interface test failed  
FR = 1 Frame counter test failed  
LC = 1 Local clock test failed  
M5 = 1 Memory test 5 failed  
M4 = 1 Memory test 4 failed  
M3 = 1 Memory test 3 failed  
M2 = 1 Memory test 2 failed  
M1 = 1 Memory test 1 failed

### Return Value

Control Name: error  
Description: Returned error codes.  
Return Type: ViUInt16  
Valid Range: 0 to 8003

Possible error codes and descriptions are as in appendix A (see pages 53 to 54).

## WA4510\_wWord

### Function

void WA4510\_wWord (ViUInt32 address, ViUInt16 value);

### Purpose

Function Name: WA4510\_wWord

Description: Writes a word to the card, as a 16 bit value to a defined 32 bit start address.

### Parameter List

address

Variable Type ViUInt32

Control Name : address

Description : 32 bit address offset from start of card.

Variable Type : ViUInt32

value

Variable Type ViUInt16

Control Name : value

Description : 16 bit value to be written to card.

Variable Type : ViInt16

## Appendix A.

### Error Codes.

Possible error codes and descriptions are as follows:

No.	MNEMONIC	DESCRIPTION
0	E_NO_ERROR	No error
10	E_INV_CARDHANDLE	Card handle is not valid
11	E_INV_INFO	Invalid information field
12	E_INV_RTNUM	Invalid RT number
20	E_NOT_BCMRT_MODE	Card is not in BCMRT mode
21	E_NOT_MRT_MODE	Card is not in MRT mode
22	E_NOT_CM_MODE	Card is not in MON mode
30	E_CARD_RUNNING	Card is already running
31	E_CARD_HALTED	Card is halted
32	E_CANNOT_ISSUE_CMD	Card not responding to command
40	E_ALLOC_CYCLEL	Cycle allocation error
41	E_ALLOC_FRAMEL	Frame allocation error
42	E_ALLOC_SYSBLOCKL	System block allocation error
43	E_ALLOC_GAPSCYAREAL	System block area allocation error
44	E_ALLOC_SAMDL	Sub-address/Mode allocation error
45	E_ALLOC_DATAGAPL	Data buffer allocation error
46	E_ALLOC_SYSGAPL	System gap allocation error
47	E_ALLOC_DATAMEM	Data memory allocation error
48	E_ALLOC_DEFAULTS	Default value allocation error
50	E_DATA_AREA_FULL	Data allocation exceeded maximum
51	E_SYS_AREA_FULL	System allocation exceeded maximum
100	E_INV_CARDTYPE	Invalid card type
101	E_INV_OPMODE	Invalid operating mode for function
102	E_INV_CARDADD	Invalid card address
200	E_INV_MSGID	Invalid message ID
201	E_INV_MSGDATA	Invalid message DATA
210	E_INV_CYCLEID	Invalid cycle ID
211	E_INV_CYCLEDATA	Invalid cycle DATA
212	E_CYCLE_AREA_FULL	Cycle allocation exceeded maximum
220	E_INV_FRAMEID	Invalid frame ID
221	E_INV_FRAMEDATA	Invalid frame DATA
222	E_FRAME_AREA_FULL	Framer allocation exceeded maximum
240	E_INV_COUNT	Invalid frame TX count

310	E_INV_SA	Invalid sub-address
311	E_INV_SATYPE	Invalid sub-address type
312	E_SA_NOT_FOUND	Sub-address not found
313	E_RT_NOT_FOUND	RT not found
320	E_INV_MD	Invalid mode code
321	E_MD_NOT_FOUND	Mode code not found
400	E_CM_RUNNING	Chron mon still running
401	E_CM_TRIG_DEF_ERR	Trigger definition error
402	E_CM_TRIG_SEQ_ERR	Trigger sequence error
403	E_CM_NO_TRIG	Trigger not set-up
404	E_CM_SRCH_ERR	Invalid search parameters
405	E_CM_SRCH_MAX	Maximum search count expired
406	E_CM_NO_MESSAGE	No messages available
407	E_CM_STOPPED	Chron mon has stopped
500	E_INV_CMND	Invalid command
600	E_SELFTEST_FAILED	Self-test has failed
601	E_CARD_NOT_PRESENT	Card is not present
602	E_INV_CLOCK	Invalid clock value
700	E_DE_MON_SETUP_ERR	Dassault monitor set-up error
701	E_DE_MON_REPORT_ERR	Dassault buffer report error
2010	E_NO_SUITABLE_GAP	No suitable gap found
2020	E_CM_SYNTAX_ERR	Chron mon syntax error
3000	E_DRIVER_INIT_FAILED	Failed to initialise Win driver
8000	E_FILE_OPEN	Cannot open file
8001	E_FILE_WRITE	Cannot write to file
8002	E_FILE_READ	Cannot read from file
8003	E_WRONG_FILE_TYPE	File type to mode mismatch.

## INDEX OF FUNCTIONS

WA4510_close	4
WA4510_crCycle	4
WA4510_crFrame	5
WA4510_crMrtRtSa	6
WA4510_crMsg	10
WA4510_defSeq	13
WA4510_defTrigs	14
WA4510_exeCommand	15
WA4510_findMsgs	16
WA4510_getMsg	17
WA4510_getRange	18
WA4510_haltBC	19
WA4510_haltMON	19
WA4510_init	20
WA4510_modBcMrt	22
WA4510_modCycle	23
WA4510_modFrame	24
WA4510_modMrtRt	25
WA4510_modMrtRtMd	27
WA4510_modMrtRtSa	28
WA4510_modMsg	32
WA4510_rdBcMrt	35
WA4510_rdCycle	36
WA4510_rdFrame	37
WA4510_rdMrtRt	38
WA4510_rdMrtRtMd	40
WA4510_rdMrtRtSa	41
WA4510_rdMsg	45
WA4510_readCLK	48
WA4510_revision	48
WA4510_runFrame	49
WA4510_runMON	49
WA4510_runMRT	50
WA4510_rWord	50
WA4510_selfTest	51
WA4510_wWord	52