



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

**Software description**

**ADDIDRIVER**

**Analogue output channels**

6<sup>th</sup> edition 05/2005



<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>2</b>	<b>ANALOGUE OUTPUT CHANNELS.....</b>	<b>2</b>
1)	b_ADDIDATA_GetNumberOfAnalogOutputs (...)	2
2)	b_ADDIDATA_GetAnalogOutputInformation (...)	3
3)	b_ADDIDATA_Init1AnalogOutput (...)	5
4)	b_ADDIDATA_InitMoreAnalogOutputs (...)	6
5)	b_ADDIDATA_Write1AnalogOutput (...)	7
6)	b_ADDIDATA_WriteMoreAnalogOutputs (...)	8
7)	b_ADDIDATA_Release1AnalogOutput (...)	9
8)	b_ADDIDATA_ReleaseMoreAnalogOutputs (...)	10
9)	b_ADDIDATA_EnableDisable1AnalogOutputSync (...)	11
10)	b_ADDIDATA_EnableDisableMoreAnalogOutputsSync (...)	12
11)	b_ADDIDATA_TriggerAnalogOutput (...)	13

## Tables

Table 1-1:	Type Declaration for Windows 98/NT/2000/XP	1
Table 2-1:	Type of analogue output	2
Table 2-2:	Output voltage range (high and low)	3
Table 2-3:	Output software and hardware polarity	4
Table 2-4:	Resolution	4



# 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 ANALOGUE OUTPUT CHANNELS

### 1) b\_ADDIDATA\_GetNumberOfAnalogOutputs (...)

#### Syntax:

```
<Return value> = b_ADDIDATA_GetNumberOfAnalogOutputs
                                     (DWORD dw_DriverHandle,
                                      PWORD pw_NumberOfChannels,
                                      PBYTE pb_AnalogOutputType)
```

#### Parameters:

##### - Input:

DWORD *dw\_DriverHandle*      Handle of the ADDI-DATA driver

##### - Output:

PWORD *pw\_NumberOfChannels*      Number of the analogue output channels

PBYTE *pb\_AnalogOutputType*      Type Array. Gives the type of each analogue output. See Table 2-1

*pb\_AnalogOutputType* [0]: Type of the first analog output.

*pb\_AnalogOutputType* [1]: Type of the second analog output.

...

*pb\_AnalogOutputType*

[\**pb\_AnalogOutputType* - 1]: Type of the last analog output

#### Task:

Returns the number of analogue outputs and the type of each analogue output.

**Table 2-1: Type of analogue output**

Define	Value	Definition
ADDIDATA_ANALOG_OUTPUT_DIRECT	0	Analog output is immediately written
ADDIDATA_ANALOG_OUTPUT_SYNC	1	Analog output can be written later

#### Calling convention:

##### ANSI C:

```
BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
WORD      w_NumberOfChannels;
BYTE      b_AnalogOutputType [12];
```

```
b_ReturnValue = b_ADDIDATA_GetNumberOfAnalogOutput
                (dw_DriverHandle,
                 &w_NumberOfChannels,
                 b_AnalogOutputType);
```

#### Return value:

1: No error

0: Error by calling up the function. Use the function "i\_ADDIDATA\_GetLastError", to find the error source.

**2) b\_ADDIDATA\_GetAnalogOutputInformation (...)****Syntax:**

<Return value> = b\_ADDIDATA\_GetAnalogOutputInformation  
 (DWORD dw\_DriverHandle,  
 WORD w\_ChannelNumber,  
 PBYTE pb\_NumberOfVoltageMode,  
 PBYTE pb\_HighVoltageRange,  
 PBYTE pb\_LowVoltageRange,  
 PBYTE pb\_SWPolarity,  
 PBYTE pb\_HWPolarity,  
 PBYTE pb\_Resolution,  
 PDWORD pdw\_Synchronisation)

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

DWORD *dw\_DriverHandle* Handle of the ADDI-DATA driver  
 WORD *w\_ChannelNumber* Analogue output number.  
 The first analogue output begins from 0.

**- Output:**

PBYTE *pb\_NumberOfVoltageMode* Number of voltage modes which can be used for the channel  
 PBYTE *pb\_HighVoltageRange* Possible voltage range for the analogue output in V. See Table 2-2.  
 PBYTE *pb\_LowVoltageRange* Possible voltage range for the analogue output in hundredths of V. See Table 2-2.  
 PBYTE *pb\_SWPolarity* Possible software polarity for the analogue output. See Table 2-3.  
 PBYTE *pb\_HWPolarity* Possible hardware polarity for the analogue output. See Table 2-3.  
 PBYTE *pb\_Resolution* Possible resolution for the analogue output. See Table 2-4.  
 PDWORD *pdw\_Synchronisation* If = 1, synchronisation is available for the analogue output.

**Task:**

Gives the number of modes (*pb\_NumberOfVoltageMode*), the voltage range (*pb\_HighVoltageRange* and *pb\_LowVoltageRange*), the polarity (*pb\_SWPolarity* and *pb\_HWPolarity*), the synchronisation (*pb\_Synchronisation*) and the resolution (*pb\_Resolution*) which can be set for the selected analog output (*b\_AnalogOutputNumber*).

**Table 2-2: Output voltage range (high and low)**

Define	Value	Definition
ADDIDATA_RANGE_0	0	First output range supported by the board
ADDIDATA_RANGE_1	1	Second output range supported by the board
ADDIDATA_RANGE_2	2	Third output range supported by the board
ADDIDATA_RANGE_3	3	Fourth output range supported by the board

**i****IMPORTANT!**

Please refer to the specific technical manual of the used board (Software functions and samples) to know the correct range for your board.

**Table 2-3: Output software and hardware polarity**

Define	Value	Definition
ADDIDATA_NONE	0	No polarity available.
ADDIDATA_UNIPOLAR	1	0 V to Max Voltage Range*
ADDIDATA_BIPOLAR	2	- Max Voltage Range* to + Max Voltage Range*
ADDIDATA_BOTH	3	Both are available

\* Max voltage range = High voltage range + (Low voltage range x 0,01)

**Table 2-4: Resolution**

Value	Resolution
10	10-bit
11	11-bit
12	12-bit
13	13-bit
14	14-bit
15	15-bit
16	16-bit
24	24-bit
32	32-bit

**Calling convention:**ANSI C:

```

BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
BYTE      b_NumberOfMode;
BYTE      b_HighRange[4];
BYTE      b_LowRange[4];
BYTE      b_Polarity[4];
BYTE      b_Resolution[4];
DWORD     dw_Synchronisation;

```

```

b_ReturnValue = b_ADDIDATA_AnalogOutputInformation
                (dw_DriverHandle,
                 0,
                 &b_NumberOfVoltageMode,
                 &b_HighRange[0],
                 &b_LowRange[0],
                 &b_Polarity[0],
                 &b_Resolution[0],
                 &dw_Synchronisation)

```

**Return value:**

1: No error

0: Error by calling up the function. Use the function "i\_ADDIDATA\_GetLastError", to find the error source.



### 3) b\_ADDIDATA\_Init1AnalogOutput (...)

### Syntax:

<Return Value> = b_ADDIDATA_Init1AnalogOutput	
(DWORD	dw_DriverHandle,
WORD	w_ChannelNumber,
BYTE	b_VoltageMode,
BYTE	b_Polarity)

### Parameters:

**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_ChannelNumber</i>	Number of the analogue output to be initialised
		The first analogue output begins from 0.
BYTE	<i>b_VoltageMode</i>	Selected voltage range. See Table 2-2.
BYTE	<i>b_Polarity</i>	Selected polarity. See Table 2-3.

**- Output:**

No output signal has occurred.

### Task:

Initialises the selected analogue output with the given range and polarity.

### Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Init1AnalogOutput
                (dw_DriverHandle,
                 0,
                 ADDIDATA_RANGE_0,
                 ADDIDATA_UNIPOLAR);
```

**Return value:**

1: No error

0: Error by calling up the function. Use the function "i\_ADDIDATA\_GetLastError",  
to find the error source.

#### 4) **b\_ADDIDATA\_InitMoreAnalogOutputs (...)**

##### Syntax:

<Return Value> = b\_ADDIDATA\_InitMoreAnalogOutputs  
 (DWORD dw\_DriverHandle,  
 WORD w\_NumberOfChannels,  
 PWORD pw\_ChannelNumber,  
 PBYTE pb\_VoltageMode,  
 PBYTE pb\_Polarity)

##### Parameters:

###### - Input:

DWORD <i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD <i>w_NumberOfChannels</i>	Number of the analogue outputs to be initialised. The first analogue output begins from 0.
PWORD <i>pw_ChannelNumber</i>	Analogueoutput array.
PBYTE <i>pb_VoltageMode</i>	Voltage or current range array. See Table 2-2.
PBYTE <i>pb_Polarity</i>	Polarity array. See Table 2-3.

###### - Output:

No output signal has occurred.

##### Task:

Initialises the analogue outputs in the selected range and polarity.

##### Calling convention:

###### ANSI C :

```

BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
WORD      w_ChannelNumber[4];
BYTE      b_VoltageMode[4];
BYTE      b_Polarity[4];

w_ChannelNumber[0] = 0;
w_ChannelNumber [1] = 2;
w_ChannelNumber [2] = 5;
w_ChannelNumber [3] = 8;
b_ VoltageMode [0] = ADDIDATA_RANGE_0;
b_ VoltageMode [1] = ADDIDATA_RANGE_0;
b_ VoltageMode [2] = ADDIDATA_RANGE_1;
b_ VoltageMode [3] = ADDIDATA_RANGE_1;
b_Polarity[0] = ADDIDATA_UNIPOLAR;
b_Polarity[1] = ADDIDATA_BIPOLAR;
b_Polarity[2] = ADDIDATA_UNIPOLAR;
b_Polarity[3] = ADDIDATA_BIPOLAR;
```

```

b_ReturnValue = b_ADDIDATA_InitMoreAnalogOutputs (dw_DriverHandle,
4,
w_ChannelNumber,
b_VoltageMode,
b_Polarity);
```

##### Return value:

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

### 5) b\_ADDIDATA\_Write1AnalogOutput (...)

### Syntax:

<Return Value> = b_ADDIDATA_Write1AnalogOutput	
	(DWORD dw_DriverHandle,
	WORD w_ChannelNumber,
	DWORD dw_ValueToWrite)

### Parameters:

**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_ChannelNumber</i>	Number of the analogue output to be written. The first analogue output begins from 0.
DWORD	<i>dw_ValueToWrite</i>	Value to be written on the selected analogue output.

**- Output:**

No output signal has occurred.

### Task:

Writes a value on 1 analogue output.

### Calling convention:

## ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Write1AnalogOutput
                (dw_DriverHandle,
                 0,
                 1000);
```

### Return value:

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

**6) b\_ADDIDATA\_WriteMoreAnalogOutputs (...)****Syntax:**

<Return Value> = b\_ADDIDATA\_WriteMoreAnalogOutputs  
 (DWORD dw\_DriverHandle,  
 WORD w\_NumberOfChannels,  
 PWORD pw\_ChannelNumber  
 PDWORD pdw\_ValueToWrite)

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

DWORD <i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD <i>w_NumberOfChannels</i>	Number of analogue outputs to be written
PWORD <i>pw_ChannelNumber</i>	Analogue output array. The first analogue output begins from 0.
PDWORD <i>pdw_ValueToWrite</i>	Array of the values to be written.

**- Output:**

No output signal has occurred.

**Task:**

Write a value on each of the selected channels.

**Calling convention:**ANSI C:

```

BYTE      b_ReturnValue;
DWORD     dw_DriverHandle;
WORD      w_ChannelNumber[4];
DWORD     b_ValueToWrite[4];

```

```

w_ChannelNumber[0] = 0;
w_ChannelNumber [1] = 2;
w_ChannelNumber [2] = 5;
w_ChannelNumber [3] = 8;
b_ValueToWrite [0] = 500;
b_ValueToWrite [1] = 1000;
b_ValueToWrite [2] = 1500;
b_ValueToWrite [3] = 2000;

```

```

b_ReturnValue = b_ADDIDATA_WriteMoreAnalogOutputs
(dw_DriverHandle,
4,
w_ChannelNumber,
b_ValueToWrite);

```

**Return value:**

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

**7) b\_ADDIDATA\_Release1AnalogOutput (...)****Syntax:**

<Return Value> = b\_ADDIDATA\_Release1AnalogOutput  
(DWORD dw\_DriverHandle,  
WORD w\_ChannelNumber)

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

DWORD <i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD <i>w_ChannelNumber</i>	Number of the analogue output number to be released. The first analogue output begins from 0.

**- Output:**

No output signal has occurred.

**Task:**

Releases the analogue output channel.

**Calling convention:**ANSI C:

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;

```
b_ReturnValue = b_ADDIDATA_Release1AnalogOutput  
                (dw_DriverHandle,  
                 0);
```

**Return value:**

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

## 8) b\_ADDIDATA\_ReleaseMoreAnalogOutputs (...)

### Syntax:

<Return Value> = b\_ADDIDATA\_ReleaseMoreAnalogOutputs  
(DWORD dw\_DriverHandle,  
WORD w\_NumberOfChannels,  
PWORD pw\_ChannelNumber)

### Parameters:

**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_NumberOfChannels</i>	Number of analogue outputs to be released.
PWORD	<i>pw_ChannelNumber</i>	Analogue output array. The first analogue output begins from 0.

**- Output:**

No output signal has occurred.

### Task:

Releases several analogue output channels.

### Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;
WORD	w_ChannelNumber[4];

```
w_ChannelNumber [0] = 0;
w_ChannelNumber [1] = 2;
w_ChannelNumber [2] = 5;
w_ChannelNumber [3] = 8;
```

```
b_ReturnValue = b_ADDIDATA_ReleaseMoreAnalogOutputs
                (dw_DriverHandle,
                 4,
                 w_ChannelNumber);
```

**Return value:**

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

**9) b\_ADDIDATA\_EnableDisable1AnalogOutputSync (...)****Syntax:**

<Return Value> = b\_ADDIDATA\_EnableDisable1AnalogOutputSync  
 (DWORD dw\_DriverHandle,  
 WORD w\_ChannelNumber,  
 DWORD dw\_EnableDisableSync)

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

DWORD <i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD <i>w_ChannelNumber</i>	Number of the analogue output to be synchronised.
	The first analogue output begins from 0.
DWORD <i>dw_EnableDisableSync</i>	If = 1, the analogue output is synchronised and written with all other analogue outputs.
	See function: b_ADDIDATA_TriggerAnalogOutput()

**- Output:**

No output signal has occurred.

**Task:**

Synchronises the analogue output to be written with all other synchronised analogue outputs.

**Calling convention:**ANSI C:

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

```
b_ReturnValue = b_ADDIDATA_EnableDisable1AnalogOutputSync
(dw_DriverHandle,
0,
1);
```

**Return value:**

1: No error

0: Error by calling up the function. Use the function "i\_ADDIDATA\_GetLastError", to find the error source.

## 10) b\_ADDIDATA\_EnableDisableMoreAnalogOutputsSync (...)

### Syntax:

<Return Value> = b\_ADDIDATA\_EnableDisableMoreAnalogOutputsSync  
(DWORD dw\_DriverHandle,  
WORD w\_NumberOfChannels,  
PWORD pw\_ChannelNumber,  
PDWORD pdw\_EnableDisableSync)

### Parameters:

**- Input:**

DWORD	<i>dw_DriverHandle</i>	Handle of the ADDI-DATA driver
WORD	<i>w_NumberOfChannels</i>	Number of analogue outputs to be released.
PWORD	<i>pw_ChannelNumber</i>	Analogue output array.
		The first analogue output begins from 0.
PDWORD	<i>pdw_EnableDisableSync</i>	If = 1 the analogue output is synchronised and written with all other analogue outputs.
		See function: b_ADDIDATA_TriggerAnalogOutput()

**- Output:**

No output signal has occurred.

### Task:

Synchronises the selected analogue outputs (pw\_ChannelNumber).

### Calling convention:

ANSI C :

BYTE	b_ReturnValue;
DWORD	dw_DriverHandle;
WORD	w_ChannelNumber[4];
DWORD	dw_EnableDisableSync [4];

```
w_ChannelNumber[0] = 0;
w_ChannelNumber[1] = 2;
w_ChannelNumber[2] = 5;
w_ChannelNumber[3] = 8;
b_EnableDisableSync [0] = 0;
b_EnableDisableSync [1] = 1;
b_EnableDisableSync [2] = 0;
b_EnableDisableSync [3] = 1;
b_ReturnValue = b_ADDIDATA_EnableDisableMoreAnalogOutputsSync
(dw_DriverHandle,
4,
w_ChannelNumber,
dw_EnableDisableSync);
```

**Return value:**

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



**11) b\_ADDIDATA\_TriggerAnalogOutput (...)****Syntax:**

<Return Value> = b\_ADDIDATA\_TriggerAnalogOutput  
(DWORD dw\_DriverHandle)

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

DWORD *dw\_DriverHandle*                      Handle of the ADDI-DATA driver

**- Output:**

No output signal has occurred.

**Task:**

Triggers the writing on all synchronised analog output channels.

**Calling convention:**ANSI C:

BYTE                      b\_ReturnValue;  
DWORD                    dw\_DriverHandle;

b\_ReturnValue = b\_ADDIDATA\_TriggerAnalogOutput  
(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.