How to use the BrainVision TriggerBox application interface

Once the TriggerBox software is installed, a serial port is available for each application which is able to handle serial port communications. The COM port number can be found in the “Ports (COM & LPT)” section of the device manager. Search the entry “TriggerBox VirtualSerial Port (COMx)” and take the COM port number.

From your application open the serial port. The serial port settings have no influence on the transmission speed of the interface because it’s a pure virtual port and data is always transmitted as fast as possible. So you can either leave the default port settings or, if your application requires those parameters, you can use dummy values like:

- Bits per second: 9600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None

Each byte written to the COM port is transmitted to the eight output lines (Bit 0 – Bit 7) of the TriggerBox “Output (to Amp)” connector.

Reading one byte from the serial port returns the state of the TriggerBox “Input (8-15)” lines.

If you are finished, the output line should be reset to their default levels by writing a 0xFF to the serial port and the port must be closed.

Examples

Below are examples for the TriggerBox usage in

- C++
- Python
- C#
- MATLAB®
- MATLAB® + Psychophysics Toobox

Please note: For the MATLAB® example you need at least the TriggerBox driver version 1.1.1.

Version 001 02/2017

Brain Products GmbH | Zeppelinstrasse 7 | 82205 Gilching, Germany

C++ Example

```
#include "stdafx.h"
#include <windows.h>
#include <iostream>

int _tmain(int argc, _TCHAR* argv[])
{
   BYTE data;
   DWORD bytes;

   // Open the Windows device manager,
   // search for the "TriggerBox VirtualSerial Port (COMx)"
   // in "Ports (COM & LPT)" and enter the COM port number as file name.
   // For the usage of serial ports larger than COM9
   // see https://support.microsoft.com/en-us/kb/115831
   HANDLE hPort = CreateFile(_T("\\\\.\\COM6"),
      GENERIC_WRITE | GENERIC_READ, 0, NULL, OPEN_EXISTING, 0, NULL);

   // Set the port to an initial state
   data = 0x00;
   WriteFile(hPort, &data, 1, &bytes, NULL);

   // Set Bit 0, Pin 2 of the Output(to Amp) connector
   data = 0x01;
   WriteFile(hPort, &data, 1, &bytes, NULL);

   // Read 8 bits from the Input(8-15) connector
   ReadFile(hPort, &data, 1, &bytes, NULL);
   std::cout << (int)data << std::endl;

   // Reset Bit 0, Pin 2 of the Output(to Amp) connector
   data = 0x00;
   WriteFile(hPort, &data, 1, &bytes, NULL);

   // Reset the port to its default state
   data = 0xFF;
   WriteFile(hPort, &data, 1, &bytes, NULL);

   // Close the serial port
   CloseHandle(hPort);

   return 0;
}
```

Python Example

```
import serial

# Open the Windows device manager,
# search for the "TriggerBox VirtualSerial Port (COM6)"
# in "Ports /COM & LPT)" and enter the COM port number in the constructor.
port = serial.Serial("COM6")

# Set the port to an initial state
port.write([0x00])

# Set Bit 0, Pin 2 of the Output(to Amp) connector
port.write([0x01])

# Read 8 bits from the Input(8-15) connector
print "0x%X"%ord(port.read(1))

# Reset Bit 0, Pin 2 of the Output(to Amp) connector
port.write([0x00])

# Reset the port to its default state
port.write([0xFF])


# Close the serial port
port.close()
```

C# Example

```
using System;
using System.Collections.Generic;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TriggerBox
{
    class Program
    {
        static void Main(string[] args)
        {
            Byte[] data = { (Byte)0 };


            // Open the Windows device manager,
            // search for the "TriggerBox VirtualSerial Port (COMx)"
            // in "Ports /COM & LPT)" and enter the COM port number in the constructor.
            SerialPort TriggerBox = new SerialPort("COM6");
            TriggerBox.Open();

            // Set the port to an initial state
            data[0] = 0x00;
            TriggerBox.Write(data,0,1);

            // Set Bit 0, Pin 2 of the Output(to Amp) connector
            data[0] = 0x01;
            TriggerBox.Write(data, 0, 1);

            // Read 8 bits from the Input(8-15) connector
            Console.WriteLine(TriggerBox.ReadByte());

            // Reset Bit 0, Pin 2 of the Output(to Amp) connector
            data[0] = 0x00;
            TriggerBox.Write(data, 0, 1);

            // Reset the port to its default state
            data[0] = 0xFF;
            TriggerBox.Write(data, 0, 1);

            // Close the serial port
            TriggerBox.Close();
        }
    }
}
```

MATLAB® Example

```
%%% Sample code to write to and read from TriggerBox from Brain Products GmbH

%%% Important note: Make sure the toggle switches on the TriggerBox
%%% are set to 'PCx' (and NOT 'Inx'; x=BitNo) for all used bits
%%% i.e. if e.g. only bits 0 to 3 are used, switches 0 to 3 need to be set

%%% Further information: See Matlab Help "Getting Started with Serial I/O"

% !!! Important Note: When using virtual serial ports
% !!! remember to restart MATLAB, once you plug in the TriggerBox,
% !!! for the new port to show up in MATLAB.
% !!! This is to allow the underlying serial API to update the list
% !!! of available serial ports.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%         START of sample code         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% To construct the right serial port object,
% open the Windows device manager and search for the "TriggerBox VirtualSerial Port
(COMx)"
% in section "Ports (COM & LPT)" and adjust/enter the COM port number x in the following
% constructor
SerialPortObj=serial('COM3', 'TimeOut', 1); % in this example x=3

% To connect the serial port object with serial port hardware
fopen(SerialPortObj);

% Set the port to zero state 0
fwrite(SerialPortObj, 0,'sync');

% Set Bit 0 (Pin 2 of the Output(to Amp) connector)
fwrite(SerialPortObj, 1,'sync');

% To read data from the TriggerBox
% first toggle the RequestToSend flag
SerialPortObj.RequestToSend = 'off';
SerialPortObj.RequestToSend = 'on';
% then read at least one byte from the TriggerBox
fread(SerialPortObj, max([1 SerialPortObj.BytesAvailable]))

% Reset the port to zero state 0
fwrite(SerialPortObj, 0,'sync');

% Reset the port (i.e. bit 0 to 7) to its resting state 255
fwrite(SerialPortObj, 255,'sync');
```

Version 001 02/2017

Brain Products GmbH | Zeppelinstrasse 7 | 82205 Gilching, Germany

```
% Then disconnect/close the serial port object from the serial port
fclose(SerialPortObj);

% Remove the serial port object from memory
delete(SerialPortObj);

% Remove the serial port object from the MATLAB® workspace
clear SerialPortObj;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%          END of sample code          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


Brain Products GmbH | Zeppelinstrasse 7 | 82205 Gilching, Germany
Version 001 02/2017

MATLAB® Psychophysics Toolbox Example

```
%%% Sample code to write to and read from TriggerBox from Brain Products GmbH
%%% using the Psychophysics Toolbox functions

%%% Important note: Make sure the toggle switches on the TriggerBox
%%% are set to 'PCx' (and NOT 'Inx'; x=BitNo) for all used bits
%%% i.e. if e.g. only bits 0 to 3 are used, switches 0 to 3 need to be set

%%% Further information: See http://docs.psychtoolbox.org/IOPort

% !!! Important Note: When using virtual serial ports
% !!! remember to restart MATLAB, once you plug in the TriggerBox,
% !!! for the new port to show up in MATLAB.
% !!! This is to allow the underlying serial API to update the list
% !!! of available serial ports.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%         START of sample code         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% To construct and open the right serial port object,
% open the Windows device manager and search for the "TriggerBox VirtualSerial Port
(COMx)"
% in section "Ports (COM & LPT)" and adjust/enter the COM port number x in the following
% constructor
TB = IOPort('OpenSerialPort', 'COM6'); % in this example x=6

% Set the port to zero state 0
IOPort('Write', TB, uint8(0), 0);

% Set Bit 0 (Pin 2 of the Output(to Amp) connector)
IOPort('Write', TB, uint8(1), 0);

% Read data from the TriggerBox
IOPort('Read', TB, 0, 1)

% Reset the port to zero state 0
IOPort('Write', TB, uint8(0), 0);

% Reset the port (i.e. bit 0 to 7) to its resting state 255
IOPort('Write', TB, uint8(255), 0);

% Then disconnect/close the serial port object from the serial port
IOPort('Close', TB);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%          END of sample code          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


Brain Products GmbH | Zeppelinstrasse 7 | 82205 Gilching, Germany