USB-MULTI API Programming (Rev 1.0)

API (Application Programming Interface)

Board Level API Functions

Overview

BOOL **OpenDAQDevice (int nBoard)**

BOOL **ResetSystem (int nBoard)**

BOOL **CloseDAQDevice (int nBoard)**

OpenDAQDevice

It opens a device. You may call this function at the very first time you run the program and some suspicious operation.

BOOL **OpenDAQDevice (int nBoard)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

Return Value:

If the device open succeeds, it returns “TRUE”.

If the device open fails, it returns “FALSE”.

ResetSystem

Reset USB-MULTI device. (It is not usually used, but, it calls to reset in case of bug or strange motion.)

BOOL **ResetSystem(int nBoard)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

Return Value:

If the function fail to reset, it returns “FALSE”.

If the function succeed to reset, it returns “TRUE”.

CloseDAQDevice

It closes USB-MULTI series device opened. If use of device is finished, it can certainly close a device for making it other programs so as usable.

BOOL **CloseDAQDevice (int nBoard)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

Return Value:

If the close function fails, it returns “FALSE”.

If the close function succeeds, it returns “TRUE”.

Digital Input/ Output API Functions

Overview

BOOL **DIO_Set_Direction (int nBoard, int nCnt, BYTE *byBuf)**

BOOL **DIO_Set_Data (int nBoard, int nCnt, BYTE *byBuf)**

BOOL **DIO_Get_Data (int nBoard, int nCnt, BYTE *byBuf)**

DIO_Set_Direction

It sets up a direction of 15 Digital I/O.

If each bit set up '1', it is an output. If each bit set up '0', it is an input.

BOOL **DIO_Set_Direction (int nBoard, int nCnt, BYTE *byBuf)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up "0" because of supporting just one Current USB-MULTI board

nCnt : It displays how do the byte transfer. It will transfer 2 bytes because of USB-MULTI has 15bits. So, it sets up "2".

*byBuf : The data buffer shall be prepared as much as a number of byte to set up. For example, if bit 14 and bit 15 only become output and the others become input, it sets 0x4020 to 16 bits value. So, the byBuf[0] is 0x20. The byBuf[1] is 0x40.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

DIO_Set_Data

It sets up an output value of 15 Digital I/O.

If each bit set up '1', an output is "High". If each bit set up '0', an output is "Low".

BOOL **DIO_Set_Data (int nBoard, int nCnt, BYTE *byBuf)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up "0" because of supporting just one Current USB-MULTI board

nCnt : It displays how do the byte transfer. It will transfer 2 bytes because of USB-MULTI has 15bits. So, it sets up "2".

*byBuf : The data buffer shall be prepared as much as a number of byte to set up. For example, if bit 14 and bit 15 only become output and the others become input, it sets 0x4020 to 16 bits value. So, the byBuf[0] is 0x20. The byBuf[1] is 0x40.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

DIO_Get_Data

It reads an input value of 15 Digital I/O.

If each bit set up '1', an output is "High". If each bit set up '0', an output is "Low".

BOOL **DIO_Get_Data (int nBoard, int nCnt, BYTE *byBuf)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up "0" because of supporting just one Current USB-MULTI board

nCnt : It displays how do the byte transfer. It will transfer 2 bytes because of USB-MULTI has 15bits. So, it sets up "2".

*byBuf : The data buffer shall be prepared as much as a number of byte to read.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Misc API Functions

Overview

BOOL **MISC_Set_SSR (int nBoard, BOOL bAction)**

BOOL **MISC_Get_SSR (int nBoard, BOOL *bState)**

BOOL **MISC_Get_Photo (int nBoard, BYTE *byState)**

BOOL **MISC_Get_Count (int nBoard, WORD *wValue)**

BOOL **MISC_Reset_Count (int nBoard)**

BOOL **MISC_Set_Timer (int nBoard, WORD wValue)**

BOOL **MISC_Reset_Timer (int nBoard)**

MISC_Set_SSR

SSR(Solid State Relay) ON/OFF.

BOOL **MISC_Set_SSR (int nBoard, BOOL bAction)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

bAction : If it is “TRUE” state, it’s “ON”. If it is “FALSE” state, it is “OFF”.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Get_SSR

It reads a start of SSR(Solid State Relay).

BOOL **MISC_Get_SSR (int nBoard, BOOL *bState)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

*bState : If it is “TRUE” state, the relay state is “ON”.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Get_Photo

It reads a state of Photo-coupler isolated DIN.

BOOL **MISC_Get_Photo (int nBoard, BYTE *byState)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

*byState : It displays a state value of DIN0,1.

byState bit0: DIO0 (0 : OFF, 1: ON)

byState bit1: DIO1 (0 : OFF, 1: ON)

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Get_Count

현재의 counter 값을 읽어 온다.

BOOL **MISC_Get_Count (int nBoard, WORD *wValue)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

*wValue : The buffer that it read from Counter value

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Reset_Count

It resets a value of current counter (initialization to “0”).

BOOL **MISC_Reset_Count (int nBoard)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Set_Timer

It sets up current Timer value.

BOOL **MISC_Set_Timer (int nBoard, WORD wValue)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

wValue : The timer value that it’ll set up. A width of output pulse is 50nSEC, period setup is changed by timer register value. If it is 0xFFFF, the minimum period is 500nSEC (0.5usec). If it is 0x0000, the maximum period is 32.768mSEC (0.5 x 65536 = 32768uSEC).

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

MISC_Reset_Timer

It resets current Timer value, it make it so as not to let output pulse.

BOOL **MISC_Reset_Timer (int nBoard)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

DAC API Functions

Overview

BOOL **DAC_Set (int nBoard, BYTE byCh, WORD wValue)**

DAC_Set

Each channel output value of DAC set up.

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.

(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

byCh : DAC Channel Number, 0/1/2.

wValue : DAC Value(Refer to USB-MULTI Manual chapter 5.2)

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

ADC API Functions

Overview

BOOL **ADC_Get (int nBoard, BYTE byCh, WORD *wValue)**

BOOL **ADC_Get_All (int nBoard, BYTE byCh, WORD *wValue)**

ADC_Get

It reads a value of each channel of ADC.

BOOL **ADC_Get (int nBoard, BYTE byCh, WORD *wValue)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.
(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

byCh : ADC Channel Number, from 0 to 7.

*wValue : ADC Value (Refer to USB-MULTI Manual chapter 5.2)

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

ADC_Get_All

It reads a value of each channel of ADC.

BOOL **ADC_Get_All (int nBoard, BYTE byCh, WORD *wValue)**

Parameters:

nBoard : The PC can use several USB-MULTI board, it can select one of them.
(notice) USB-MULTI Model Number always sets up “0” because of supporting just one Current USB-MULTI board

byCh : ADC Channel Number, from 0 to 7.

*wValue : Buffer point which it will read, it shall prepare a buffer of minimum 8WORD(16byte). (Refer to USB-MULTI Manual chapter 5.2)

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.