

top16.dll API

Version: 2.0
10-September-2010

The following files should accompany this document:

FTD2XX.dll DLL from FTDI chip
top16.dll Functions to interface the Top16 Module, requires the FTD2XX.dll

The following functions are provided by top16.dll to interface the Top16 USB Module.

<u>int32_t ListTop16Boards(char stringArray[][20]).....</u>	<u>1</u>
<u>int32_t SetOutputs(uint32_t Handle, uint8_t mask, uint8_t set).....</u>	<u>2</u>
<u>int32_t SetOutputsB(uint32_t Handle, uint8_t mask, uint8_t set).....</u>	<u>2</u>
<u>int32_t GetInputs(uint32_t Handle).....</u>	<u>3</u>
<u>int32_t readAnalogInput(uint32_t Handle, int16_t input, uint8_t gain).....</u>	<u>4</u>
<u>int32_t SetPWM(uint32_t Handle, uint8_t pwm, uint8_t output).....</u>	<u>5</u>
<u>int32_t ReadPulseCount(uint32_t Handle, int16_t input).....</u>	<u>5</u>
<u>uint32_t OpenBoard(char* board).....</u>	<u>6</u>
<u>int32_t CloseBoard(uint32_t Handle).....</u>	<u>6</u>
<u>int32_t dllversion().....</u>	<u>6</u>

int32_t ListTop16Boards(char stringArray[][20])

Search for attached TCTEC Top16 boards.
The stringArray will be filled with the names of the boards.
It must point to a 2 dimensional char (8bit) array of dimensions [n][20]
where n is the number of boards expected.

Returns:

The number of top16 boards found or 0 if none found.

int32_t SetOutputs(uint32_t Handle, uint8_t mask, uint8_t set)

Set the digital output states.

8 bits of the 'mask' determines which outputs will be affected:

- 1 = output will be affected
- 0 = leave alone (don't change)

'set' determines whether the outputs in mask will be switched on or switched off.

- 1 = output on (pulled low)
- 0 = output off (not pulled low).

The value of set can be either 1 or 0, therefore the outputs specified in mask can all either be switched on or off.

When controlling relays, 'Break before Make' should be used, so when changing the state of the outputs, any outputs to be switched off should be applied first, followed by a delay, then any outputs to be switched on should be switched on, this will avoid transient conditions, due to the relays taking some time (milli Seconds) to switch.

See setOutputsB to specify a new output state consisting of a combination of states (ON and OFF).

Returns:

The state of the outputs after the action.

0 to FF:

- 1 = output is on (pulled low).
- 0 = output is off (not pulled low)

A negative number indicates an error:

- 1 invalid board handle (board not found)
- 2 DLL function not loaded (FTD2XX.DLL not found)
- 3 Invalid bit mask (must be 0 to FF)
- 4 Undefined error

int32_t SetOutputsB(uint32_t Handle, uint8_t mask, uint8_t set)

Set the digital output states.

8 bits of the 'mask' determines which outputs will be affected:

- 1 = output will be affected
- 0 = leave alone (don't change)

8 bits of 'set' determine whether the corresponding output will be switched on or switched off.

- 1 = output on (pulled low)
- 0 = output off (not pulled low).

When controlling relays, 'Break before Make' should be used, so when changing the state of the outputs, any outputs to be switched off should be applied first, followed by a delay, then any outputs to be switched on should be switched on, this will avoid transient conditions, due to the relays taking some time (milliseconds) to switch.

Returns:

The state of the outputs after the action.

0 to FF:

- 1 = output is on (pulled low).
- 0 = output is off (not pulled low)

A negative number indicates an error:

- 1 invalid board handle (board not found)
- 2 DLL function not loaded (FTD2XX.DLL not found)
- 3 Invalid bit mask (must be 0 to FF)
- 4 Undefined error

int32_t GetInputs(uint32_t Handle)

Read the digital inputs.

Returns:

- The state of the digital inputs
- 0 to FF:
- 1 = input is high
- 0 = input is low

A negative number indicates an error:

- 1 invalid board handle (board not found)
- 2 DLL function not loaded (FTD2XX.DLL not found)
- 4 Undefined error

int32_t readAnalogInput(uint32_t Handle, int16_t input, uint8_t gain)

Read an analog input.

The internal analog amplifier is set to the gain specified and an input is sampled.

Handle: A handle to an open board (returned by OpenBoard)

input: The input to read (1 to 8)

gain: The gain setting

Gain setting	Gain
'Z'	1
'Y'	2
'X'	4
'W'	8
'V'	16
'U'	32
'T'	64

Returns:

The raw analog input value (12 bit).

A negative number indicates an error:

- 1 invalid board handle
- 2 DLL function not loaded (FTDXX.DLL not found)
- 3 invalid gain
- 4 undefined error

int32_t SetPWM(uint32_t Handle, uint8_t pwm, uint8_t output)

Set a PWM value on an output.

pwm: (0x00 to 0xFF) pwm width

output: the output to set (1 to 8)

Returns:

- 0 success
- 1 invalid pwm value
- 2 invalid output
- 4 undefined error

int32_t ReadPulseCount(uint32_t Handle, int16_t input)

Read the pulses (negative edges) on a particular input.

Handle: A handle to an open board (returned by OpenBoard)

input: The input to read (1 to 8)

Returns:

The pulse count 0 to 32767 (rolls over after 32767)

uint32_t OpenBoard(char* board)

Initialize a top16 board, returns a Handle (uint32) to opened board if successful.
Must be called first before set and get commands.

Returns:

>0 Handle of opened board.

int32_t CloseBoard(uint32_t Handle)

Close an open board.

The board must be closed when application has finished communicating with it.

Closing a board releases a handle and frees up memory it does not affect the actual hardware or its output states.

Handle Handle of open board.

Returns:

0 success

int32_t dllversion()

Returns the version of this DLL.

Returns:

major and minor version number.

i.e

11 = version 1.1

24 = version 2.4