



Technical support:  
+49 (0)7223 / 9493-0

**Software description**

**ADDIDRIVER**

**Common functions**

Edition: 08.01 – 02/2006

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
<b>2</b>	<b>SOFTWARE FUNCTIONS.....</b>	<b>4</b>
<b>2.1</b>	<b>Common functions .....</b>	<b>4</b>
	1) i_ADDIDATA_OpenWin32Driver (..) .....	4
	2) i_ADDIDATA_GetCurrentDriverHandle (..) .....	6
	3) v_ADDIDATA_GetDriverVersion (..) .....	7
	4) b_ADDIDATA_GetLocalisation (..) .....	8
	5) b_ADDIDATA_CloseWin32Driver (..) .....	11
<b>2.2</b>	<b>Error .....</b>	<b>12</b>
	1) i_ADDIDATA_GetLastError (...) .....	12
	2) i_ADDIDATA_GetLastErrorAndSource (...) .....	13
	3) b_ADDIDATA_EnableErrorMessage (...) .....	30
	4) b_ADDIDATA_DisableErrorMessage (...) .....	32
	5) b_ADDIDATA_FormatErrorMessage (...) .....	34

## Tables

Table 1-1: Type Declaration for Windows 98/NT/2000/XP .....	3
Table 2-1: Return values for i_ADDIDATA_OpenWin32Driver .....	5
Table 2-2: Function number .....	14
Table 2-3: Error number .....	21

# 1 INTRODUCTION

**i**

## IMPORTANT!

Note the following conventions in the text:

Function: "b\_ADDIDATA\_GetNumberOfAnalogInputs"  
Variable *dw\_DriverHandle*

Table 1-1: Type Declaration for Windows 98/NT/2000/XP

	Borland C	Microsoft C	Borland Pascal	Microsoft Visual Basic Windows
<b>VOID</b>	void	void	pointer	any
<b>BYTE</b>	unsigned char	unsigned char	byte	integer
<b>INT</b>	int	int	integer	integer
<b>WORD</b>	unsigned short int	unsigned short int	long	long
<b>DWORD</b>	long	long	longint	long
<b>PBYTE</b>	unsigned char *	unsigned char *	var byte	integer
<b>PINT</b>	int *	int *	var integer	integer
<b>PWORD</b>	unsigned short int *	unsigned short int *	var long	long
<b>PCHAR</b>	char *	char *	var string	string
<b>PDWORD</b>	long *	long *	var longint	long
<b>DOUBLE</b>	double	double	double	double

## 2 SOFTWARE FUNCTIONS

### 2.1 Common functions

#### 1) i\_ADDIDATA\_OpenWin32Driver (..)

##### Syntax:

<Return value> = i\_ADDIDATA\_OpenWin32Driver (BYTE b\_CompilerDefine,  
PDWORD pdw\_DriverHandle)

##### Parameters:

##### - Input:

BYTE *b\_CompilerDefine*

The user has to choose the language under Windows in which he wants to program

- ADDIDATA\_DLL\_COMPILER\_C = 0:

The user programs in C.

- ADDIDATA\_DLL\_COMPILER\_VB = 2:

The user programs in Visual Basic for Windows.

- ADDIDATA\_DLL\_COMPILER\_VB\_5 or  
ADDIDATA\_DLL\_COMPILER\_VB\_6 = 4:

The user programs in Visual Basic 5 or 6 for Windows NT or Windows 95.

- ADDIDATA\_DLL\_COMPILER\_PASCAL = 1:

The user programs in Pascal or Delphi.

- ADDIDATA\_DLL\_LABVIEW = 3:

The user programs in Labview.

##### - Output:

PDWORD *pdw\_DriverHandle*

Returns the driver handle. This parameter is the first to be called up for each function.

##### Task:

This function loads the ADDIDATA driver. All information concerning the ADDI-DATA boards is read. The user choose the language in which he/she wants to program through the parameter *b\_CompilerDefine*.

##### Calling convention:

##### ANSI C:

INT i\_ReturnValue;  
DWORD dw\_DriverHandle;

i\_ReturnValue = i\_ADDIDATA\_OpenWin32Driver (ADDIDATA\_DLL\_COMPILER\_C,  
&dw\_DriverHandle);

##### Return value:

0: No error

-100.....: See Table 2-1

Table 2-1: Return values for i\_ADDIDATA\_OpenWin32Driver

Constant	Return value	Meaning
<b>Common function errors</b>		
ADDIDATA_FUNCTIONALITY_NOT_AVAILABLE	-100	Functionality not available
ADDIDATA_FUNCTIONALITY_NO_RING_0	-101	This function cannot be called up in the synchronous mode
ADDIDATA_SYSTEM_ERROR	-102	Cannot detect the 32-bit system
ADDIDATA_NO_FREE_DRIVER_HANDLE_FOUND	-103	No driver handle available (up to 25 drivers can be used)
ADDIDATA_DRIVER_SHARED_MEMORY_ERROR	-104	The allocation of the user shared memory is wrong
ADDIDATA_DRIVER_OPENING_ERROR	-105	The ADDI-DATA sys/vxd driver cannot be loaded
ADDIDATA_CURRENT_PROCESS_HDL_NOT_AVAILABLE	-106	The current process handle is not available
ADDIDATA_GET_SHARED_MEMORY_ERROR	-107	Error by allocation of the shared memory
ADDIDATA_DRIVER_HANDLE_ERROR	-108	Driver handle passed to the function not available
ADDIDATA_DRIVER_NOT_OPEN	-109	Driver is not open
ADDIDATA_ONE_OR_MORE_BOARDS_NOT_FOUND	-110	One or more boards not found
ADDIDATA_DRIVER_ALREADY_OPEN	-111	Driver already open
ADDIDATA_REGISTRY_PROBLEM	-112	The registry information is not complete
ADDIDATA_ADDIDATA_SHARED_PATH_NOT_AVAILABLE	-113	The path of the shared files is not available
ADDIDATA_TEMPERATURE_CONVERT_FILE_ERROR	-114	There is an error in the temperature convert file
ADDIDATA_TEMPERATURE_BUFFER_SHARED_MEMORY_ERROR	-115	Problem by allocation of the shared memory for the converting buffer
ADDIDATA_COMPILER_DEFINED_ERROR	-116	The compiler defined is not available
ADDIDATA_MULTIPROCESS_NOT_AVAILABLE_IN_THIS_VERSION	-117	Multiprocess operation is not supported by this driver version.
ADDIDATA_READY_BIT_TIMEOUT_OCCUR	-118	Timeout occurred while ready bit was waiting.
ADDIDATA_ERROR_FILE_NOT_AVAILABLE	-119	The error file „error.dat“ and „function.dat“ are not available in the shared directory.
ADDIDATA_ERROR_STRING_TOO_SMALL	-120	The error string parameter(s) is or are too small.
ADDIDATA_ERROR_STRING_NOT_FOUND	-121	Error string not found in the error file.
ADDIDATA_REGISTRY_NOT_UPDATED	-122	The registry has not been updated. Start ADDIREG to update it.
ADDIDATA_DRIVER_FREE_SHARED_MEMORY_ERROR	-123	Error in the free shared memory
ADDIDATA_DRIVER_RING_0_PREPARATION_ERROR	-124	Error in preparation of ring 0
ADDIDATA_DRIVER_MEMORY_MAP_ERROR	-125	Memory mapping error
ADDIDATA_REGISTRY_COULD_NOT_BE_TESTED	-126	The registry could not be tested
ADDIDATA_ADDEVICEMAPPER_ALREADY_STARTED	-127	ADDEVICE MAPPER program is already started.
ADDIDATA_EXTERN_DRIVER_DLL_LOAD_ERROR	-128	External DLL file cannot be loaded
ADDIDATA_EXTERN_DRIVER_APCI_1500_DLL_NOT_FOUND	-129	External DLL file not found (only for the <b>APCI-1500</b> )
ADDIDATA_EXTERN_DRIVER_APCI_1500_DLL_VERSION_ERROR	-130	The version of the external DLL file (only for the <b>APCI-1500</b> ) is not correct.
ADDIDATA_VIRTUAL_BOARD_FILE_NOT_FOUND	-131	File of the virtual board not found
ADDIDATA_REALBOARD_XML_FILE_NOT_FOUND	-132	Real board XML file not found
ADDIDATA_REALBOARD_XML_FILE_BACKUP_ERROR	-133	Real board XML file error backup error
ADDIDATA_COPY_VIRTUAL_BOARD_FILE_ERROR	-134	Virtual board file copy error
ADDIDATA_ADDEVICEMAPER_CALL_ERROR	-135	ADDevice Mapper call error
ADDIDATA_SOAP_NEW_ERROR	-146	Soap new function error
ADDIDATA_SOCKET_WSASTARTUP_ERROR	-174	Socket startup error
ADDIDATA_SOCKET_GETHOSTBYNAME_ERROR	-175	Socket: Get host by name error
ADDIDATA_CREATE_SOCKET_ERROR	-176	Create socket error
ADDIDATA_SOCKET_CONNECTOR_ERROR	-177	Socket connect error
ADDIDATA_MUTEX_CREATION_ERROR	-178	Mutex creation error

## 2) i\_ADDIDATA\_GetCurrentDriverHandle (..)

**Syntax:**

<Return value> = i\_ADDIDATA\_GetCurrentDriverHandle  
(PDWORD pdw\_DriverHandle)

**Parameters:****- Input:**

No input parameter has occurred.

**- Output:**

PDWORD *pdw\_DriverHandle* Returns the driver handle of the process  
which calls up the function

**Task:**

Gives the driver handle of the process which calls up the function.

**Calling convention:**ANSI C:

INT i\_ReturnValue;  
DWORD dw\_DriverHandle;

i\_ReturnValue = i\_ADDIDATA\_GetCurrentDriverHandle (&dw\_DriverHandle);

**Return value:**

0 : No error has occurred.

-106 : ADDIDATA\_CURRENT\_PROCESS\_HDL\_NOT\_AVAILABLE : The current  
process handle is not available

### 3) v\_ADDIDATA\_GetDriverVersion (..)

**Syntax:**

<Return value> = v\_ADDIDATA\_GetDriverVersion  
(PDWORD pdw\_DriverVersion)

**Parameters:****- Input:**

No input signal has occurred.

**- Output:**

PDWORD *pdw\_DriverVersion* Returns the driver version in Hex form:  
0x09000210 means 0900/0210

**Task:**

This function returns the version of the ADDIDriver.

**Calling convention:**ANSI C:

DWORD dw\_DriverVersion;

v\_ADDIDATA\_GetDriverVersion (&dw\_DriverVersion);

**Return value:**

No value returned

#### 4) b\_ADDIDATA\_GetLocalisation (..)

##### Syntax:

```
<Return value> = b_ADDIDATA_GetLocalisation
                    (str_RequestInformation s_RequestInformation,
                     DWORD dw_RequestInformationStructSize,
                     pstr_LocalisationInformation ps_LocalisationInformation,
                     DWORD dw_LocalisationInformationStructSize)
```

##### Parameters

###### - Input:

str\_RequestInformation s\_RequestInformation:  
Structure for requesting the localisation information

typedef struct

```
{
    WORD w_EntityVirtualIndex;    // Virtual Index
    BYTE b_EntityType;           // ADDIDATA_LOCALISATION_CHANNEL or
                                // ADDIDATA_LOCALISATION_MODULE
    BYTE b_Reserved1;           // Not used

    WORD w_Functionality;        // ADDIDATA_NameOfTheFunctionality
    BYTE b_Reserved2[2];        // Not used
}
```

}str\_RequestInformation,\*pstr\_RequestInformation;

with

ADDIDATA\_LOCALISATION\_CHANNEL 0

ADDIDATA\_LOCALISATION\_MODULE 1

and

```
ADDIDATA_DIGITAL_INPUT    0
ADDIDATA_DIGITAL_OUTPUT   1
ADDIDATA_ANALOG_INPUT     2
ADDIDATA_ANALOG_OUTPUT    3
ADDIDATA_TIMER            4
ADDIDATA_WATCHDOG         5
ADDIDATA_TEMPERATURE       6
ADDIDATA_COUNTER          7
ADDIDATA_RESISTANCE       9
ADDIDATA_PRESSURE         11
ADDIDATA_TRANSDUCER       12
```

DWORD dw\_RequestInformationStructSize

Size of the structure requesting the localisation information

DWORD dw\_LocalisationInformationStructSize

Size of the structure containing the localisation information



**- Output:**

pstr\_LocalisationInformation ps\_LocalisationInformation

Structure containing the localisation information

typedef struct

```
{
WORD    w_ChannelRealIndex;    // Real index of the channel
WORD    w_ModuleRealIndex;    // Real Index of the module

CHAR    c_BoardName[MAX_PATH]; // Board name of the device containing
                                   the entity

DWORD   dw_BoardAddress[6];    // Base address of the device containing
                                   the entity
BYTE    b_BoardInterrupt;    // Interrupt address of the device containing
                                   the entity
BYTE    b_DeviceNbr;          // Device number (BIOS) of the device
                                   containing the entity

BYTE    b_Reserved[2];        // Not used

CHAR    c_PCISlotNumberInformation[MAX_PATH];
                                   // PCI slot of the device containing the entity

DWORD   dw_DeviceSerialNumber; // Serial number of the device containing the entity
DWORD   dw_Reserved;           // Not used

}str_LocalisationInformation,*pstr_LocalisationInformation;
```

**Task:**

Localise the entity (channel or module) of the virtual board in the real hardware configuration. Can be called up by the driver.

**Calling convention:**

ANSI C:

```
INT    i_ReturnValue;
str_RequestInformation s_RequestInformation;
str_LocalisationInformation s_LocalisationInformation;

s_RequestInformation.w_EntityVirtualIndex = 0;
s_RequestInformation.b_EntityType =
ADDIDATA_LOCALISATION_CHANNEL;
s_RequestInformation.w_Functionality = ADDIDATA_DIGITAL_INPUT;

i_ReturnValue = i_ADDIDATA_GetLocalisation
                                   (s_RequestInformation,
                                   sizeof(str_RequestInformation),
                                   &s_LocalisationInformation,
                                   sizeof(str_LocalisationInformation));
```

**Return value:**

0 : No error

ADDIDATA\_ENTITY\_COULD\_NOT\_BE\_FOUND: The entity content in the request information could not be found.

ADDIDATA\_ENTITY\_TYPE\_IS\_WRONG : The entity type is wrong.

ADDIDATA\_ENTITY\_FUNCTIONALITY\_IS\_WRONG : The entity functionality is wrong.

ADDIDATA\_REQUEST\_INFORMATION\_STRUCT\_SIZE\_IS\_WRONG :

The dw\_RequestInformationStructSize is wrong.

ADDIDATA\_LOCALISATION\_INFORMATION\_STRUCT\_SIZE\_IS\_WRONG

The dw\_LocalisationInformationStructSize is wrong.

with

-1000: ADDIDATA\_ENTITY\_COULD\_NOT\_BE\_FOUND

-1001: ADDIDATA\_ENTITY\_TYPE\_IS\_WRONG

-1002: ADDIDATA\_ENTITY\_FUNCTIONALITY\_IS\_WRONG

-1003: ADDIDATA\_REQUEST\_INFORMATION\_STRUCT\_SIZE\_IS\_WRONG

-1004: ADDIDATA\_LOCALISATION\_INFORMATION\_STRUCT\_SIZE\_IS\_WRONG

5) `b_ADDIDATA_CloseWin32Driver` (..)**i****IMPORTANT!****Call up this function each time you want to quit the user program!****Syntax:**`<Return value> = b_ADDIDATA_CloseWin32Driver (DWORD dw_DriverHandle)`**Parameters****- Input:**`DWORD dw_DriverHandle`                      Handle of the ADDI-DATA driver**- Output:**

No output signal has occurred.

**Task:**

Closes the ADDI-DATA driver and blocks the access to all boards.

**Calling convention:**ANSI C:

```
BYTE      b_ReturnValue;  
DWORD     dw_DriverHandle;
```

```
b_ReturnValue = b_ADDIDATA_CloseWin32Driver      (dw_DriverHandle);
```

**Return value:**

1: No error

0: Error by calling up the function. Use the function "`i_ADDIDATA_GetLastError`", to find the error source.

## 2.2 Error

### i

#### IMPORTANT!

Each function of the ADDIDRIVER returns a value of the type Byte.

When the return value is "False", the user can receive the error source through the functions described below.

#### 1) i\_ADDIDATA\_GetLastError (...)

##### Syntax:

```
<Return value> = i_ADDIDATA_GetLastError (DWORD dw_DriverHandle,
                                           PWORD pw_FunctionNumber,
                                           PINT pi_ErrorCode,
                                           PBYTE pb_ErrorLevel)
```

##### Parameters:

###### - Input:

DWORD *dw\_DriverHandle*      Handle of the driver

###### - Output:

PWORD *pw\_FunctionNumber*      Number of the function on which an error has occurred. See Table 2-2

PINT *pi\_ErrorCode*      Error number. See Table 2-3

PBYTE *pb\_ErrorLevel*      0: An error has occurred in the application (ring 3)  
                                  1: The function which caused the error has been called up in the user interrupt routine.  
                                  This parameter has only a meaning for 32-bit operating systems. (ring 0)

##### Task:

Transmits the last error message.

##### Calling convention:

ANSI C:

WORD      w\_FunctionNumber;

INT      i\_ErrorCode;

INT      i\_ErrorLevel;

INT      i\_ReturnValue;

```
i_ReturnValue = i_ADDIDATA_GetLastError (dw_DriverHandle,
                                           &w_FunctionNumber,
                                           &i_ErrorCode,
                                           &b_ErrorLevel);
```

##### Return value:

0: No error

1: Error message present

- 101: Function cannot be called up in the user interrupt routine

- 102: Error message not found

**2) i\_ADDIDATA\_GetLastErrorAndSource (...)****Syntax:**

<Return value> = i\_ADDIDATA\_GetLastErrorAndSource  
 (DWORD dw\_DriverHandle,  
 PWORD pw\_FunctionNumber,  
 PINT pi\_ErrorCode,  
 PBYTE pb\_ErrorLevel,  
 PDWORD pdw\_ErrorSource)

**Parameters:****- Input:**

DWORD dw\_DriverHandle      Handle of the driver

**- Output:**

PWORD pw\_FunctionNumber      Number of the function on which an error has occurred. See Table 2-2  
 PINT pi\_ErrorCode      Error number. See Table 2-3  
 PBYTE pb\_ErrorLevel      0: An error has occurred in the application (ring 3)  
                                  1: The function which caused the error has been called up in the user interrupt routine.  
                                  This parameter has only a meaning for 32-bit operating systems. (ring 0)  
 PDWORD pdw\_ErrorSource      Error source value.

**Task:**

Transmits the last error message.

**Calling convention:**ANSI C:

WORD w\_FunctionNumber;  
 INT i\_ErrorCode;  
 INT i\_ErrorLevel;  
 INT i\_ReturnValue;  
 DWORD dw\_ErrorSource;

i\_ReturnValue = i\_ADDIDATA\_GetLastErrorAndSource

(dw\_DriverHandle,  
 &w\_FunctionNumber,  
 &i\_ErrorCode,  
 &i\_ErrorLevel,  
 &dw\_ErrorSource);

**Return value:**

0: No error  
 1: Error message present  
 - 101: Function cannot be called up in the user interrupt routine  
 - 102: Error message not found

Table 2-2: Function number

Constant	Return value	Function
<b>Common function</b>		
ADDIDATA_OPEN_WIN32_DRIVER	10	i_ADDIDATA_OpenWin32Driver
ADDIDATA_CLOSE_WIN32_DRIVER	11	b_ADDIDATA_CloseWin32Driver
ADDIDATA_GET_CURRENT_DRIVER_HANDLE	14	i_ADDIDATA_GetCurrentDriverHandle

**Interrupt**

ADDIDATA_SET_FUNCTIONALITY_INT_ROUTINE	50	b_ADDIDATA_SetFunctionalityIntRoutineWin32
ADDIDATA_TEST_INTERRUPT	51	b_ADDIDATA_TestInterrupt
ADDIDATA_RESET_FUNCTIONALITY_INT_ROUTINE	52	b_ADDIDATA_ResetFunctionalityIntRoutine

**Error Messages**

ADDIDATA_GET_LAST_ERROR_MESSAGE	100	i_ADDIDATA_GetLastError
ADDIDATA_ENABLE_ERROR_MESSAGE	101	b_ADDIDATA_EnableErrorMessage
ADDIDATA_DISABLE_ERROR_MESSAGE	102	b_ADDIDATA_DisableErrorMessage
ADDIDATA_FORMAT_ERROR_MESSAGE	103	b_ADDIDATA_FormatErrorMessage

**Timer**

ADDIDATA_GET_NUMBER_OF_TIMERS	200	b_ADDIDATA_GetNumberOfTimers
ADDIDATA_GET_TIMER_INFORMATION	201	b_ADDIDATA_GetTimerInformation
ADDIDATA_INIT_TIMER	202	b_ADDIDATA_InitTimer
ADDIDATA_ENABLE_DISABLE_TIMER_INTERRUPT	203	b_ADDIDATA_EnableDisableTimerInterrupt
ADDIDATA_START_TIMER	204	b_ADDIDATA_StartTimer
ADDIDATA_START_ALL_TIMERS	205	b_ADDIDATA_StartAllTimers
ADDIDATA_TRIGGER_TIMER	206	b_ADDIDATA_TriggerTimer
ADDIDATA_TRIGGER_ALL_TIMERS	207	b_ADDIDATA_TriggerAllTimers
ADDIDATA_STOP_TIMER	208	b_ADDIDATA_StopTimer
ADDIDATA_STOP_ALL_TIMERS	209	b_ADDIDATA_StopAllTimers
ADDIDATA_READ_TIMER_VALUE	210	b_ADDIDATA_ReadTimerValue
ADDIDATA_READ_TIMER_STATUS	211	b_ADDIDATA_ReadTimerStatus
ADDIDATA_ENABLE_DISABLE_TIMER_HARDWARE_GATE	212	b_ADDIDATA_EnableDisableTimerHardwareGate
ADDIDATA_GET_TIMER_HARDWARE_GATE_STATUS	213	b_ADDIDATA_GetTimerHardwareGateStatus
ADDIDATA_ENABLE_DISABLE_TIMER_HARDWARE_TRIGGER	214	b_ADDIDATA_EnableDisableTimerHardwareTrigger
ADDIDATA_GET_TIMER_HARDWARE_TRIGGER_STATUS	215	b_ADDIDATA_GetTimerHardwareTriggerStatus
ADDIDATA_ENABLE_DISABLE_TIMER_HARDWARE_OUTPUT	216	b_ADDIDATA_EnableDisableTimerHardwareOutput
ADDIDATA_GET_TIMER_HARDWARE_OUTPUT_STATUS	217	b_ADDIDATA_GetTimerHardwareOutputStatus
ADDIDATA_RELEASE_TIMER	218	b_ADDIDATA_ReleaseTimer
ADDIDATA_TEST_TIMER_ASYNCHRONOUS_FIFO_FULL	219	b_ADDIDATA_TestTimerAsynchronousFIFOFull
ADDIDATA_GET_TIMER_INFORMATION_EX	220	b_ADDIDATA_GetTimerInformationEx

## Constant

## Return value

## Function

## Watchdog

ADDIDATA_GET_NUMBER_OF_WATCHDOGS	300	b_ADDIDATA_GetNumberOfWatchdogs
ADDIDATA_GET_WATCHDOG_INFORMATION	301	b_ADDIDATA_GetWatchdogInformation
ADDIDATA_INIT_WATCHDOG	302	b_ADDIDATA_InitWatchdog
ADDIDATA_ENABLE_DISABLE_WATCHDOG_INTERRUPT	303	b_ADDIDATA_EnableDisableWatchdogInterrupt
ADDIDATA_START_WATCHDOG	304	b_ADDIDATA_StartWatchdog
ADDIDATA_START_ALL_WATCHDOGS	305	b_ADDIDATA_StartAllWatchdogs
ADDIDATA_TRIGGER_WATCHDOG	306	b_ADDIDATA_TriggerWatchdog
ADDIDATA_TRIGGER_ALL_WATCHDOGS	307	b_ADDIDATA_TriggerAllWatchdogs
ADDIDATA_STOP_WATCHDOG	308	b_ADDIDATA_StopWatchdog
ADDIDATA_STOP_ALL_WATCHDOGS	309	b_ADDIDATA_StopAllWatchdogs
ADDIDATA_READ_WATCHDOG_STATUS	310	b_ADDIDATA_ReadWatchdogStatus
ADDIDATA_ENABLE_DISABLE_WATCHDOG_HARDWARE_GATE	311	b_ADDIDATA_EnableDisableWatchdogHardwareGate
ADDIDATA_GET_WATCHDOG_HARDWARE_GATE_STATUS	312	b_ADDIDATA_GetWatchdogHardwareGateStatus
ADDIDATA_ENABLE_DISABLE_WATCHDOG_HARDWARE_TRIGGER	313	b_ADDIDATA_EnableDisableWatchdogHardwareTrigger
ADDIDATA_GET_WATCHDOG_HARDWARE_TRIGGER_STATUS	314	b_ADDIDATA_GetWatchdogHardwareTriggerStatus
ADDIDATA_GET_WARNING_DELAY_INFORMATION	315	b_ADDIDATA_GetWarningDelayInformation
ADDIDATA_INIT_WARNING_DELAY	316	b_ADDIDATA_InitWarningDelay
ADDIDATA_ENABLE_DISABLE_WATCHDOG_WARNING_RELAY	317	b_ADDIDATA_EnableDisableWatchdogWarningRelay
ADDIDATA_ENABLE_DISABLE_WATCHDOG_RESET_RELAY	318	b_ADDIDATA_EnableDisableWatchdogResetRelay
ADDIDATA_RELEASE_WATCHDOG	319	b_ADDIDATA_ReleaseWatchdog
ADDIDATA_GET_WATCHDOG_INFORMATION_EX	320	b_ADDIDATA_GetWatchdogInformationEx
ADDIDATA_SET_WATCHDOG_RESET_RELAY_MODE	321	b_ADDIDATA_SetWatchdogResetRelayMode
ADDIDATA_TEST_WATCHDOG_ASYNCHRONOUS_FIFO_FULL	322	b_ADDIDATA_TestWatchdogAsynchronousFIFOFull
ADDIDATA_ENABLE_DISABLE_WATCHDOG_HARDWARE_OUTPUT	323	b_ADDIDATA_EnableDisableWatchdogHardwareOutput
ADDIDATA_GET_WATCHDOG_HARDWARE_OUTPUT_STATUS	324	b_ADDIDATA_GetWatchdogHardwareOutputStatus

## Temperature

ADDIDATA_INIT_TEMPERATURE_CHANNEL	400	b_ADDIDATA_InitTemperatureChannel
ADDIDATA_READ_I_TEMPERATURE_CHANNEL	401	b_ADDIDATA_ReadITemperatureChannel
ADDIDATA_GET_NUMBER_OF_TEMPERATURE_CHANNELS	402	b_ADDIDATA_GetNumberOfTemperatureChannels
ADDIDATA_CONVERT_DIGITAL_TO_REAL_TEMPERATURE_VALUE	403	b_ADDIDATA_ConvertDigitalToRealTemperatureValue
ADDIDATA_GET_TEMPERATURE_CHANNEL_INFORMATION	404	b_ADDIDATA_GetTemperatureChannelInformation
ADDIDATA_INIT_TEMPERATURE_WARNING	405	b_ADDIDATA_InitTemperatureWarning
ADDIDATA_ENABLE_DISABLE_TEMPERATURE_WARNING_CHANNEL	406	b_ADDIDATA_EnableDisableTemperatureWarningChannel
ADDIDATA_START_ALL_TEMPERATURE_WARNINGS	407	b_ADDIDATA_StartAllTemperatureWarnings
ADDIDATA_STOP_ALL_TEMPERATURE_WARNINGS	408	b_ADDIDATA_StopAllTemperatureWarnings
ADDIDATA_RELEASE_TEMPERATURE_WARNING	409	b_ADDIDATA_ReleaseTemperatureWarning
ADDIDATA_RELEASE_TEMPERATURE_CHANNEL	410	b_ADDIDATA_ReleaseTemperatureChannel
ADDIDATA_READ_MORE_TEMPERATURE_CHANNELS	411	b_ADDIDATA_ReadMoreTemperatureChannels
ADDIDATA_INIT_TEMPERATURE_CHANNEL_SCAN	412	b_ADDIDATA_InitTemperatureChannelSCAN
ADDIDATA_START_TEMPERATURE_CHANNEL_SCAN	413	b_ADDIDATA_StartTemperatureChannelSCAN
ADDIDATA_GET_TEMPERATURE_CHANNEL_SCAN_STATUS	414	b_ADDIDATA_GetTemperatureChannelSCANStatus
ADDIDATA_CONVERT_DIGITAL_TO_REAL_TEMPERATURE_VALUE_SCAN	415	b_ADDIDATA_ConvertDigitalToRealTemperatureValueSCAN
ADDIDATA_STOP_TEMPERATURE_CHANNEL_SCAN	416	b_ADDIDATA_StopTemperatureChannelSCAN
ADDIDATA_CLOSE_TEMPERATURE_CHANNEL_SCAN	417	b_ADDIDATA_CloseTemperatureChannelSCAN
ADDIDATA_GET_NUMBER_OF_TEMPERATURE_MODULES	419	b_ADDIDATA_GetNumberOfTemperatureModules
ADDIDATA_GET_NUMBER_OF_TEMPERATURE_CHANNELS_FOR_THE_MODULE	420	b_ADDIDATA_GetNumberOfTemperatureChannelsForTheModule
ADDIDATA_CONVERT_DIGITAL_TO_REAL_TEMPERATURE_WITH_CORRECTION_PARAMETERS	421	b_ADDIDATA_ConvertDigitalToRealTemperatureWithCorrectionParameters
ADDIDATA_CONVERT_MORE_DIGITAL_TO_REAL_TEMPERATURE_VALUES	422	b_ADDIDATA_ConvertMoreDigitalToRealTemperatureValues
ADDIDATA_TEST_TEMPERATURE_CHANNEL_SHORT_CIRCUIT	423	b_ADDIDATA_TestTemperatureChannelShortCircuit
ADDIDATA_TEST_TEMPERATURE_CHANNEL_CONNECTION	424	b_ADDIDATA_TestTemperatureChannelConnection
ADDIDATA_TEST_TEMPERATURE_ASYNCHRONOUS_FIFO_FULL	425	b_ADDIDATA_TestTemperatureAsynchronousFIFOFull

## Constant

## Return value

## Function

## Digital inputs

ADDIDATA_GET_NUMBER_OF_DIGITAL_INPUTS	500	b_ADDIDATA_GetNumberOfDigitalInputs
ADDIDATA_GET_DIGITAL_INPUT_INFORMATION	501	b_ADDIDATA_GetDigitalInputInformation
ADDIDATA_READ_1_DIGITAL_INPUT	502	b_ADDIDATA_Read1DigitalInput
ADDIDATA_READ_2_DIGITAL_INPUTS	503	b_ADDIDATA_Read2DigitalInputs
ADDIDATA_READ_4_DIGITAL_INPUTS	504	b_ADDIDATA_Read4DigitalInputs
ADDIDATA_READ_8_DIGITAL_INPUTS	505	b_ADDIDATA_Read8DigitalInputs
ADDIDATA_READ_16_DIGITAL_INPUTS	506	b_ADDIDATA_Read16DigitalInputs
ADDIDATA_READ_32_DIGITAL_INPUTS	507	b_ADDIDATA_Read32DigitalInputs
ADDIDATA_INIT_DIGITAL_INPUT_INTERRUPT	508	b_ADDIDATA_InitDigitalInputInterrupt
ADDIDATA_ENABLE_DISABLE_DIGITAL_INPUT_INTERRUPT	509	b_ADDIDATA_EnableDisableDigitalInputInterrupt
ADDIDATA_RELEASE_DIGITAL_INPUT_INTERRUPT	510	b_ADDIDATA_ReleaseDigitalInputInterrupt
ADDIDATA_GET_DIGITAL_INPUT_INFORMATION_EX	511	b_ADDIDATA_GetDigitalInputInformationEx
ADDIDATA_INIT_1_DIGITAL_INPUT_LEVEL	512	b_ADDIDATA_Init1DigitalInputLevel
ADDIDATA_READ_1_DIGITAL_INPUT_STATUS	513	b_ADDIDATA_Read1DigitalInputStatus
ADDIDATA_READ_MORE_DIGITAL_INPUT_STATUS	514	b_ADDIDATA_ReadMoreDigitalInputStatus
ADDIDATA_READ_1_DIGITAL_INPUT_VALUE	515	b_ADDIDATA_Read1DigitalInputValue
ADDIDATA_READ_MORE_DIGITAL_INPUT_VALUE	516	b_ADDIDATA_ReadMoreDigitalInputValue
ADDIDATA_CONVERT_1_DIGITAL_INPUT_VALUE	517	b_ADDIDATA_Convert1DigitalInputValueInAnalogValue
ADDIDATA_GET_DIGITAL_INPUT_MODULE_FILTER_INFORMATION	518	b_ADDIDATA_GetDigitalInputModuleFilterInformation
ADDIDATA_INIT_DIGITAL_INPUT_MODULE_FILTER	519	b_ADDIDATA_InitDigitalInputModuleFilter
ADDIDATA_ENABLE_DISABLE_DIGITAL_INPUT_MODULE_FILTER	520	b_ADDIDATA_EnableDisableDigitalInputModuleFilter
ADDIDATA_SET_DIGITAL_INPUT_MODULE_LEVEL_SELECTION	521	b_ADDIDATA_SetDigitalInputModuleLevelSelection
ADDIDATA_SAVE_DIGITAL_INPUT_MODULE_LEVEL	522	b_ADDIDATA_SaveDigitalInputModuleLevel
ADDIDATA_TEST_DIGITAL_INPUT_ASYNCHRONOUS_FIFO_FULL	523	b_ADDIDATA_TestDigitalInputAsynchronousFIFOFull

## Digital outputs

ADDIDATA_GET_NUMBER_OF_DIGITAL_OUTPUTS	600	b_ADDIDATA_GetNumberOfDigitalOutputs
ADDIDATA_GET_DIGITAL_OUTPUT_INFORMATION	601	b_ADDIDATA_GetDigitalOutputInformation
ADDIDATA_SET_DIGITAL_OUTPUT_MEMORY_ON	602	b_ADDIDATA_SetDigitalOutputMemoryOn
ADDIDATA_SET_DIGITAL_OUTPUT_MEMORY_OFF	603	b_ADDIDATA_SetDigitalOutputMemoryOff
ADDIDATA_SET_1_DIGITAL_OUTPUT_ON	604	b_ADDIDATA_Set1DigitalOutputOn
ADDIDATA_SET_2_DIGITAL_OUTPUTS_ON	605	b_ADDIDATA_Set2DigitalOutputsOn
ADDIDATA_SET_4_DIGITAL_OUTPUTS_ON	606	b_ADDIDATA_Set4DigitalOutputsOn
ADDIDATA_SET_8_DIGITAL_OUTPUTS_ON	607	b_ADDIDATA_Set8DigitalOutputsOn
ADDIDATA_SET_16_DIGITAL_OUTPUTS_ON	608	b_ADDIDATA_Set16DigitalOutputsOn
ADDIDATA_SET_32_DIGITAL_OUTPUTS_ON	609	b_ADDIDATA_Set32DigitalOutputsOn
ADDIDATA_SET_1_DIGITAL_OUTPUT_OFF	610	b_ADDIDATA_Set1DigitalOutputOff
ADDIDATA_SET_2_DIGITAL_OUTPUTS_OFF	611	b_ADDIDATA_Set2DigitalOutputsOff
ADDIDATA_SET_4_DIGITAL_OUTPUTS_OFF	612	b_ADDIDATA_Set4DigitalOutputsOff
ADDIDATA_SET_8_DIGITAL_OUTPUTS_OFF	613	b_ADDIDATA_Set8DigitalOutputsOff
ADDIDATA_SET_16_DIGITAL_OUTPUTS_OFF	614	b_ADDIDATA_Set16DigitalOutputsOff
ADDIDATA_SET_32_DIGITAL_OUTPUTS_OFF	615	b_ADDIDATA_Set32DigitalOutputsOff
ADDIDATA_GET_1_DIGITAL_OUTPUT_STATUS	616	b_ADDIDATA_Get1DigitalOutputStatus
ADDIDATA_GET_2_DIGITAL_OUTPUT_STATUS	617	b_ADDIDATA_Get2DigitalOutputStatus
ADDIDATA_GET_4_DIGITAL_OUTPUT_STATUS	618	b_ADDIDATA_Get4DigitalOutputStatus
ADDIDATA_GET_8_DIGITAL_OUTPUT_STATUS	619	b_ADDIDATA_Get8DigitalOutputStatus
ADDIDATA_GET_16_DIGITAL_OUTPUT_STATUS	620	b_ADDIDATA_Get16DigitalOutputStatus
ADDIDATA_GET_32_DIGITAL_OUTPUT_STATUS	621	b_ADDIDATA_Get32DigitalOutputStatus
ADDIDATA_INIT_DIGITAL_OUTPUT_INTERRUPT	623	b_ADDIDATA_InitDigitalOutputInterrupt
ADDIDATA_ENABLE_DISABLE_DIGITAL_OUTPUT_INTERRUPT	624	b_ADDIDATA_EnableDisableDigitalOutputInterrupt
ADDIDATA_RELEASE_DIGITAL_OUTPUT_INTERRUPT	625	b_ADDIDATA_ReleaseDigitalOutputInterrupt
ADDIDATA_TEST_DIGITAL_OUTPUT_ASYNCHRONOUS_FIFO_FULL	626	b_ADDIDATA_TestDigitalOutputAsynchronousFIFOFull



## Constant

## Return value

## Function

## Analog outputs

ADDIDATA_GET_NUMBER_OF_ANALOG_OUTPUTS	700	b_ADDIDATA_GetNumberOfAnalogOutputs
ADDIDATA_GET_ANALOG_OUTPUT_INFORMATION	701	b_ADDIDATA_GetAnalogOutputInformation
ADDIDATA_INIT_1_ANALOG_OUTPUT	702	b_ADDIDATA_Init1AnalogOutput
ADDIDATA_INIT_MORE_ANALOG_OUTPUTS	703	b_ADDIDATA_InitMoreAnalogOutputs
ADDIDATA_WRITE_1_ANALOG_OUTPUT	704	b_ADDIDATA_Write1AnalogOutput
ADDIDATA_WRITE_MORE_ANALOG_OUTPUTS	705	b_ADDIDATA_WriteMoreAnalogOutputs
ADDIDATA_RELEASE_1_ANALOG_OUTPUT	706	b_ADDIDATA_Release1AnalogOutput
ADDIDATA_RELEASE_MORE_ANALOG_OUTPUTS	707	b_ADDIDATA_ReleaseMoreAnalogOutputs
ADDIDATA_ENABLE_DISABLE_1_ANALOG_OUTPUT_SYNC	708	b_ADDIDATA_EnableDisable1AnalogOutputSync
ADDIDATA_ENABLE_DISABLE_MORE_ANALOG_OUTPUTS_SYNC	709	b_ADDIDATA_EnableDisableMoreAnalogOutputsSync
ADDIDATA_TRIGGER_ANALOG_OUTPUT	710	b_ADDIDATA_TriggerAnalogOutput

## Analog inputs

ADDIDATA_GET_NUMBER_OF_ANALOG_INPUTS	800	b_ADDIDATA_GetNumberOfAnalogInputs
ADDIDATA_GET_NUMBER_OF_ANALOG_INPUT_MODULES	801	b_ADDIDATA_GetNumberOfAnalogInputModules
ADDIDATA_GET_NUMBER_OF_ANALOG_INPUTS_FOR_THE_MODULE	802	b_ADDIDATA_GetNumberOfAnalogInputsForTheModule
ADDIDATA_GET_ANALOG_INPUT_INFORMATION	803	b_ADDIDATA_GetAnalogInputInformation
ADDIDATA_INIT_ANALOG_INPUT	804	b_ADDIDATA_InitAnalogInput
ADDIDATA_TEST_ANALOG_INPUT_SHORT_CIRCUIT	805	b_ADDIDATA_TestAnalogInputShortCircuit
ADDIDATA_TEST_ANALOG_INPUT_CONNECTION	806	b_ADDIDATA_TestAnalogInputConnection
ADDIDATA_READ_1_ANALOG_INPUT	807	b_ADDIDATA_Read1AnalogInput
ADDIDATA_READ_MORE_ANALOG_INPUTS	808	b_ADDIDATA_ReadMoreAnalogInputs
ADDIDATA_CONVERT_DIGITAL_TO_REAL_ANALOG_VALUE	809	b_ADDIDATA_ConvertDigitalToRealAnalogValue
ADDIDATA_CONVERT_MORE_DIGITAL_TO_REAL_ANALOG_VALUES	810	b_ADDIDATA_ConvertMoreDigitalToRealAnalogValues
ADDIDATA_INIT_ANALOG_INPUT_SCAN	811	b_ADDIDATA_InitAnalogInputSCAN
ADDIDATA_START_ANALOG_INPUT_SCAN	812	b_ADDIDATA_StartAnalogInputSCAN
ADDIDATA_GET_ANALOG_INPUT_SCAN_STATUS	813	b_ADDIDATA_GetAnalogInputSCANStatus
ADDIDATA_CONVERT_DIGITAL_TO_REAL_ANALOG_VALUE_SCAN	814	b_ADDIDATA_ConvertDigitalToRealAnalogValueSCAN
ADDIDATA_STOP_ANALOG_INPUT_SCAN	815	b_ADDIDATA_StopAnalogInputSCAN
ADDIDATA_CLOSE_ANALOG_INPUT_SCAN	816	b_ADDIDATA_CloseAnalogInputSCAN
ADDIDATA_RELEASE_ANALOG_INPUT	817	b_ADDIDATA_ReleaseAnalogInput
ADDIDATA_TEST_ANALOG_INPUT_ASYNCHRONOUS_FIFO_FULL	818	b_ADDIDATA_TestAnalogInputAsynchronousFIFOFull
ADDIDATA_GET_ANALOG_INPUT_MODULE_NUMBER	819	b_ADDIDATA_GetAnalogInputModuleNumber
ADDIDATA_GET_ANALOG_INPUT_MODULE_AUTO_REFRESH_ACQUISITION_INFORMATION	820	b_ADDIDATA_GetAnalogInputModuleAutoRefreshInformation
ADDIDATA_GET_ANALOG_INPUT_MODULE_SEQUENCE_ACQUISITION_INFORMATION	821	b_ADDIDATA_GetAnalogInputModuleSequenceInformation
ADDIDATA_GET_ANALOG_INPUT_AUTO_REFRESH_CHANNEL_POINTER	822	b_ADDIDATA_GetAnalogInputAutoRefreshChannelPointer
ADDIDATA_GET_ANALOG_INPUT_AUTO_REFRESH_MODULE_POINTER	823	b_ADDIDATA_GetAnalogInputAutoRefreshModulePointer
ADDIDATA_START_ANALOG_INPUT_AUTO_REFRESH	824	b_ADDIDATA_StartAnalogInputAutoRefresh
ADDIDATA_STOP_ANALOG_INPUT_AUTO_REFRESH	825	b_ADDIDATA_StopAnalogInputAutoRefresh
ADDIDATA_READ_1_ANALOG_INPUT_AUTO_REFRESH_VALUE	826	b_ADDIDATA_Read1AnalogInputAutoRefreshValue
ADDIDATA_INIT_ANALOG_INPUT_SEQUENCE_ACQUISITION	827	b_ADDIDATA_InitAnalogInputSequenceAcquisition
ADDIDATA_START_ANALOG_INPUT_SEQUENCE_ACQUISITION	828	b_ADDIDATA_StartAnalogInputSequenceAcquisition
ADDIDATA_PAUSE_ANALOG_INPUT_SEQUENCE_ACQUISITION	829	b_ADDIDATA_PauseAnalogInputSequenceAcquisition
ADDIDATA_STOP_ANALOG_INPUT_SEQUENCE_ACQUISITION	830	b_ADDIDATA_StopAnalogInputSequenceAcquisition
ADDIDATA_RELEASE_ANALOG_INPUT_SEQUENCE_ACQUISITION	831	b_ADDIDATA_ReleaseAnalogInputSequenceAcquisition
ADDIDATA_CONVERT_DIGITAL_TO_REAL_ANALOG_VALUE_SEQUENCE	832	b_ADDIDATA_ConvertDigitalToRealAnalogValueSequence
ADDIDATA_GET_ANALOG_INPUT_SEQUENCE_ACQUISITION_HANDLE_STATUS	833	b_ADDIDATA_GetAnalogInputSequenceAcquisitionHandleStatus
ADDIDATA_GET_ANALOG_INPUT_HARDWARE_TRIGGER_INFORMATION	834	b_ADDIDATA_GetAnalogInputHardwareTriggerInformation
ADDIDATA_ENABLE_DISABLE_ANALOG_INPUT_HARDWARE_TRIGGER	835	b_ADDIDATA_EnableDisableAnalogInputHardwareTrigger
ADDIDATA_GET_ANALOG_INPUT_HARDWARE_TRIGGER_STATUS	836	b_ADDIDATA_GetAnalogInputHardwareTriggerStatus
ADDIDATA_ENABLE_DISABLE_ANALOG_INPUT_SOFTWARE_TRIGGER	837	b_ADDIDATA_EnableDisableAnalogInputSoftwareTrigger
ADDIDATA_ANALOG_INPUT_SOFTWARE_TRIGGER	838	b_ADDIDATA_AnalogInputSoftwareTrigger
ADDIDATA_GET_ANALOG_INPUT_SOFTWARE_TRIGGER_STATUS	839	b_ADDIDATA_GetAnalogInputSoftwareTriggerStatus
ADDIDATA_GET_ANALOG_INPUT_MODULE_GENERAL_INFORMATION	840	b_ADDIDATA_GetAnalogInputAutoRefreshModuleCounterPointer

ADDIDATA_GET_ANALOG_INPUT_MODULE_SINGLE_ACQUISITION_INFORMATION	841	b_ADDIDATA_GetAnalogInputModuleGeneralInformation
ADDIDATA_GET_ANALOG_INPUT_AUTO_REFRESH_MODULE_COUNTER_POINTER	842	b_ADDIDATA_GetAnalogInputModuleSingleAcquisitionInformation
ADDIDATA_GET_ANALOG_INPUT_MODULE_SCAN_ACQUISITION_INFORMATION	843	b_ADDIDATA_GetAnalogInputModuleSCANInformation
ADDIDATA_INIT_ANALOG_INPUT_SCAN_ACQUISITION	844	b_ADDIDATA_InitAnalogInputSCANAcquisition
ADDIDATA_GET_ANALOG_INPUT_SOFTWARE_TRIGGER_INFORMATION	845	b_ADDIDATA_GetAnalogInputSoftwareTriggerInformation

## Resistance

ADDIDATA_GET_NUMBER_OF_RESISTANCE_CHANNELS	900	b_ADDIDATA_GetNumberOfResistanceChannels
ADDIDATA_GET_NUMBER_OF_RESISTANCE_CHANNEL_MODULES	901	b_ADDIDATA_GetNumberOfResistanceChannelModules
ADDIDATA_GET_NUMBER_OF_RESISTANCE_CHANNELS_FOR_THE_MODULE	902	b_ADDIDATA_GetNumberOfResistanceChannelsForTheModule
ADDIDATA_GET_RESISTANCE_CHANNELS_INFORMATION	903	b_ADDIDATA_GetResistanceChannelInformation
ADDIDATA_INIT_RESISTANCE_CHANNEL	904	b_ADDIDATA_InitResistanceChannel
ADDIDATA_TEST_RESISTANCE_CHANNEL_SHORT_CIRCUIT	905	b_ADDIDATA_TestResistanceChannelShortCircuit
ADDIDATA_TEST_RESISTANCE_CHANNEL_CONNECTION	906	b_ADDIDATA_TestResistanceChannelConnection
ADDIDATA_READ_1_RESISTANCE_CHANNEL	907	b_ADDIDATA_Read1ResistanceChannel
ADDIDATA_READ_MORE_RESISTANCE_CHANNELS	908	b_ADDIDATA_ReadMoreResistanceChannels
ADDIDATA_CONVERT_DIGITAL_TO_REAL_RESISTANCE_VALUE	909	b_ADDIDATA_ConvertDigitalToRealResistanceValue
ADDIDATA_CONVERT_MORE_DIGITAL_TO_REAL_RESISTANCE_VALUES	910	b_ADDIDATA_ConvertMoreDigitalToRealResistanceValues
ADDIDATA_INIT_RESISTANCE_CHANNEL_SCAN	911	b_ADDIDATA_InitResistanceChannelSCAN
ADDIDATA_START_RESISTANCE_CHANNEL_SCAN	912	b_ADDIDATA_StartResistanceChannelSCAN
ADDIDATA_GET_RESISTANCE_CHANNEL_SCAN_STATUS	913	b_ADDIDATA_GetResistanceChannelSCANStatus
ADDIDATA_CONVERT_DIGITAL_TO_REAL_RESISTANCE_VALUE_SCAN	914	b_ADDIDATA_ConvertDigitalToRealResistanceValueSCAN
ADDIDATA_STOP_RESISTANCE_CHANNEL_SCAN	915	b_ADDIDATA_StopResistanceChannelSCAN
ADDIDATA_CLOSE_RESISTANCE_CHANNEL_SCAN	916	b_ADDIDATA_CloseResistanceChannelSCAN
ADDIDATA_RELEASE_RESISTANCE_CHANNEL	917	b_ADDIDATA_ReleaseResistanceChannel
ADDIDATA_TEST_RESISTANCE_CHANNEL_ASYNCHRONOUS_FIFO_FULL	918	b_ADDIDATA_TestResistanceChannelAsynchronousFIFOFull

## Constant

## Return value

## Function

## Counter

ADDIDATA_GET_NUMBER_OF_COUNTERS	1000	b_ADDIDATA_GetNumberOfCounters
ADDIDATA_GET_COUNTER_INFORMATION	1001	b_ADDIDATA_GetCounterInformation
ADDIDATA_INIT_COUNTER	1002	b_ADDIDATA_InitCounter
ADDIDATA_COUNTER_DIRECTION	1003	b_ADDIDATA_SetCounterDirection
ADDIDATA_ENABLE_DISABLE_COUNTER_INTERRUPT	1004	b_ADDIDATA_EnableDisableCounterInterrupt
ADDIDATA_START_COUNTER	1005	b_ADDIDATA_StartCounter
ADDIDATA_START_ALL_COUNTERS	1006	b_ADDIDATA_StartAllCounters
ADDIDATA_CLEAR_COUNTER	1007	b_ADDIDATA_ClearCounter
ADDIDATA_TRIGGER_COUNTER	1008	b_ADDIDATA_TriggerCounter
ADDIDATA_TRIGGER_ALL_COUNTERS	1009	b_ADDIDATA_TriggerAllCounters
ADDIDATA_STOP_COUNTER	1010	b_ADDIDATA_StopCounter
ADDIDATA_STOP_ALL_COUNTERS	1011	b_ADDIDATA_StopAllCounters
ADDIDATA_READ_COUNTER_VALUE	1012	b_ADDIDATA_ReadCounterValue
ADDIDATA_READ_COUNTER_STATUS	1013	b_ADDIDATA_ReadCounterStatus
ADDIDATA_ENABLE_DISABLE_COUNTER_HARDWARE_GATE	1014	b_ADDIDATA_EnableDisableCounterHardwareGate
ADDIDATA_GET_COUNTER_HARDWARE_GATE_STATUS	1015	b_ADDIDATA_GetCounterHardwareGateStatus
ADDIDATA_ENABLE_DISABLE_COUNTER_HARDWARE_TRIGGER	1016	b_ADDIDATA_EnableDisableCounterHardwareTrigger
ADDIDATA_GET_COUNTER_HARDWARE_TRIGGER_STATUS	1017	b_ADDIDATA_GetCounterHardwareTriggerStatus
ADDIDATA_ENABLE_DISABLE_COUNTER_HARDWARE_OUTPUT	1018	b_ADDIDATA_EnableDisableCounterHardwareOutput
ADDIDATA_GET_COUNTER_HARDWARE_OUTPUT_STATUS	1019	b_ADDIDATA_GetCounterHardwareOutputStatus
ADDIDATA_RELEASE_COUNTER	1020	b_ADDIDATA_ReleaseCounter
ADDIDATA_TEST_COUNTER_ASYNCHRONOUS_FIFO_FULL	1021	b_ADDIDATA_TestCounterAsynchronousFIFOFull
ADDIDATA_GET_COUNTER_INFORMATION_EX	1022	b_ADDIDATA_GetCounterInformationEx

## Pressure

ADDIDATA_GET_NUMBER_OF_PRESSURE_CHANNELS	1100	b_ADDIDATA_GetNumberOfPressureChannels
ADDIDATA_GET_NUMBER_OF_PRESSURE_MODULES	1101	b_ADDIDATA_GetNumberOfPressureModules
ADDIDATA_GET_NUMBER_OF_PRESSURE_CHANNELS_FOR_THE_MODULE	1102	b_ADDIDATA_GetNumberOfPressureChannelsForTheModule
ADDIDATA_GET_PRESSURE_CHANNELS_INFORMATION	1103	b_ADDIDATA_GetPressureChannelInformation
ADDIDATA_INIT_PRESSURE_CHANNEL	1104	b_ADDIDATA_InitPressureChannel
ADDIDATA_READ_1_PRESSURE_CHANNEL	1105	b_ADDIDATA_Read1PressureChannel
ADDIDATA_READ_MORE_PRESSURE_CHANNELS	1106	b_ADDIDATA_ReadMorePressureChannels
ADDIDATA_CONVERT_DIGITAL_TO_REAL_PRESSURE_VALUE	1107	b_ADDIDATA_ConvertDigitalToRealPressureValue
ADDIDATA_CONVERT_MORE_DIGITAL_TO_REAL_PRESSURE_VALUES	1108	b_ADDIDATA_ConvertMoreDigitalToRealPressureValues
ADDIDATA_INIT_PRESSURE_CHANNEL_SCAN	1109	b_ADDIDATA_InitPressureSCAN
ADDIDATA_START_PRESSURE_CHANNEL_SCAN	1110	b_ADDIDATA_StartPressureSCAN
ADDIDATA_GET_PRESSURE_CHANNEL_SCAN_STATUS	1111	b_ADDIDATA_GetPressureSCANStatus
ADDIDATA_CONVERT_DIGITAL_TO_REAL_PRESSURE_VALUE_SCAN	1112	b_ADDIDATA_ConvertDigitalToRealPressureValueSCAN
ADDIDATA_STOP_PRESSURE_CHANNEL_SCAN	1113	b_ADDIDATA_StopPressureChannelSCAN
ADDIDATA_CLOSE_PRESSURE_CHANNEL_SCAN	1114	b_ADDIDATA_ClosePressureChannelSCAN
ADDIDATA_RELEASE_PRESSURE_CHANNEL	1115	b_ADDIDATA_ReleasePressureChannel
ADDIDATA_TEST_PRESSURE_CHANNEL_ASYNCHRONOUS_FIFO_FULL	1118	b_ADDIDATA_TestPressureChannelAsynchronousFIFOFull

## Transducer

ADDIDATA_GET_NUMBER_OF_TRANSDUCERS_CHANNELS	1200	b_ADDIDATA_GetNumberOfTransducerChannels
ADDIDATA_GET_NUMBER_OF_TRANSDUCER_MODULES	1201	b_ADDIDATA_GetNumberOfTransducerModules
ADDIDATA_GET_NUMBER_OF_TRANSDUCER_CHANNELS_FOR_THE_MODULE	1202	b_ADDIDATA_GetNumberOfTransducerChannelsForTheModule
ADDIDATA_GET_TRANSDUCER_CHANNEL_MODULE_NUMBER	1203	b_ADDIDATA_GetTransducerChannelModuleNumber
ADDIDATA_GET_TRANSDUCER_MODULE_GENERAL_INFORMATION	1204	b_ADDIDATA_GetTransducerModuleGeneralInformation
ADDIDATA_GET_TRANSDUCER_MODULE_SINGLE_ACQUISITION_INFORMATION	1205	b_ADDIDATA_GetTransducerModuleSingleAcquisitionInformation
ADDIDATA_GET_TRANSDUCER_MODULE_AUTO_REFRESH_ACQUISITION_INFORMATION	1206	b_ADDIDATA_GetTransducerModuleAutoRefreshInformation
ADDIDATA_GET_TRANSDUCER_MODULE_SEQUENCE_ACQUISITION_INFORMATION	1207	b_ADDIDATA_GetTransducerModuleSequenceInformation
ADDIDATA_INIT_TRANSDUCER_CHANNEL	1208	b_ADDIDATA_InitTransducerChannel
ADDIDATA_RELEASE_TRANSDUCER_CHANNEL	1209	b_ADDIDATA_ReleaseTransducerChannel
ADDIDATA_READ_1_TRANSDUCER_CHANNEL	1210	b_ADDIDATA_Read1TransducerChannel
ADDIDATA_CONVERT_DIGITAL_TO_REAL_METRIC_VALUE	1211	b_ADDIDATA_ConvertDigitalToRealMetricValue
ADDIDATA_READ_MORE_TRANSDUCER_CHANNELS	1212	b_ADDIDATA_ReadMoreTransducerChannels
ADDIDATA_CONVERT_MORE_DIGITAL_TO_REAL_METRIC_VALUE	1213	b_ADDIDATA_ConvertMoreDigitalToRealMetricValue
ADDIDATA_GET_TRANSDUCER_HARDWARE_TRIGGER_INFORMATION	1214	b_ADDIDATA_GetTransducerHardwareTriggerInformation
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_HARDWARE_TRIGGER	1215	b_ADDIDATA_EnableDisableTransducerHardwareTrigger
ADDIDATA_GET_TRANSDUCER_HARDWARE_TRIGGER_STATUS	1216	b_ADDIDATA_GetTransducerHardwareTriggerStatus
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_SOFTWARE_TRIGGER	1217	b_ADDIDATA_EnableDisableTransducerSoftwareTrigger
ADDIDATA_TRANSDUCER_SOFTWARE_TRIGGER	1218	b_ADDIDATA_TransducerSoftwareTrigger
ADDIDATA_GET_TRANSDUCER_SOFTWARE_TRIGGER_STATUS	1219	b_ADDIDATA_GetTransducerSoftwareTriggerStatus
ADDIDATA_GET_TRANSDUCER_HARDWARE_GATE_INFORMATION	1220	b_ADDIDATA_GetTransducerHardwareGateInformation
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_HARDWARE_GATE	1221	b_ADDIDATA_EnableDisableTransducerHardwareGate
ADDIDATA_GET_TRANSDUCER_HARDWARE_GATE_STATUS	1222	b_ADDIDATA_GetTransducerHardwareGateStatus
ADDIDATA_TEST_TRANSDUCER_CHANNEL_SECONDARY_CONNECTION	1223	b_ADDIDATA_TestTransducerChannelSecondaryConnection
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_MODULE_PRIMARY_CONNECTION_TEST	1224	b_ADDIDATA_EnableDisableTransducerModulePrimaryConnectionTest
ADDIDATA_TEST_TRANSDUCER_MODULE_PRIMARY_CONNECTION	1225	b_ADDIDATA_TestTransducerModulePrimaryConnection
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_MODULE_PRIMARY_SHORT_CIRCUIT_INTERRUPT	1226	b_ADDIDATA_EnableDisableTransducerModulePrimaryShortCircuitInterrupt
ADDIDATA_INIT_TRANSDUCER_SEQUENCE_ACQUISITION	1227	b_ADDIDATA_InitTransducerSequenceAcquisition
ADDIDATA_START_TRANSDUCER_SEQUENCE_ACQUISITION	1228	b_ADDIDATA_StartTransducerSequenceAcquisition
ADDIDATA_PAUSE_TRANSDUCER_SEQUENCE_ACQUISITION	1229	b_ADDIDATA_PauseTransducerSequenceAcquisition
ADDIDATA_STOP_TRANSDUCER_SEQUENCE_ACQUISITION	1230	b_ADDIDATA_StopTransducerSequenceAcquisition
ADDIDATA_RELEASE_TRANSDUCER_SEQUENCE_ACQUISITION	1231	b_ADDIDATA_ReleaseTransducerSequenceAcquisition

ADDIDATA_CONVERT_TRANSDUCER_SEQUENCE_DIGITAL_TO_REAL_METRIC_VALUE	1232	b_ADDIDATA_ConvertTransducerSequenceDigitalToRealMetricValue
ADDIDATA_GET_TRANSDUCER_SEQUENCE_ACQUISITION_HANDLE_STATUS	1233	b_ADDIDATA_GetTransducerSequenceAcquisitionHandleStatus
ADDIDATA_REARM_TRANSDUCER_MODULE_PRIMARY_SHORT_CIRCUIT_CONNECTION_TEST	1234	b_ADDIDATA_RearmTransducerModulePrimaryShortCircuitConnectionTest
ADDIDATA_GET_TRANSDUCER_MODULE_CONVERT_TIME_DIVISION_FACTOR_INFORMATION	1235	b_ADDIDATA_GetTransducerModuleConvertTimeDivisionFactorInformation
ADDIDATA_INIT_TRANSDUCER_MODULE_CONVERT_TIME_DIVISION_FACTOR	1236	b_ADDIDATA_InitTransducerModuleConvertTimeDivisionFactor
ADDIDATA_RELEASE_TRANSDUCER_MODULE_CONVERT_TIME_DIVISION_FACTOR	1237	b_ADDIDATA_ReleaseTransducerModuleConvertTimeDivisionFactor
ADDIDATA_GET_TRANSDUCER_AUTO_REFRESH_CHANNEL_POINTER	1238	b_ADDIDATA_GetTransducerAutoRefreshChannelPointer
ADDIDATA_GET_TRANSDUCER_AUTO_REFRESH_MODULE_POINTER	1239	b_ADDIDATA_GetTransducerAutoRefreshModulePointer
ADDIDATA_GET_TRANSDUCER_AUTO_REFRESH_MODULE_COUNTER_POINTER	1240	b_ADDIDATA_GetTransducerAutoRefreshModuleCounterPointer
ADDIDATA_START_TRANSDUCER_AUTO_REFRESH	1241	b_ADDIDATA_StartTransducerAutoRefresh
ADDIDATA_STOP_TRANSDUCER_AUTO_REFRESH	1242	b_ADDIDATA_StopTransducerAutoRefresh
ADDIDATA_SET_TRANSDUCER_AUTO_REFRESH_AVERAGE_VALUE	1243	b_ADDIDATA_SetTransducerAutoRefreshAverageValue
ADDIDATA_SET_TRANSDUCER_AUTO_REFRESH_AVERAGE_MODE	1244	b_ADDIDATA_SetTransducerAutoRefreshAverageMode
ADDIDATA_SET_TRANSDUCER_AUTO_REFRESH_THREAD	1245	b_ADDIDATA_TransducerAutoRefreshThread
ADDIDATA_GET_TRANSDUCER_HARDWARE_TRIGGER_INFORMATION_EX	1246	b_ADDIDATA_GetTransducerHardwareTriggerInformationEx
ADDIDATA_ENABLE_DISABLE_TRANSDUCER_HARDWARE_TRIGGER_EX	1247	b_ADDIDATA_EnableDisableTransducerHardwareTriggerEx
ADDIDATA_READ_1_TRANSDUCER_AUTO_REFRESH_VALUE	1248	b_ADDIDATA_Read1TransducerAutoRefreshValue
ADDIDATA_READ_TRANSDUCER_AUTO_REFRESH_COUNTER_VALUE	1249	b_ADDIDATA_ReadTransducerAutoRefreshCounterValue

Table 2-3: Error number

Constant	Return value	Meaning
<b>Interrupt errors</b>		
ADDIDATA_INTERRUPT_USER_SHARED_MEMORY_SIZE_ERROR	-150	Selected size for the user shared memory is wrong
ADDIDATA_INTERRUPT_SHARED_MEMORY_MODE_ERROR	-151	Selected mode for the user shared memory is wrong
ADDIDATA_INTERRUPT_USER_CALLING_MODE_ERROR	-152	The selected user calling mode is wrong
ADDIDATA_INTERRUPT_FUNCTIONALITY_ALREADY_INSTALLED	-153	Interrupt for the selected functionality is already installed
ADDIDATA_INTERRUPT_USER_SHARED_MEMORY_NOT_FOUND	-154	Cannot find the user shared memory
ADDIDATA_INTERRUPT_USER_SHARED_MEMORY_ALLOCATION_ERROR	-155	No memory available for the allocation of the user shared memory
ADDIDATA_INTERRUPT_USER_INSTALLATION_FUNCTION_ERROR	-156	Error when installing the user function
ADDIDATA_INTERRUPT_PREPARE_FUNCTIONALITY_INTERRUPT_FUNCTION_ERROR	-157	Error by preparing the interrupt function for each functionality
ADDIDATA_INTERRUPT_PREPARE_API_INTERRUPT_FUNCTION_ERROR	-158	Error by preparing the API interrupt function
ADDIDATA_INTERRUPT_INSTALL_API_INTERRUPT_FUNCTION_ERROR	-159	API interrupt installation error. Interrupt already used
ADDIDATA_INTERRUPT_NO_INDEX_AVAILABLE	-160	The selected functionality interrupt cannot be enabled
ADDIDATA_INTERRUPT_INDEX_NUMBER_ERROR	-161	The selected functionality interrupt cannot be disabled
ADDIDATA_INTERRUPT_FUNCTIONALITY_NUMBER_ERROR	-162	The selected functionality number is wrong
ADDIDATA_INTERRUPT_FUNCTIONALITY_NOT_INSTALLED	-163	User interrupt function not installed
ADDIDATA_INTERRUPT_END_FUNCTIONALITY_EXEC_ERROR	-164	Error by releasing the functionality interrupt
ADDIDATA_INTERRUPT_API_UNINSTALL_ERROR	-165	Error when uninstalling the API interrupt
ADDIDATA_INTERRUPT_END_API_EXEC_ERROR	-166	Error by releasing the API interrupt
ADDIDATA_INTERRUPT_FUNCTIONALITY_SELECTION_ERROR	-167	The selected interrupt functionality is wrong
ADDIDATA_INTERRUPT_API_THREAD_CREATION_ERROR	-168	Error in the creation of the API interrupt thread
ADDIDATA_INTERRUPT_API_EVENT_CREATION_ERROR	-169	Error in the creation of the API interrupt event
ADDIDATA_INTERRUPT_NO_INTERRUPT_INITIALISED	-170	No interrupt initialised
ADDIDATA_INTERRUPT_FLAG_ERROR	-171	Interrupt flag error
ADDIDATA_ACPI_ACTIV_AND_USE_OF_WINDOWS_NT4	-172	ACPI mode is set and has conflict with Windows NT 4.0
ADDIDATA_PNP_OS_ACPI_AND_USE_OF_WINDOWS_NT4	-173	Plug & Play option and ACPI mode is set: conflict with Windows NT 4.0

**Timer errors**

ADDIDATA_TIMER_NUMBER_ERROR	-200	The selected timer number is wrong
ADDIDATA_TIMER_RELOAD_VALUE_ERROR	-201	The timer reload value is wrong
ADDIDATA_TIMER_UNIT_ERROR	-202	The selected time unit for the timer is wrong
ADDIDATA_TIMER_MODE_ERROR	-203	The selected timer mode is wrong
ADDIDATA_TIMER_INTERRUPT_FLAG_ERROR	-204	Timer interrupt flag is wrong
ADDIDATA_TIMER_HARDWARE_GATE_LEVEL_ERROR	-205	Timer hardware gate not available
ADDIDATA_TIMER_HARDWARE_GATE_LEVEL_SELECTION_ERROR	-206	The selected hardware gate level is wrong
ADDIDATA_TIMER_HARDWARE_GATE_FLAG_ERROR	-207	The flag of the hardware gate is wrong
ADDIDATA_TIMER_HARDWARE_TRIGGER_LEVEL_ERROR	-208	Timer hardware trigger not available
ADDIDATA_TIMER_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	-209	Hardware trigger level selection error
ADDIDATA_TIMER_HARDWARE_TRIGGER_FLAG_ERROR	-210	The flag of the hardware trigger is wrong
ADDIDATA_TIMER_HARDWARE_OUTPUT_LEVEL_ERROR	-211	Timer output not available
ADDIDATA_TIMER_HARDWARE_OUTPUT_LEVEL_SELECTION_ERROR	-212	Timer output level selection error
ADDIDATA_TIMER_HARDWARE_OUTPUT_FLAG_ERROR	-213	The flag of the timer output is wrong
ADDIDATA_TIMER_ALREADY_USED	-214	Timer already used by another process or used as a watchdog

Constant

Return value

Meaning

Watchdog errors

ADDIDATA_WATCHDOG_NUMBER_ERROR	-300	Watchdog number error
ADDIDATA_WATCHDOG_DELAY_VALUE_ERROR	-301	Watchdog delay selection error
ADDIDATA_WATCHDOG_UNIT_ERROR	-302	Watchdog unit selection error
ADDIDATA_WATCHDOG_INTERRUPT_FLAG_ERROR	-303	The flag of the watchdog interrupt is wrong
ADDIDATA_WATCHDOG_HARDWARE_GATE_LEVEL_ERROR	-304	Watchdog hardware gate not available
ADDIDATA_WATCHDOG_HARDWARE_GATE_LEVEL_SELECTION_ERROR	-305	Hardware gate level selection error
ADDIDATA_WATCHDOG_HARDWARE_GATE_FLAG_ERROR	-306	the flag of the hardware gate is wrong
ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_LEVEL_ERROR	-307	Watchdog hardware trigger not available
ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	-308	Hardware trigger level selection error
ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_FLAG_ERROR	-309	The flag of the hardware trigger is wrong
ADDIDATA_WARNING_DELAY_VALUE_ERROR	-310	The selected warning delay is wrong
ADDIDATA_WARNING_DELAY_TIME_UNIT_ERROR	-311	The selected time unit for the warning delay is wrong
ADDIDATA_WATCHDOG_WARNING_RELAY_FLAG_ERROR	-312	The flag of the warning relay is wrong
ADDIDATA_WATCHDOG_RESET_RELAY_FLAG_ERROR	-313	The flag of the reset relay is wrong
ADDIDATA_WATCHDOG_ALREADY_USED	-314	Watchdog already used by another process or used as a timer
ADDIDATA_WATCHDOG_INFORMATION_STRUCTURE_INVALID_SIZE	-315	The specified size is wrong.
ADDIDATA_WATCHDOG_RESET_RELAY_MODE_SELECTION_ERROR	-316	The specified reset relay mode is wrong.

Analog measurement errors

ADDIDATA_ANALOG_MEASURE_CHANNEL_NUMBER_ERROR	-400	The channel number is wrong
ADDIDATA_ANALOG_MEASURE_CONVERSION_STARTED	-401	Conversion already started. Cannot start a new conversion
ADDIDATA_ANALOG_MEASURE_WARNING_FLAG_ERROR	-402	Warning not available
ADDIDATA_ANALOG_MEASURE_CONVERTING_TIME_ERROR	-403	The selected converting time value is wrong
ADDIDATA_ANALOG_MEASURE_CONVERTING_TIME_UNIT_ERROR	-404	The selected converting time unit is wrong
ADDIDATA_ANALOG_MEASURE_INTERRUPT_FLAG_ERROR	-405	Interrupt flag selection error
ADDIDATA_ANALOG_MEASURE_INTERRUPT_NOT_AVAILABLE	-406	Interrupt not available
ADDIDATA_ANALOG_MEASURE_WARNING_VALUE_ERROR	-407	Warning value selection error
ADDIDATA_ANALOG_MEASURE_CHANNEL_ALREADY_USED	-408	Channel already used by another process
ADDIDATA_ANALOG_MEASURE_WARNING_ALREADY_USED	-409	Warning already used by another process
ADDIDATA_ANALOG_MEASURE_GAIN_ERROR	-410	Gain selection is wrong
ADDIDATA_ANALOG_MEASURE_POLARITY_PARAMETER_ERROR	-411	Polarity parameter is wrong
ADDIDATA_ANALOG_MEASURE_OFFSET_RANGE_ERROR	-412	Offset range selection is wrong
ADDIDATA_ANALOG_MEASURE_CHANNEL_NOT_INITIALISED	-413	The channel to convert is not initialised
ADDIDATA_ANALOG_MEASURE_POLARITY_MODE_NOT_AVAILABLE	-414	Polarity mode is not available
ADDIDATA_ANALOG_MEASURE_SCAN_MODE_NOT_AVAILABLE	-415	The selected SCAN mode is not available
ADDIDATA_ANALOG_MEASURE_SCAN_MODE_PARAMETER_ERROR	-416	The SCAN mode parameter is wrong
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_MODE_NOT_AVAILABLE	-417	The external trigger mode is not available
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_NOT_AVAILABLE	-418	The external trigger is not available
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_PARAMETER_ERROR	-419	The external trigger parameter is wrong
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_MODE_NOT_AVAILABLE	-420	The external gate mode is not available
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_NOT_AVAILABLE	-421	The external gate is not available
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_PARAMETER_ERROR	-422	The external gate parameter is wrong
ADDIDATA_ANALOG_MEASURE_COMMON_GAIN_ERROR	-423	The gain must be the same for all channels
ADDIDATA_ANALOG_MEASURE_COMMON_POLARITY_ERROR	-424	The polarity must be the same for all channels
ADDIDATA_ANALOG_MEASURE_COMMON_OFFSET_RANGE_ERROR	-425	The offset range must be the same for all channels
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_NOT_AVAILABLE	-426	The SCAN delay is not available
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_VALUE_ERROR	-427	The value for the SCAN delay cannot be used
ADDIDATA_ANALOG_MEASURE_INTERRUPT_NOT_INSTALLED	-428	The interrupt function is not installed
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_INITIALISED	-429	The SCAN is not initialised
ADDIDATA_ANALOG_MEASURE_MODULE_NOT_AVAILABLE	-430	The selected module is not available
ADDIDATA_ANALOG_MEASURE_SCAN_ALREADY_STARTED	-431	The selected scan is already started
ADDIDATA_ANALOG_MEASURE_SCAN_COUNTER_VALUE_ERROR	-432	The SCAN counter value is wrong
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_STARTED	-433	The selected SCAN is not started
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_STOPPED	-434	The selected SCAN is not stopped

Constant	Return value	Meaning
<b>Analog measurement errors</b>		
ADDIDATA_ANALOG_MEASURE_SCAN_SAME_MODULE_REQUIRED	-435	All channels for the SCAN process must be from the same module
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_AVAILABLE	-436	The SCAN functionality is not available
ADDIDATA_ANALOG_MEASURE_SCAN_CHANNEL_SELECTION_ERROR	-437	Since the SCAN is determined by hardware, the channel selection is wrong. The first channel must be the first channel of the module and the last channel must be the last channel of the module.
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_MODE_NOT_AVAILABLE	-438	The selected SCAN delay mode is not available.
ADDIDATA_ANALOG_MEASURE_COUPLING_MODE_NOT_AVAILABLE	-439	The coupling mode is not available
ADDIDATA_ANALOG_MEASURE_COMMON_COUPLING_ERROR	-440	The coupling mode must be the same for all channels
ADDIDATA_ANALOG_MEASURE_COUPLING_PARAMETER_ERROR	-441	The coupling mode parameter is wrong
ADDIDATA_ANALOG_MEASURE_HARDWARE_INDEX_ERROR	-442	The module index is not available
ADDIDATA_ANALOG_MEASURE_CONVERSION_ERROR	-443	The conversion cannot be completed
ADDIDATA_ANALOG_MEASURE_INVALID_STRUCTURE	-444	The structure passed as a parameter is invalid
ADDIDATA_ANALOG_MEASURE_SIGN_TEST_PARAMETER_ERROR	-445	The sign test parameter is not available
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_FLAG_ERROR	-446	The flag of the hardware trigger is wrong
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_COUNTER_ERROR	-447	The selected counter value for the trigger is wrong
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_ALREADY_USED	-448	Hardware trigger already initialised by another process
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_NOT_INITIALISED	-449	Hardware trigger not initialised
ADDIDATA_ANALOG_MEASURE_SOFTWARE_TRIGGER_NOT_AVAILABLE	-450	Software trigger not available
ADDIDATA_ANALOG_MEASURE_SOFTWARE_TRIGGER_MODE_NOT_AVAILABLE	-451	The selected mode for the software trigger is not available
ADDIDATA_ANALOG_MEASURE_SOFTWARE_TRIGGER_FLAG_ERROR	-452	The flag of the software trigger is wrong
ADDIDATA_ANALOG_MEASURE_SOFTWARE_TRIGGER_ALREADY_USED	-453	Software trigger already initialised by another process
ADDIDATA_ANALOG_MEASURE_SOFTWARE_TRIGGER_NOT_INITIALISED	-454	Software trigger not initialised
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_FLAG_ERROR	-455	The flag of the hardware gate is wrong
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_ALREADY_USED	-456	Hardware gate already initialised by another process
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_NOT_INITIALISED	-457	Hardware gate not initialised
ADDIDATA_ANALOG_MEASURE_SEQUENCE_SAME_MODULE_REQUIRED	-458	All channels for the sequence process must be from the same module
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DELAY_NOT_AVAILABLE	-459	The sequence delay is not available
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DELAY_VALUE_ERROR	-460	The delay value cannot be used for the sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DELAY_TIME_UNIT_ERROR	-461	The selected delay time unit is wrong
ADDIDATA_ANALOG_MEASURE_SEQUENCE_ALREADY_STARTED	-462	The selected sequence is already started
ADDIDATA_ANALOG_MEASURE_SEQUENCE_COUNTER_VALUE_ERROR	-463	The number of selected sequences is wrong
ADDIDATA_ANALOG_MEASURE_SEQUENCE_INTERRUPT_COUNTER_VALUE_ERROR	-464	The number of interrupt sequences is wrong
ADDIDATA_ANALOG_MEASURE_SEQUENCE_NOT_STARTED	-465	The selected sequence is not started
ADDIDATA_ANALOG_MEASURE_SEQUENCE_NOT_STOPPED	-466	The selected sequence is not stopped
ADDIDATA_ANALOG_MEASURE_SEQUENCE_CHANNEL_SELECTION_ERROR	-467	Since the sequence is determined by hardware, the channel selection is wrong. The first channel must be the first channel of the module and the last channel must be the last channel of the module.
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DELAY_MODE_NOT_AVAILABLE	-468	The selected sequence delay mode is not available.
ADDIDATA_ANALOG_MEASURE_SEQUENCE_NO_FREE_HANDLE_FOUND	-469	No free handle found for the sequence initialisation. Max 10 handles for one module available
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MEMORY_ALLOCATION_ERROR	-470	Sequence memory allocation error.
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DESCRIPTION_LIST_CREATION_ERROR	-471	Error in the memory allocation for the sequence description list
ADDIDATA_ANALOG_MEASURE_SEQUENCE_HANDLE_ERROR	-472	Selected sequence handle is wrong
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DELAY_MODE_SETTING_ERROR	-473	Error by initialising the sequence delay
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DESCRIPTION_LIST_SETTING_ERROR	-474	Error in the initialisation of the sequence description list
ADDIDATA_ANALOG_MEASURE_SEQUENCE_RESET_DESCRIPTION_LIST_ERROR	-475	Error by resetting the sequence description list
ADDIDATA_ANALOG_MEASURE_SEQUENCE_ENABLE_TRANSFER_ERROR	-476	Error for enabling the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_START_TRANSFER_ERROR	-477	Error for starting the transfer sequence

ADDIDATA_ANALOG_MEASURE_SEQUENCE_STOP_TRANSFER_ERROR	-478	Error for stopping the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_CONTINUE_TRANSFER_ERROR	-479	Error for continuing the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_PAUSE_TRANSFER_ERROR	-480	Error by transfer sequence pause
ADDIDATA_ANALOG_MEASURE_SEQUENCE_ABORT_TRANSFER_ERROR	-481	Error for aborting the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_ENABLE_TRANSFER_INTERRUPT_ERROR	-482	Error by enabling the interrupt of the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_DISABLE_TRANSFER_INTERRUPT_ERROR	-483	Error by disabling the interrupt of the transfer sequence
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_INITIALISATION_ERROR	-484	Error in the initialisation of the sequence list
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_MODE_INITIALISATION_ERROR	-485	Error in the initialisation of the sequence mode
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_MODE_CLEAR_INDEX_ERROR	-486	Error by clearing the sequence index
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_START_ERROR	-487	Error by starting the sequence module
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_PAUSE_ERROR	-488	Error in the pause of the sequence module
ADDIDATA_ANALOG_MEASURE_SEQUENCE_MODULE_STOP_ERROR	-489	Error by stopping the sequence module
ADDIDATA_TRANSDUCER_TYPE_SELECTION_ERROR	-490	Selected transducer type is not available
ADDIDATA_TRANSDUCER_FREQUENCY_SELECTION_ERROR	-491	Selected frequency for the selected transducer is not available
ADDIDATA_TRANSDUCER_PRIMARY_MODULE_CONNECTION_TEST_ALREADY_USED	-492	Transducer module connection test already initialised by another process
ADDIDATA_TRANSDUCER_PRIMARY_MODULE_CONNECTION_FLAG_ERROR	-493	The flag of the primary connection test is wrong
ADDIDATA_TRANSDUCER_NO_SENSOR_CONNECTED	-494	No transducer connected to the module
ADDIDATA_ANALOG_MEASURE_AUTO_REFRESH_NOT_STARTED	-495	Auto refresh acquisition not started
ADDIDATA_ANALOG_MEASURE_AUTO_REFRESH_NOT_STOPPED	-496	Auto refresh acquisition not stopped
ADDIDATA_TRANSDUCER_PRIMARY_SHORT_CIRCUIT_OCCUR	-497	Primary short circuit occurred
ADDIDATA_ANALOG_MEASURE_SEQUENCE_SAME_FREQUENCY_REQUIRED	-498	Same frequency required for all transducers
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_CYCLE_ERROR	-499	Hardware trigger cycle counter value is wrong
ADDIDATA_ANALOG_MEASURE_CONVERT_TIME_DIVISION_FACTOR_ALREADY_USED	-900	Conversion time division factor already initialised by another process
ADDIDATA_ANALOG_MEASURE_CONVERT_TIME_DIVISION_FACTOR_VALUE_ERROR	-901	Convert time division factor selection is wrong
ADDIDATA_ANALOG_MEASURE_CONVERT_TIME_DIVISION_FACTOR_NOT_INITIALISED	-902	Convert time division factor not initialised
ADDIDATA_ANALOG_MEASURE_CREATE_THREAD_ERROR	-903	Thread error created
ADDIDATA_ANALOG_MEASURE_CHANNEL_GROUP_INITIALISATION_NOT_THE_SAME	-904	Channel group initialisation is not the same
ADDIDATA_ANALOG_MEASURE_AVERAGE_VALUE_ERROR	-905	Average value error

## Digital input errors

ADDIDATA_DIGITAL_INPUT_CHANNEL_NUMBER_ERROR	-500	The number of the digital input channel is wrong
ADDIDATA_DIGITAL_INPUT_PORT_NUMBER_ERROR	-501	The number of the digital input port is wrong
ADDIDATA_DIGITAL_INPUT_ACCESS_ERROR	-502	No digital input access port is available
ADDIDATA_DIGITAL_INPUT_INTERRUPT_ALREADY_USED	-503	The digital input interrupt is already used
ADDIDATA_DIGITAL_INPUT_INTERRUPT_ALREADY_RELEASED	-504	The digital input interrupt is already released
ADDIDATA_DIGITAL_INPUT_INTERRUPT_FLAG_ERROR	-505	The digital input interrupt flag is wrong
ADDIDATA_DIGITAL_INPUT_INTERRUPT_LOGIC_ERROR	-506	The digital input interrupt logic is wrong
ADDIDATA_DIGITAL_INPUT_INFORMATION_STRUCTURE_INVALID_SIZE	-507	The structure size passed is invalid
ADDIDATA_DIGITAL_INPUT_LEVEL_VALUE_SELECTION_ERROR	-508	The selection of the input level value is invalid
ADDIDATA_DIGITAL_INPUT_CHANNEL_ARRAY_SIZE_ERROR	-509	The channel array size passed is invalid
ADDIDATA_DIGITAL_INPUT_CHANNEL_SELECTION_ERROR	-510	The number of the digital input channel is wrong
ADDIDATA_DIGITAL_INPUT_MODULE_SELECTION_ERROR	-511	The number of the digital input module is wrong
ADDIDATA_DIGITAL_INPUT_FILTER_UNIT_ERROR	-512	The selected time unit for the filter is wrong
ADDIDATA_DIGITAL_INPUT_FILTER_DELAY_VALUE_ERROR	-513	Error in the filter time selection
ADDIDATA_DIGITAL_INPUT_FILTER_NOT_INITIALISED	-514	Digital input filter not initialised
ADDIDATA_DIGITAL_INPUT_FILTER_FLAG_ERROR	-515	The digital input filter flag parameter is wrong
ADDIDATA_DIGITAL_INPUT_LEVEL_FLAG_ERROR	-516	The digital input level flag parameter is wrong
ADDIDATA_DIGITAL_INPUT_MASK_ERROR	-517	The set mask for digital input interrupt is wrong

## Digital output errors

ADDIDATA_DIGITAL_OUTPUT_CHANNEL_NUMBER_ERROR	-600	The number of the digital output channel is wrong
ADDIDATA_DIGITAL_OUTPUT_PORT_NUMBER_ERROR	-601	The number of the digital output port is wrong
ADDIDATA_DIGITAL_OUTPUT_VALUE_ERROR	-602	The digital output value is wrong
ADDIDATA_DIGITAL_OUTPUT_ACCESS_ERROR	-603	No digital output access port is available
ADDIDATA_DIGITAL_OUTPUT_MEMORY_ALREADY_ENABLED	-604	The digital output memory is already active
ADDIDATA_DIGITAL_OUTPUT_MEMORY_ALREADY_DISABLED	-605	The digital output memory is already deactivated



ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_ALREADY_USED	-606	The digital output interrupt is already used
ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_ALREADY_RELEASED	-607	The digital output interrupt is already released
ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_FLAG_ERROR	-608	The digital output interrupt flag is wrong

## Constant

## Return value

## Meaning

## Analog output errors

ADDIDATA_ANALOG_OUTPUT_CHANNEL_NUMBER_ERROR	-700	The analog output channel number is wrong
ADDIDATA_ANALOG_OUTPUT_VOLTAGE_MODE_ERROR	-701	The analog output voltage mode is wrong
ADDIDATA_ANALOG_OUTPUT_POLARITY_ERROR	-702	The analog output polarity is wrong
ADDIDATA_ANALOG_OUTPUT_VALUE_ERROR	-703	The analog output value is wrong
ADDIDATA_ANALOG_OUTPUT_ENABLE_DISABLE_SYNC_ERROR	-704	No analog output synchronisation is available
ADDIDATA_ANALOG_OUTPUT_ALREADY_USED	-705	Analog output channel already used by another process
ADDIDATA_ANALOG_OUTPUT_TIMEOUT	-706	Timeout has occurred
ADDIDATA_ANALOG_OUTPUT_TIMEOUT_ERROR	-707	Conversion timeout for the analog outputs could not be measured
ADDIDATA_ANALOG_OUTPUT_ALREADY_RELEASED	-708	Analog output already released

## Error sources

## Common function errors

## pdw\_ErrorSource

## Pdw\_ErrorSource contain

## Timer errors

ADDIDATA_TIMER_NUMBER_ERROR	The selected timer number
ADDIDATA_TIMER_RELOAD_VALUE_ERROR	The selected wrong timer reload value
ADDIDATA_TIMER_UNIT_ERROR	The selected wrong time unit
ADDIDATA_TIMER_MODE_ERROR	The selected wrong timer mode
ADDIDATA_TIMER_INTERRUPT_FLAG_ERROR	Timer selected wrong interrupt flag
ADDIDATA_TIMER_HARDWARE_GATE_LEVEL_ERROR	Timer selected wrong hardware gate
ADDIDATA_TIMER_HARDWARE_GATE_LEVEL_SELECTION_ERROR	The selected wrong hardware gate level
ADDIDATA_TIMER_HARDWARE_GATE_FLAG_ERROR	The selected wrong flag of the hardware gate
ADDIDATA_TIMER_HARDWARE_TRIGGER_LEVEL_ERROR	Timer selected wrong hardware trigger level
ADDIDATA_TIMER_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	Hardware selected wrong trigger level
ADDIDATA_TIMER_HARDWARE_TRIGGER_FLAG_ERROR	The selected wrong flag of the hardware trigger
ADDIDATA_TIMER_HARDWARE_OUTPUT_LEVEL_ERROR	Timer selected wrong output level selection
ADDIDATA_TIMER_HARDWARE_OUTPUT_LEVEL_SELECTION_ERROR	Timer selected wrong output level selection
ADDIDATA_TIMER_HARDWARE_OUTPUT_FLAG_ERROR	The selected wrong flag of the timer output
ADDIDATA_TIMER_ALREADY_USED	The selected wrong timer number

## Counter errors

ADDIDATA_COUNTER_NUMBER_ERROR	The selected wrong counter number
ADDIDATA_COUNTER_RELOAD_VALUE_ERROR	The selected wrong counter reload value
ADDIDATA_COUNTER_COUNTER_UP_DOWN_FLAG_ERROR	Counter selected wrong Up/Down selection
ADDIDATA_COUNTER_INPUT_LEVEL_ERROR	Counter selected wrong input level selection
ADDIDATA_COUNTER_INTERRUPT_FLAG_ERROR	Counter selected wrong interrupt flag
ADDIDATA_COUNTER_HARDWARE_GATE_LEVEL_ERROR	Counter selected wrong hardware gate
ADDIDATA_COUNTER_HARDWARE_GATE_LEVEL_SELECTION_ERROR	The selected wrong hardware gate level
ADDIDATA_COUNTER_HARDWARE_GATE_FLAG_ERROR	The selected wrong hardware flag
ADDIDATA_COUNTER_HARDWARE_TRIGGER_LEVEL_ERROR	Counter selected wrong hardware trigger
ADDIDATA_COUNTER_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	Hardware selected wrong trigger level
ADDIDATA_COUNTER_HARDWARE_TRIGGER_FLAG_ERROR	The selected wrong hardware trigger flag
ADDIDATA_COUNTER_HARDWARE_OUTPUT_LEVEL_ERROR	Counter selected wrong output level
ADDIDATA_COUNTER_HARDWARE_OUTPUT_LEVEL_SELECTION_ERROR	Counter selected wrong output level
ADDIDATA_COUNTER_HARDWARE_OUTPUT_FLAG_ERROR	The selected wrong flag of the counter output
ADDIDATA_COUNTER_ALREADY_USED	The selected wrong counter number

## Watchdog errors

ADDIDATA_WATCHDOG_NUMBER_ERROR	The selected wrong watchdog number
ADDIDATA_WATCHDOG_DELAY_VALUE_ERROR	The selected wrong watchdog delay
ADDIDATA_WATCHDOG_UNIT_ERROR	The selected wrong watchdog unit
ADDIDATA_WATCHDOG_INTERRUPT_FLAG_ERROR	The selected wrong interrupt flag
ADDIDATA_WATCHDOG_HARDWARE_GATE_LEVEL_ERROR	The selected wrong watchdog hardware gate
ADDIDATA_WATCHDOG_HARDWARE_GATE_LEVEL_SELECTION_ERROR	The selected wrong hardware gate level
ADDIDATA_WATCHDOG_HARDWARE_GATE_FLAG_ERROR	The selected wrong hardware gate flag
ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_LEVEL_ERROR	The selected wrong watchdog hardware trigger
ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	The selected wrong hardware trigger level

ADDIDATA_WATCHDOG_HARDWARE_TRIGGER_FLAG_ERROR	The selected wrong hardware trigger flag
ADDIDATA_WARNING_DELAY_VALUE_ERROR	The selected wrong warning delay value
ADDIDATA_WARNING_DELAY_TIME_UNIT_ERROR	The selected wrong warning time unit
ADDIDATA_WATCHDOG_WARNING_RELAY_FLAG_ERROR	The selected wrong warning relay flag
ADDIDATA_WATCHDOG_RESET_RELAY_FLAG_ERROR	The selected wrong reset relay flag
ADDIDATA_WATCHDOG_ALREADY_USED	The selected wrong watchdog number
ADDIDATA_WATCHDOG_RESET_RELAY_MODE_SELECTION_ERROR	The selected wrong reset relay mode.

### Analog measure errors

ADDIDATA_ANALOG_MEASURE_CHANNEL_NUMBER_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_CONVERSION_STARTED	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_CONVERTING_TIME_ERROR	The selected wrong converting time value
ADDIDATA_ANALOG_MEASURE_CONVERTING_TIME_UNIT_ERROR	The selected wrong converting time unit
ADDIDATA_ANALOG_MEASURE_INTERRUPT_FLAG_ERROR	The selected wrong interrupt flag
ADDIDATA_ANALOG_MEASURE_INTERRUPT_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_WARNING_VALUE_ERROR	The selected wrong warning value
ADDIDATA_ANALOG_MEASURE_CHANNEL_ALREADY_USED	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_WARNING_ALREADY_USED	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_GAIN_ERROR	The selected wrong gain
ADDIDATA_ANALOG_MEASURE_POLARITY_PARAMETER_ERROR	The selected wrong polarity
ADDIDATA_ANALOG_MEASURE_OFFSET_RANGE_ERROR	The selected wrong offset range
ADDIDATA_ANALOG_MEASURE_CHANNEL_NOT_INITIALISED	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_POLARITY_MODE_NOT_AVAILABLE	The selected wrong polarity mode
ADDIDATA_ANALOG_MEASURE_SCAN_MODE_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_SCAN_MODE_PARAMETER_ERROR	The selected wrong SCAN mode
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_MODE_NOT_AVAILABLE	The selected wrong external trigger mode
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_EXTERN_TRIGGER_PARAMETER_ERROR	The selected wrong external trigger parameter
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_MODE_NOT_AVAILABLE	The selected wrong external gate mode
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_EXTERN_GATE_PARAMETER_ERROR	The selected wrong external gate parameter
ADDIDATA_ANALOG_MEASURE_COMMON_GAIN_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_COMMON_POLARITY_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_COMMON_OFFSET_RANGE_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_NOT_AVAILABLE	The selected wrong SCAN delay mode
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_VALUE_ERROR	The selected wrong SCAN delay value
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_INITIALISED	The selected wrong SCAN handle
ADDIDATA_ANALOG_MEASURE_MODULE_NOT_AVAILABLE	The selected wrong module number
ADDIDATA_ANALOG_MEASURE_SCAN_ALREADY_STARTED	The selected SCAN handle
ADDIDATA_ANALOG_MEASURE_SCAN_COUNTER_VALUE_ERROR	The selected wrong SCAN counter value
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_STARTED	The selected SCAN handle
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_STOPPED	The selected SCAN handle
ADDIDATA_ANALOG_MEASURE_SCAN_SAME_MODULE_REQUIRED	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_SCAN_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_SCAN_CHANNEL_SELECTION_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_SCAN_DELAY_MODE_NOT_AVAILABLE	The selected wrong SCAN delay mode
ADDIDATA_ANALOG_MEASURE_COUPLING_MODE_NOT_AVAILABLE	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_COMMON_COUPLING_ERROR	The selected wrong channel number
ADDIDATA_ANALOG_MEASURE_COUPLING_PARAMETER_ERROR	The selected wrong coupling mode parameter
ADDIDATA_ANALOG_MEASURE_SIGN_TEST_PARAMETER_ERROR	The selected wrong sign test parameter

### Digital input errors

ADDIDATA_DIGITAL_INPUT_CHANNEL_NUMBER_ERROR	The selected wrong digital input channel
ADDIDATA_DIGITAL_INPUT_PORT_NUMBER_ERROR	The selected wrong digital input port
ADDIDATA_DIGITAL_INPUT_INTERRUPT_ALREADY_USED	The selected wrong digital input channel
ADDIDATA_DIGITAL_INPUT_INTERRUPT_ALREADY_RELEASED	The selected wrong digital input channel
ADDIDATA_DIGITAL_INPUT_INTERRUPT_FLAG_ERROR	The selected wrong digital input channel
ADDIDATA_DIGITAL_INPUT_INTERRUPT_LOGIC_ERROR	The selected wrong digital input channel

**Digital output errors**

ADDIDATA_DIGITAL_OUTPUT_CHANNEL_NUMBER_ERROR	The selected wrong digital output channel
ADDIDATA_DIGITAL_OUTPUT_PORT_NUMBER_ERROR	The selected wrong digital output port
ADDIDATA_DIGITAL_OUTPUT_VALUE_ERROR	The selected wrong digital output channel
ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_ALREADY_USED	The selected wrong digital output channel
ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_ALREADY_RELEASED	The selected wrong digital output channel
ADDIDATA_DIGITAL_OUTPUT_INTERRUPT_FLAG_ERROR	The selected wrong digital output channel

**Analog output errors**

ADDIDATA_ANALOG_OUTPUT_CHANNEL_NUMBER_ERROR	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_VOLTAGE_MODE_ERROR	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_POLARITY_ERROR	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_VALUE_ERROR	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_ALREADY_USED	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_TIMEOUT	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_TIMEOUT_ERROR	The selected wrong analog output channel
ADDIDATA_ANALOG_OUTPUT_ALREADY_RELEASED	The selected wrong analog output channel

**Calling convention:****ANSI C :**

```

WORD      w_FunctionNumber;
INT        i_ErrorCode;
INT        i_ErrorLevel;
INT        i_ReturnValue;
DWORD dw_ErrorSource;

```

```

i_ReturnValue = i_ADDIDATA_GetLastErrorAndSource (dw_DriverHandle,
&w_FunctionNumber,
&i_ErrorCode,
&i_ErrorLevel,
&dw_ErrorSource);

```

**Return value:**

- 0: No error
- 1: Error message present
  - 101: Function cannot be called up in the user interrupt routine
  - 102: Error message not found

**Counter errors**

ADDIDATA_COUNTER_NUMBER_ERROR	-800	The selected counter number is wrong
ADDIDATA_COUNTER_RELOAD_VALUE_ERROR	-801	The counter reload value is wrong
ADDIDATA_COUNTER_COUNTER_UP_DOWN_FLAG_ERROR	-802	Counter Up/Down selection error
ADDIDATA_COUNTER_INPUT_LEVEL_ERROR	-803	Counter input level selection error
ADDIDATA_COUNTER_INTERRUPT_FLAG_ERROR	-804	Counter interrupt flag is wrong
ADDIDATA_COUNTER_HARDWARE_GATE_LEVEL_ERROR	-805	Counter hardware gate not available
ADDIDATA_COUNTER_HARDWARE_GATE_LEVEL_SELECTION_ERROR	-806	The selected hardware gate level is wrong
ADDIDATA_COUNTER_HARDWARE_GATE_FLAG_ERROR	-807	The flag of the hardware gate is wrong
ADDIDATA_COUNTER_HARDWARE_TRIGGER_LEVEL_ERROR	-808	Counter hardware trigger not available
ADDIDATA_COUNTER_HARDWARE_TRIGGER_LEVEL_SELECTION_ERROR	-809	Hardware trigger level selection error
ADDIDATA_COUNTER_HARDWARE_TRIGGER_FLAG_ERROR	-810	The flag of the hardware trigger is wrong
ADDIDATA_COUNTER_HARDWARE_OUTPUT_LEVEL_ERROR	-811	Counter output not available
ADDIDATA_COUNTER_HARDWARE_OUTPUT_LEVEL_SELECTION_ERROR	-812	Counter output level selection error
ADDIDATA_COUNTER_HARDWARE_OUTPUT_FLAG_ERROR	-813	The flag of the counter output is wrong
ADDIDATA_COUNTER_ALREADY_USED	-814	Counter already used by another process or used as a watchdog

ADDIDATA_COUNTER_INFORMATION_STRUCTURE_INVALID_SIZE	-815	The specified size is wrong.
-----------------------------------------------------	------	------------------------------

**Calling convention:**ANSI C :

WORD w\_FunctionNumber;

INT i\_ErrorCode;

INT i\_ErrorLevel;

INT i\_ReturnValue;

i\_ReturnValue = i\_ADDIDATA\_GetLastError(dw\_DriverHandle,  
&w\_FunctionNumber,  
&i\_ErrorCode,  
&b\_ErrorLevel);

**Return value:**

0: No error

1: Error message present

- 101: Function cannot be called up in the user interrupt routine

- 102: Error message not found

3) **b\_ADDIDATA\_EnableErrorMessage (...)****i****IMPORTANT!****This function is only available for 32-bit operating systems****Syntax:**

```
<Return value> = b_ADDIDATA_EnableErrorMessage
                    (DWORD dw_DriverHandle,
                     HANDLE h_WndHandle,
                     WORD   w_Message)
```

**Parameters:****- Input:**

DWORD <i>dw_DriverHandle</i>	Handle of the driver
HANDLE <i>h_WndHandle</i>	Handle of the process/task
WORD <i>w_Message</i>	Number of the message to be sent

**- Output:**

No output has occurred.

**Task:**

Allows the transmission of a message to a process or task, when an error has occurred by calling up an **ADDI-DATA** API function.

**Calling convention:**

ANSI C:

```
BYTE      b_ReturnValue;
HANDLE     h_WndHandle;
```

```
b_ReturnValue = b_ADDIDATA_EnableErrorMessage (dw_DriverHandle
                                                h_WndHandle
                                                WM_USER);
```

**Return value:**

- 1: No error
- 0: Error by calling up the function. Use the function "i\_ADDIDATA\_GetLastError", to find the error source.

- To receive the error message under Visual C++ proceed as follows:

In the CALLBACK WndProc(HWND hWnd, UINT nMessage, WPARAM wParam, LPARAM lParam) function, add the case WM\_USER.

- To receive the error message under Delphi proceed as follows:

```
// Declare the message
const
    CM_ERRORMESSAGE = WM_USER;

// Declare the function to be called up when the message occurs
TSampleForm = class(TForm)
    ExecutionBox: TGroupBox;
    ...
protected
    procedure CMErrorMessage(var Message: TMessage);
message CM_ERRORMESSAGE;
private
    { Private-Deklarationen }
public
    { Public-Deklarationen }
end;

// Delphi will call up this function once the message has occurred
procedure TSampleForm.CMErrorMessage(var Message:
    TMessage);
Begin
    ...
    inherited;
end;
```



### **WARNING!**

Call up the function `b_ADDIDATA_EnableErrorMessageEnable` only once the window is active.

In the creating method of the window, the windows is active only at the end of the creating method.

If you do not proceed in this way, the `GetActiveWindow ()` will return 0.

No message can be then returned.

#### 4) b\_ADDIDATA\_DisableErrorMessage (...)

##### Syntax:

<Return value> = b\_ADDIDATA\_DisableErrorMessage (DWORD dw\_DriverHandle)

##### Parameters:

###### - Input:

DWORD *dw\_DriverHandle*                      Handle of the driver

###### - Output:

No output has occurred.

##### Task:

Disables the transmission of a message to a process or task, when an error has occurred by calling up an **ADDI-DATA** API function.

**i**

### IMPORTANT!

**This function is only available for 32-bit operating systems.**

##### Calling convention:

ANSI C:

```
BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
```

```
b_ReturnValue = b_ADDIDATA_DisableErrorMessage (dw_DriverHandle);
```

##### Return value:

- 1: No error
- 0: Error by call-up of the function. Use the function "i\_ADDIDATA\_GetLastError", to find the error source.

- To receive the error message under Visual C++ proceed as follows:

In the CALLBACK WndProc(HWND hWnd, UINT nMessage, WPARAM wParam, LPARAM lParam) function, add the case WM\_USER.

- To receive the error message under Delphi proceed as follows:

```
// Declare the message
const
    CM_ERRORMESSAGE = WM_USER;

// Declare the function to be called up when the message occurs
TSampleForm = class(TForm)
    ExecutionBox: TGroupBox;
    ...
protected
    procedure CMErrorMessage(var Message: TMessage);
message CM_ERRORMESSAGE;
private
    { Private-Deklarationen }
```



```
public
{ Public-Deklarationen }
end;

// Delphi will call up this function once the message has occurred
procedure TSampleForm.CMErrorMessage (var Message:
    TMessage);
Begin
    ...
    inherited;
end;
```

**WARNING!**

Call up the function `b_ADDIDATA_EnableErrorMessageEnable` only once the window is active.

In the creating method of the window, the windows is active only at the end of the creating method.

If you do not proceed in this way, the `GetActiveWindow ()` will return 0.  
No message can be then returned.

## 5) b\_ADDIDATA\_FormatErrorMessage (...)

### Syntax:

<Return value> = b\_ADDIDATA\_FormatErrorMessage  
 (DWORD dw\_DriverHandle  
 INT i\_ErrorNumber,  
 PCHAR pc\_ErrorString,  
 WORD w\_ErrorStringSize,  
 WORD w\_FunctionNumber,  
 PCHAR pc\_FunctionName,  
 WORD w\_FunctionStringSize)

### Parameters:

#### - Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the driver
INT	<i>i_ErrorNumber</i>	Error code of the function "i_ADDIDATA_GetLastError"
WORD	<i>w_ErrorStringSize</i>	Size of the pc_ErrorString parameter.
WORD	<i>w_FunctionNumber</i>	Function number. This information is transmitted through the function "i_ADDIDATA_GetLastError"
WORD	<i>w_FunctionStringSize</i>	Size of the pc_FunctionName parameter.

#### - Output:

PCHAR	<i>pc_ErrorString</i>	Character string including the error
PCHAR	<i>pc_FunctionName</i>	Character string including the name of the function which has generated the error.

### Task:

Returns the character strings of the error and of the function name for the *i\_ErrorNumber* and *pc\_FunctionName* parameters.

### Calling convention:

#### ANSI C :

```
BYTE  b_ReturnValue;
CHAR  c_ErrorString[50];
CHAR  c_FunctionName[50];
```

```
b_ReturnValue = i_ADDIDATA_FormatErrorMessage (dw_DriverHandle
-100,
c_ErrorString,
sizeof (c_ErrorString),
301,
c_FunctionName,
sizeof (c_FunctionName));
```

### Return value:

1: No error  
 0: Error by call-up of the function. Use the function "i\_ADDIDATA\_GetLastError",  
 to find the error source.