



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

Software description

ADDIDRIVER

Digital output channels

Edition: 07.01 – 08/2006

1	INTRODUCTION	1
2	DIGITAL OUTPUT CHANNELS	2
1)	b_ADDIDATA_GetNumberOfDigitalOutputs(...)	2
2)	b_ADDIDATA_GetDigitalOutputInformation (...)	3
3)	b_ADDIDATA_SetDigitalOutputMemoryOn (...)	5
4)	b_ADDIDATA_SetDigitalOutputMemoryOff (...)	6
4)	b_ADDIDATA_SetDigitalOutputMemoryOnEx (...)	6
5)	b_ADDIDATA_Set1DigitalOutputOn (...)	8
6)	b_ADDIDATA_Set1DigitalOutputOff (...)	9
7)	b_ADDIDATA_Set2DigitalOutputsOn (...)	10
8)	b_ADDIDATA_Set2DigitalOutputsOff (...)	11
9)	b_ADDIDATA_Set4DigitalOutputsOn (...)	12
10)	b_ADDIDATA_Set4DigitalOutputsOff (...)	13
11)	b_ADDIDATA_Set8DigitalOutputsOn (...)	14
12)	b_ADDIDATA_Set8DigitalOutputsOff (...)	16
13)	b_ADDIDATA_Set16DigitalOutputsOn (...)	17
14)	b_ADDIDATA_Set16DigitalOutputsOff (...)	18
15)	b_ADDIDATA_Set32DigitalOutputsOn (...)	19
16)	b_ADDIDATA_Set32DigitalOutputsOff (...)	20
17)	b_ADDIDATA_Get1DigitalOutputStatus (...)	21
18)	b_ADDIDATA_Get2DigitalOutputStatus (...)	22
19)	b_ADDIDATA_Get4DigitalOutputStatus (...)	23
20)	b_ADDIDATA_Get8DigitalOutputStatus (...)	24
21)	b_ADDIDATA_Get16DigitalOutputStatus (...)	25
22)	b_ADDIDATA_Get32DigitalOutputStatus (...)	26
23)	b_ADDIDATA_InitDigitalOutputInterrupt (...)	27
24)	b_ADDIDATA_ReleaseDigitalOutputInterrupt (...)	29
25)	b_ADDIDATA_EnableDisableDigitalOutputInterrupt (...)	30

Tables

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

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
VOID	void	void	pointer	any
BYTE	unsigned char	unsigned char	byte	integer
INT	int	int	integer	integer
WORD	unsigned short int	unsigned short int	long	long
DWORD	long	long	longint	long
PBYTE	unsigned char *	unsigned char *	var byte	integer
PINT	int *	int *	var integer	integer
PWORD	unsigned short int *	unsigned short int *	var long	long
PCHAR	char *	char *	var string	string
PDWORD	long *	long *	var longint	long
DOUBLE	double	double	double	double

2 DIGITAL OUTPUT CHANNELS

1) b_ADDIDATA_GetNumberOfDigitalOutputs(...)

Syntax:

```
<Return value> = b_ADDIDATA_GetNumberOfDigitalOutputs
                    (DWORD dw_DriverHandle,
                     PWORD pw_NumberOfChannels)
```

Parameters:

- Input:

DWORD *dw_DriverHandle* Handle of the **ADDI-DATA driver**

- Output:

PWORD *pw_NumberOfChannels* Number of digital output channels

Task:

Returns the number of available digital output channels.

Calling convention:

ANSI C:

```
BYTE              b_ReturnValue;
DWORD             dw_DriverHandle;
WORD              w_NumberOfChannels;
```

```
b_ReturnValue = b_ADDIDATA_GetNumberOfDigitalOutput
                    (dw_DriverHandle,
                     &w_NumberOfChannels);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError",
to find the error source.

2) b_ADDIDATA_GetDigitalOutputInformation (...)

Syntax:

<Return value> = b_ADDIDATA_GetDigitalOutputInformation

(DWORD dw_DriverHandle,
WORD w_DigitalOutputNumber,
PBYTE pb_DigitalOutputType
PBYTE pb_DigitalOutputInterrupt)

Parameters:

- Input:

DWORD *dw_DriverHandle* Handle of the ADDI-DATA driver
WORD *w_DigitalOutputNumber* Number of the selected digital output

- Output:

PBYTE *pb_DigitalOutputType* Type of the selected input
 0 Output type not defined.
 1 – 15 5V Type
 1 Differential RS485/422
 2 Open drain, low-side, load at default voltage
 3 TTL
 16 – 31 12 Type
 32 – 47 24 Type
 32 High-side, load at ground
 33 Open drain, low side, load at default voltage
 34 24 V direct optocoupler output / drain source)
 48 – 255 any other type
 64-79 Relay
 65 Relay 1C (change-over contacts)
 PBYTE *pb_DigitalOutputInterrupt* Type of the possible interrupt
 1 = Vcc diagnostic,
 2 = CC-diagnostic.

Task:

Returns information about the digital output channels.

Calling convention:

ANSI C:

BYTE b_ReturnValue;
DWORD dw_DriverHandle;
BYTE b_DigitalOutputType
BYTE b_DigitalOutputInterrupt;

b_ReturnValue = b_ADDIDATA_GetDigitalOutputInformation

(dw_DriverHandle,
0,
&b_DigitalOutputType
&b_DigitalOutputInterrupt);

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

3) b_ADDIDATA_SetDigitalOutputMemoryOn (...)

Syntax:

<Return value> = b_ADDIDATA_SetDigitalOutputMemoryOn
(DWORD dw_DriverHandle)

Parameters:**- Input:**

DWORD *dw_DriverHandle* Handle of the ADDI-DATA driver

- Output:

No output signal has occurred.

Task:

Activates the digital output memory and sets all digital outputs off. After calling up this function, the outputs you have previously activated with the function

"i_ADDIDATA_SetXDigitalOutputOn" are not reset.

You can reset them with the function "i_ADDIDATA_SetXDigitalOutputOff".

Calling convention:ANSI C:

BYTE b_ReturnValue;
DWORD dw_DriverHandle;

b_ReturnValue = b_ADDIDATA_SetDigitalOutputMemoryOn
(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.

4)b_ADDIDATA_SetDigitalOutputMemoryOff (...)**Syntax:**

<Return value> = b_ADDIDATA_SetDigitalOutputMemoryOff
(DWORD dw_DriverHandle)

Parameters:**- Input:**

DWORD dw_DriverHandle Handle of the ADDI-DATA driver

- Output:

No output signal has occurred.

Task:

Deactivates the digital output memory.

Calling convention:

ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_SetDigitalOutputMemoryOff
(dw_DriverHandle);
```

Return value:

1: No error

0: Error by calling up of the function. Use the function

"i_ADDIDATA_GetLastError", to find the error source.

4) b_ADDIDATA_SetDigitalOutputMemoryOnEx (...)**Syntax:**

<Return value> = b_ADDIDATA_SetDigitalOutputMemoryOnEx
(DWORD dw_DriverHandle)

Parameters:**- Input:**

DWORD dw_DriverHandle Handle of the ADDI-DATA driver

- Output:

No output signal has occurred.

Task:

Activates the digital output memory and save the state from all digital outputs. After calling up this function, the outputs you have previously activated with the function

"i_ADDIDATA_SetXDigitalOutputOn" are not reset.

You can reset them with the function "i_ADDIDATA_SetXDigitalOutputOff".

!!! Attention

The digital outputs state from the APCI-1500/APCI-3120/APCI-3001/APCI-3200/APCI-3300 can not be read. This function call set the outputs from this board to off.

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_SetDigitalOutputMemoryOnEx  
                (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.

5) b_ADDIDATA_Set1DigitalOutputOn (...)

Syntax:

<Return value> = b_ADDIDATA_Set1DigitalOutputOn
(DWORD dw_DriverHandle,
WORD w_Channel)

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_Channel</i>	Number of the output to be set.

- Output:

No output signal has occurred.

Task:

Sets the output which was passed through the variable *b_Channel*.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Set1DigitalOutputOn(dw_DriverHandle,0);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError",
to find the error source.

6) b_ADDIDATA_Set1DigitalOutputOff (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set1DigitalOutputOff
                                     (DWORD    dw_DriverHandle,
                                     WORD      w_Channel)
```

Parameters:

- Input:

DWORD dw_DriverHandle	Handle of the ADDI-DATA driver
BYTE w_Channel	Number of the output to be reset (0 to 31)

- Output:

No output signal has occurred.

Task:

Resets the output which was passed with the Variable *w_Channel*. Resetting an output means setting this output to "Low".

i

WICHTIG!

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_Set1DigitalOutputOff (dw_DriverHandle,1);
```

Return value:

1: No error

0: Error by callup of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

7) b_ADDIDATA_Set2DigitalOutputsOn (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set2DigitalOutputsOn
                                     (DWORD    dw_DriverHandle,
                                     BYTE      b_Port,
                                     BYTE      b_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 3)

- Output:

No output signal has occurred.

Task:

Sets one or several outputs of a port.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Calling convention:

ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_Set2DigitalOutputsOn (dw_DriverHandle,0,3);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

8) b_ADDIDATA_Set2DigitalOutputsOff (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Set2DigitalOutputsOff
                                     (DWORD    dw_DriverHandle,
                                     BYTE      b_Port,
                                     BYTE      b_PortValue)
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 3)

- Output:

No output signal has occurred.

Task:

Resets one or several outputs of a port.

Resetting one output means setting the output to "Low" (0). The other channels hold their state.

i**IMPORTANT!**

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_Set2DigitalOutputsOff (dw_DriverHandle,0,3);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

9) b_ADDIDATA_Set4DigitalOutputsOn (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set4DigitalOutputsOn
                                     (DWORD    dw_DriverHandle,
                                     BYTE      b_Port,
                                     BYTE      b_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 15)

- Output:

No output signal has occurred.

Task:

Sets one or several outputs of a port.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Calling convention:

ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_Set4DigitalOutputsOn (dw_DriverHandle,0,3);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

10) b_ADDIDATA_Set4DigitalOutputsOff (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Set4DigitalOutputsOff
                                     (DWORD    dw_DriverHandle,
                                     BYTE      b_Port,
                                     BYTE      b_PortValue)
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 15)

- Output:

No output signal has occurred.

Task:

Resets one or several outputs of a port.

Resetting one output means setting the output to "Low" (0). The other channels hold their state.

i**IMPORTANT!**

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_Set4DigitalOutputsOff (dw_DriverHandle,0,3);
```

Return value:

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

11) b_ADDIDATA_Set8DigitalOutputsOn (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set8DigitalOutputsOn (DWORD   dw_DriverHandle,
                                                    BYTE     b_Port,
                                                    BYTE     b_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 255)

- Output:

No output signal has occurred.

Task:

Sets one or several outputs of a port.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

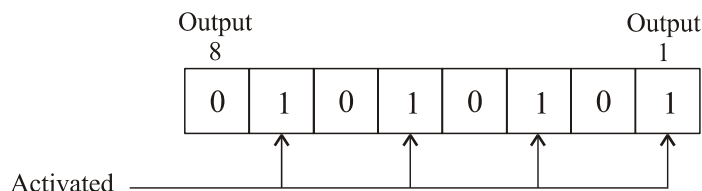
If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Example:

Switching on the digital output memory (ON)

see function "b_ADDIDATA_SetDigitalOutputMemoryOn (...)

```
b_Port    = 1
b_PortValue = 55 Hex
```

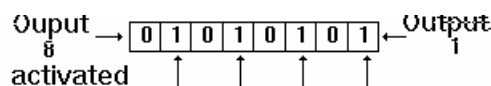


The outputs 1, 3, 5, 7 are set. The others outputs are hold their status.

Switching off the digital output memory (OFF)

see function "b_ADDIDATA_SetDigitalOutputMemoryOff (...)

```
b_Port    = 1
b_PortValue = 55 Hex
```



The outputs 1, 3, 5, 7 are set. The others are reset.

Calling convention:

ANSI C:

```
BYTE      b_ReturnValue;
```



```
DWORD      dw_DriverHandle;  
b_ReturnValue = b_ADDIDATA_Set8DigitalOutputsOn (dw_DriverHandle,1,1);
```

Return value:

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

12) b_ADDIDATA_Set8DigitalOutputsOff (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set8DigitalOutputsOff
                                     (DWORD    dw_DriverHandle,
                                     BYTE      b_Port,
                                     BYTE      b_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
BYTE	<i>b_PortValue</i>	Output value (0 to 255)

- Output:

No output signal has occurred.

Task:

Resets one or several outputs of a port.

Resetting one output means setting the output to "Low" (0). The other channels hold their state.

i

IMPORTANT!

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_Set8DigitalOutputsOff (dw_DriverHandle,1,1);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

13) b_ADDIDATA_Set16DigitalOutputsOn (...)

Syntax:

<Return value> = b_ADDIDATA_Set16DigitalOutputsOn
(DWORD dw_DriverHandle,
BYTE b_Port,
WORD w_PortValue)

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
WORD	<i>w_PortValue</i>	Output value (0 to 65535)

- Output:

No output signal has occurred.

Task:

Sets one or several outputs of a port.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Set16DigitalOutputsOn(dw_DriverHandle,0,1);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

14) b_ADDIDATA_Set16DigitalOutputsOff (...)

Syntax:

```
<Return value> = b_ADDIDATA_Set16DigitalOutputsOff
                                     (DWORD   dw_DriverHandle,
                                     BYTE     b_Port,
                                     WORD     w_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
WORD	<i>w_PortValue</i>	Output value (0 to 65535)

- Output:

No output signal has occurred.

Task:

Resets one or several outputs of a port.

Resetting one output means setting the output to "Low" (0). The other channels hold their state.

i

IMPORTANT!

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_Set16DigitalOutputsOff (dw_DriverHandle,1,1);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError", to find the error source.

15) b_ADDIDATA_Set32DigitalOutputsOn (...)

Syntax:

<Return value> = b_ADDIDATA_Set32DigitalOutputsOn	
	(DWORD dw_DriverHandle,
	BYTE b_Port,
	DWORD dw_PortValue)

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the output port
DWORD	<i>dw_PortValue</i>	Output value (0 to 2 ³² -1)

- Output:

No output signal has occurred.

Task:

Sets one or several outputs of a port.

Setting one output means setting the output to "High".

If you have switched on the digital output memory (ON), the selected outputs channels are set to "1". The other channels hold their state.

If you have switched off the digital output memory (OFF), the selected outputs are set to "1". The other channels are reset.

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Set32DigitalOutputsOn(dw_DriverHandle,0,1);
```

Return value:

1: No error

0: Error by call-up of the function. Use the function "i_ADDIDATA_GetLastError",
to find the error source.

16) b_ADDIDATA_Set32DigitalOutputsOff (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Set32DigitalOutputsOff
                    (DWORD dw_DriverHandle,
                     BYTE   b_Port,
                     DWORD dw_PortValue)
```

Parameters:**- Input:**

DWORD <i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE <i>b_Port</i>	Number of the output port
DWORD <i>dw_PortValue</i>	Output value (0 to $2^{32}-1$)

- Output:

No output signal has occurred.

Task:

Resets one or several outputs of a port.

Resetting one output means setting the output to "Low" (0). The other channels hold their state.

i**IMPORTANT!**

**You can use this function only if the digital output memory is ON.
See function b_ADDIDATA_SetDigitalOutputMemoryOn (..).**

Calling convention:

ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_Set32DigitalOutputsOff(dw_DriverHandle,0,1);
```

Return value:

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

17) b_ADDIDATA_Get1DigitalOutputStatus (...)

Syntax:

<Return value> = b_ADDIDATA_Get1DigitalOutputStatus		
	(DWORD	dw_DriverHandle,
	WORD	w_Channel,
	PBYTE	pb_ChannelStatus)

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_Channel</i>	Number of the output to be read

- Output:

PBYTE	<i>pb_ChannelStatus</i>	Status of the digital output
		0 -> low
		1 -> high

Task:

Return the status of a digital output. The variable *w_Channel* passes the output to be read (0 to 31). A value is returned through the variable *pb_ChannelStatus*: 0 (low) or 1 (high).

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;
BYTE	b_ChannelStatus;

[illegible]

Return value:

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

18) b_ADDIDATA_Get2DigitalOutputStatus (...)

Syntax:

<Return value> = b_ADDIDATA_Get2DigitalOutputStatus	
	(DWORD dw_DriverHandle,
	BYTE b_Port,
	PBYTE pb_PortValue)

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the 2-bit output port to be read

- Output:

PBYTE	<i>pb_PortValue</i>	Status of the digital output port (0 to 3)
-------	---------------------	---

Task:

Returns the status of a 2-bit port. The variable *b_Port* passes the port to be read. A value is returned through the variable *pb_PortValue*.

Calling convention:

ANSI C :

```

BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
BYTE      b_PortValue;

```

[illegible]

Return value:

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

19) b_ADDIDATA_Get4DigitalOutputStatus (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Get8DigitalOutputStatus
                                (DWORD dw_DriverHandle,
                                BYTE    b_Port,
                                PBYTE   pb_PortValue)
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the 4-bit output port to be read

- Output:

PBYTE	<i>pb_PortValue</i>	Status of the digital output port (0 to 15)
-------	---------------------	--

Task:

Returns the status of a 4-bit port. The variable *b_Port* passes the port to be read. A value is returned through the variable *pb_PortValue*.

Calling convention:ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_Get4DigitalOutputStatus (dw_DriverHandle,
                                                    1,
                                                    &b_PortValue);
```

Return value:

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

20) b_ADDIDATA_Get8DigitalOutputStatus (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Get8DigitalOutputStatus
                                (DWORD dw_DriverHandle,
                                BYTE    b_Port,
                                PBYTE   pb_PortValue)
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_Port</i>	Number of the 8-bit output port to be read

- Output:

PBYTE	<i>pb_PortValue</i>	Status of the digital output port (0 to 255)
-------	---------------------	---

Task:

Returns the status of an 8-bit port. The variable *b_Port* passes the port to be read. A value is returned through the variable *pb_PortValue*.

Calling convention:ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_Get8DigitalOutputStatus (dw_DriverHandle,
                                                    0,
                                                    &b_PortValue);
```

Return value:

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

21) b_ADDIDATA_Get16DigitalOutputStatus (...)**Syntax:**

```
<Return value> = b_ADDIDATA_Get16DigitalOutputStatus
                                (DWORD    dw_DriverHandle,
                                BYTE      b_PortNbr,
                                PWORD     pw_PortValue)
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_PortNbr</i>	Number of the 16-bit output port to be read

- Output:

PWORD	<i>pw_InputValue</i>	Digital output status of a port (0 to 65535)
-------	----------------------	--

Task:

Returns the status of a 16-bit port.

Calling convention:ANSI C:

```
BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
WORD      w_PortValue;
```

```
b_ReturnValue = b_ADDIDATA_Get16DigitalOutputStatus
                                (dw_DriverHandle,
                                0,
                                &w_PortValue);
```

Return value:

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

22) b_ADDIDATA_Get32DigitalOutputStatus (...)

Syntax:

```
<Return value> = b_ADDIDATA_Get32DigitalOutputStatus
                                     (DWORD   dw_DriverHandle,
                                     BYTE     b_PortNbr,
                                     PDWORD  pdw_PortValue)
```

Parameters:

- Input:

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
BYTE	<i>b_PortNbr</i>	Number of the 32-bit output port to be read (1 or 2)

- Output:

PDWORD <i>pdw_InputValue</i>	Digital output status of a port (0 to $2^{32}-1$)
------------------------------	--

Task:

Returns the status of a 32-bit port.

Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;
DWORD	dw_PortValue;

```
b_ReturnValue = b_ADDIDATA_Get32DigitalOutputStatus(dw_DriverHandle, 0, &dw_PortValue);
```

Return value:

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

23) b_ADDIDATA_InitDigitalOutputInterrupt (...)**Syntax:**

```

<Return value> = b_ADDIDATA_InitDigitalOutputInterrupt
                (DWORD          dw_DriverHandle,
                 WORD           w_FirstDigitalOutput,
                 WORD           w_LastDigitalOutput,
                 BYTE           b_CCInterruptFlag,
                 BYTE           b_VCCInterruptFlag,
                 PDWORD
pdw_DigitalOutputArrayForInterruptCC,
                 PDWORD
pdw_DigitalOutputArrayForInterruptVCC);

```

Parameters:**- Input:**

DWORD	dw_DriverHandle	Handle of the ADDI-DATA driver
WORD	w_FirstDigitalOutput	Number of the first digital output to be initialised.
WORD	w_LastDigitalOutput	Number of the last digital output to be initialised.
BYTE	b_CCInterruptFlag	Flag for the CC (short-circuit) Interrupt (0 or 1).
BYTE	b_VCCInterruptFlag	Flag for the VCC (voltage drop) interrupt (0 or 1).

- Output:

PDWORD pdw_DigitalOutputArrayForInterruptCC Mask of the digital output set for the CC Interrupt (bits set to 1: enable, bit set to 0: disabled).

PDWORD pdw_DigitalOutputArrayForInterruptVCC Mask of the digital output set for the VCC Interrupt (bits set to 1: enabled; bit set to 0: disabled).

Task:

Initialises the digital output interrupt.

Calling convention:ANSI C:

```

BYTE          b_ReturnValue;
DWORD  dw_DriverHandle;
DWORD  dw_DigitalOutputArrayForInterruptCC;
DWORD  dw_DigitalOutputArrayForInterruptVCC;

b_ReturnValue = b_ADDIDATA_InitDigitalOutputInterrupt
                (dw_DriverHandle,
                 0,
                 15,
                 ADDIDATA_ENABLE,
                 ADDIDATA_ENABLE,

```

```
&dw_DigitalOutputArrayForInterruptCC,  
&dw_DigitalOutputArrayForInterruptVCC);
```

Return value:

1: No error

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

24) b_ADDIDATA_ReleaseDigitalOutputInterrupt (...)**Syntax:**

```
<Return value> = b_ADDIDATA_ReleaseDigitalOutputInterrupt
                                (DWORD   dw_DriverHandle,
                                 WORD     w_FirstDigitalOutput,
                                 WORD     w_LastDigitalOutput);
```

Parameters:**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_FirstDigitalOutput</i>	Number of the first digital output to be released.
WORD	<i>w_LastDigitalOutput</i>	Number of the last digital output to be released.

- Output:

No output signal has occurred.

Task:

Releases the digital output interrupt.

Calling convention:ANSI C :

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

```
b_ReturnValue = b_ADDIDATA_ReleaseDigitalOutputInterrupt
                (dw_DriverHandle,
                 0,
                 15);
```

Return value:

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

i**IMPORTANT!**

The **interrupt mask** for the function is detailed in the "**Interrupt**" function description. (Tables 2-1 and 2-2).

25) b_ADDIDATA_EnableDisableDigitalOutputInterrupt (...)**Syntax:**

```
<Return value> = b_ADDIDATA_EnableDisableDigitalOutputInterrupt
                    (DWORD          dw_DriverHandle,
                     WORD           w_FirstDigitalOutput,
                     WORD           w_LastDigitalOutput,
                     BYTE           b_VCCInterruptFlag,
                     BYTE           b_CCInterruptFlag);
```

Parameters:**- Input:**

DWORD	dw_DriverHandle	Handle of the ADDI-DATA driver
WORD	w_FirstDigitalOutput	Number of the first digital output to be enabled/disabled.
WORD	w_LastDigitalOutput	Number of the last digital output to be enabled/disabled
BYTE	b_VCCInterruptFlag	Flag for the VCC Interrupt (0 or 1).
BYTE	b_CCInterruptFlag	Flag for the CC Interrupt (0 or 1).

- Output:

No output signal has occurred.

Task:

Enables or disables the digital output interrupt.

Calling convention:ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_EnableDisableDigitalOutputInterrupt
                (dw_DriverHandle,
                 0,
                 15,
                 ADDIDATA_ENABLE,
                 ADDIDATA_ENABLE);
```

Return value:

1: No error

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