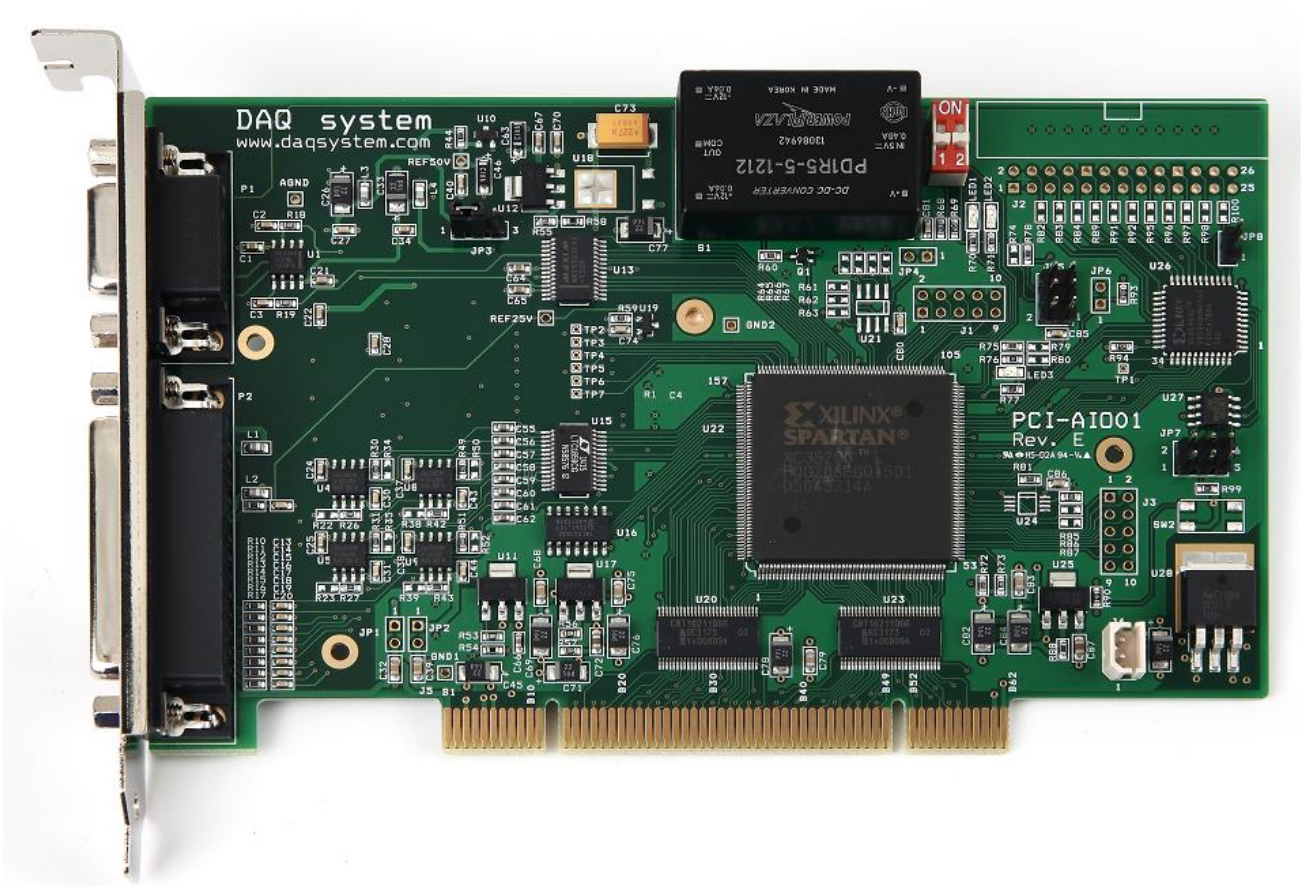
PCI-AIO01/02/04

API Manual

Version 1.0



DAQ SYSTEM
Measurement & Automation

Contents

Board Level API Functions

ADC(Analog to Digital Convertor) API Functions

DAC(Digital to Analog Convertor) API Functions

Describes the API (Application Programming Interface) for using the PCI-AIO series board. This API supports **PCI_AIO01, PCI_02, PCI_AIO04** products.

Board Level API Functions

Overview

BOOL **OpenDAQDevice (int nModel, int nBoard)**
BOOL **ResetBoard (int nModel, int nBoard)**
BOOL **CloseDAQDevice (int nModel, int nBoard)**
BOOL **GetBoardVersion (int nModel, int nBoard, int *version)**

OpenDAQDevice

This function opens the device. In the program using the PCI-AIO series board, the device must be opened by calling the function once at the beginning.

BOOL **OpenDAQDevice (int nModel, int nBoard)**

Parameters:

nModel : Write down the PCI-AIO model number.
nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.
If the function call succeeds, "TRUE" is returned.

ResetBoard

This function initializes the device currently installed in the system (PC).

BOOL **ResetBoard (int nModel, int nBoard)**

Parameters:

nModel : Write down the PCI-AIO model number.
nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.
If the function call succeeds, "TRUE" is returned.

CloseDAQDevice

This function closes all open PCI-AIO series devices. When the use of the device is finished, be sure to close the device so that other programs can use it.

BOOL CloseDAQDevice (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

GetBoardVersion

This function gets the hardware version information of the PCI-AIO device..

BOOL GetBoardVersion (int nModel, int nBoard, int *version)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

*version : It is a pointer of variable to get from version. It is a positive integer in the normal state.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC(Analog to Digital Convertor) API Functions

Overview

BOOL ADC_Init (int nModel, int nBoard)

BOOL ADC_Reset (int nModel, int nBoard)

BOOL ADC_Close (int nModel, int nBoard)

BOOL ADC_GetData (int nModel, int nBoard, int nChannel, int *nData)

BOOL ADC_SetRange (int nModel, int nBoard, int nChannel, int nRange)

BOOL ADC_ConfigChannel (int nModel, int nBoard, int nChannel, DWORD dwConfig)

BOOL ADC_ConfigChannelEx (int nModel, int nBoard, int nChannel, int nPol, int nRange)

BOOL ADC_ConfigResolution (int nModel, int nBoard, DWORD dwResol)

BOOL ADC_StartBufferedRead (int nModel, int nBoard)

BOOL ADC_StopBufferedRead (int nModel, int nBoard)

int ADC_GetBufferedData (int nModel,int nBoard,int nChannel,int nCount, int *nData)

int ADC_GetBufferedDataEx (int nModel,int nBoard,int nChannel,int nCount, float *fData)

BOOL ADC_SetAvgCounter (int nModel, int nBoard,int nChannel,int nCount)

ADC_Init

This function initializes ADC setup.

BOOL ADC_Init (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_Reset

This function resets the ADC function.

BOOL ADC_Reset (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_Close

This function is called when the ADC function is no longer used to release the used resource.

BOOL ADC_Close (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_GetData

This function reads one data value for the current ADC input.

Since the data to be read is affected by precision, input polarity and range, functions ADC_ConfigResolution(), ㅁADC_ConfigChannelEx() must be called.

The range of ADC values according to the precision is as follows [Table 3] according to the polarity setting.

[Table 1. Range of AD values according to precision]

Precision	Polarity	
	Unipolar	Bipolar
12-Bit	0 ~ +4095	-2048 ~ +2047
14-Bit	0 ~ +16383	-8192 ~ +8191
16-Bit	0 ~ +65535	-32768 ~ +32767

In this case, the calculation of the actual voltage is as follows.

Voltage = (Read/Resolution Range) *(Range Max-Min)

Ex) 0 to 5V range, when set to 12-bit

When you read 1024, the actual voltage is (1024 /4096) * (5-0) = 1.25 [V].

-10 to + 10V range, if set to 16-bit

When I read -16384 the actual voltage is (-16384 /65536) * (10-(-10)) = -5 [V].

BOOL ADC_GetData (int nModel, int nBoard, int nChannel, int *nData)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the ADC channel value.
The channel number of PCI-AIO01 is from 0 to 7.

*nData : It is a pointer to a variable to receive the ADC value to be read.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_SetRange

This function specifies the analog input range of the ADC.

BOOL ADC_SetRange (int nModel, int nBoard, int nChannel, int nRange)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : It is meaningless and sets '0'. In this function, the input characteristics of all channels are applied equally, so use the ADC_ConfigChannelEx() function to set each channel.

nRange : Set the ADC input range. The default is 0 to 10V.
0 : 0 ~ 5V, 1 : -5 ~ 5V
2 : 0 ~ 10V, 3 : -10 ~ 10V

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_ConfigChannel

This function sets the analog input polarity of the ADC channel.

BOOL ADC_ConfigChannel (int nModel, int nBoard, int nChannel, DWORD dwConfig)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : It is meaningless and sets '0'. In this function, the input characteristics of all channels are applied equally, so use the ADC_ConfigChannelEx() function to set each channel.

dwConfig : Analog input polarity
0 : Single-Ended(SE)
1 : Differential(DI)

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_ConfigChannelEx

This function sets the ADC input polarity and input range for each channel.

BOOL ADC_ConfigChannelEx (int nModel, int nBoard, int nChannel, int nPol, int nRange)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Specifies the applied channel value.

nPol : Analog input polarity.
0 : Single-Ended(SE), 1 : Differential(DI)

nRange : Set the ADC input range. The default is 0 to 10V.
0 : 0 ~ 5V, 1 : -5 ~ 5V
2 : 0 ~ 10V, 3 : -10 ~ 10V

Return Value:

If the function call fails, "FALSE" is returned.
If the function call succeeds, "TRUE" is returned.

ADC_ConfigResolution

This function sets the AD data width (in bits)
It specifies the precision of read data separately from that the board's AD converter supports 12, 14, and 16-bit three resolutions.

BOOL ADC_ConfigResolution (int nModel, int nBoard, DWORD dwResol)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

dwResol : Data bit set value. Set according to the supported AD converter data bit. 0 (default) for 12-bit ADC, 1 for 14-bit ADC, and 2 for 16-bit ADC.

Return Value :

If the function call fails, "FALSE" is returned.
If the function call succeeds, "TRUE" is returned.

ADC_StartBufferedRead

This function starts saving AD data to the buffer.

The library allocates 32,768 data buffers per channel. When this function is called, AD data is continuously stored in the buffer in a ring buffer format. The user must read the data before it is overwritten to obtain valid data.

BOOL ADC_StartBufferedRead (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_StopBufferedRead

This function stops saving AD data to the buffer.

BOOL ADC_StopBufferedRead (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

ADC_GetBufferedData

This function reads AD data stored in the buffer.

Like ADC_GetData(), the precision, input polarity and range are set using the functions ADC_ConfigResolution() and ADC_ConfigChannelEx().

The range of ADC values according to the precision is as shown in Table 3 according to the polarity setting.

In this case, the actual voltage is calculated as follows.

Voltage = (Read/Resolution Range) *(Range Max-Min)

Ex) 0 to 5V range, when set to 12-bit

When you read 1024, the actual voltage is (1024 /4096) * (5-0) = 1.25 [V].

-10 to + 10V range, if set to 16-bit

When I read -16384 the actual voltage is (-16384 /65536) * (10-(-10)) = -5 [V].

int ADC_GetBufferedData (int nModel,int nBoard,int nChannel,int nCount, int *nData)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the DAC channel value.
The channel number of PCI-AIO01 is from 0 to 1.

nCount : Specifies the number of data to be read. Specify a number within the maximum number of buffers (32,768).

*nData : It is a pointer to a variable where AD data is to be stored. Allocate the buffer size larger than the number of data to be read.

Return Value:

It is the number of data actually stored in nData.

ADC_GetBufferedDataEx

This function reads AD data stored in the buffer.

Acquires data stored in buffer in voltage level float format. Since the data value obtained is affected by precision, input polarity, and input range, ADC_ConfigResolution() and ADC_ConfigChannelEx() functions must be called.

int ADC_GetBufferedDataEx (int nModel,int nBoard,int nChannel,int nCount, float *fData)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the DAC channel value.
The channel number of PCI-AIO01 is from 0 to 1.

nCount : Specifies the number of data to be read. Specify a number within the maximum number of buffers (32,768).

*fData : It is a pointer to a variable in which voltage level data is to be stored. Allocate the buffer size larger than the number of data to be read.

Return Value:

It is the number of data actually stored in nData.

ADC_SetAvgCounter

This function sets the number of AD data to be used for the moving average.

When the number of data is 1, the moving average is not applied, and the ADC_GetBufferedData() and ADC_GetBufferedDataEx() functions acquire averaged data.

BOOL ADC_SetAvgCounter (int nModel, int nBoard,int nChannel,int nCount)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the DAC channel value.
The channel number of PCI-AIO01 is from 0 to 1.

nCount : Specifies the number of data to which the moving average is applied.
Specify a number from 1 to 255.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

DAC(Digital to Analog Convertor) API Functions

Overview

BOOL DAC_Init (int nModel, int nBoard)

BOOL DAC_Reset (int nModel, int nBoard)

BOOL DAC_Close (int nModel, int nBoard)

BOOL DAC_SetOutput (int nModel, int nBoard, int nChannel, DWORD dwData)

BOOL DAC_ConfigChannel (int nModel, int nBoard, int nChannel, DWORD dwConfig)

DAC_Init

This function initializes the settings of the DAC.

BOOL DAC_Init (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

DAC_Reset

This function resets the function of the DAC.

BOOL DAC_Reset (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

DAC_Close

This function is called when the DAC function is no longer used to release the used resource.

BOOL DAC_Close (int nModel, int nBoard)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

DAC_SetOutput

This function outputs the DAC value. The output range of the returned DAC is 0 to 10V for unipolar and -10 to 10V for bipolar.

At this time, the range of the set value is 0 to 4095, and the calculation is as follows.

Ex) When it is set in the range of 0 to 10V

When outputting 5V, the set value is (5V/10V) * 4096 = 2048.

For -10 to +10V

When outputting 5V, the set value is (10+5V/20V) * 4096 = 3072.

When outputting -5V, the set value is (10-5V/20V) * 4096 = 1024..

BOOL DAC_SetOutput (int nModel, int nBoard, int nChannel, DWORD dwData)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the DAC channel value.
The channel number of PCI-AIO01 is from 0 to 1.

dwData : Set the value to be output to the DAC.

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

DAC_ConfigChannel

This function sets the output range for each ADC channel.

BOOL DAC_ConfigChannel (int nModel, int nBoard, int nChannel, DWORD dwConfig)

Parameters:

nModel : Write down the PCI-AIO model number.

nBoard : Shows the board number currently installed in the system.
The board number is set using the DIP switch of the board.

nChannel : Write down the DAC channel value.
The channel number of PCI-AIO01 is from 0 to 1.

dwConfig : 0 : Bipolar (-10V ~ + 10V)
1 : Unipolar (0V ~ + 10V)

Return Value:

If the function call fails, "FALSE" is returned.

If the function call succeeds, "TRUE" is returned.

Memo

Contact Point

Web sit : https://www.daqsystem.com

Email : postmaster@daqsystem.com

DAQ SYSTEM
Measurement & Automation