

DODOSIM 206 FSX HARDWARE UTILITIES



USER MANUAL

(ENGLISH)

Document version: 03-05-10/1

This document and the flight simulation software described within are for home entertainment use only.

Copyright© 2010 DodoSim. All Rights Reserved.

DODOSIM 206 FSX HARDWARE UTILITIES

USER MANUAL

1 INTRODUCTION

The *DodoSim 206 FSX* is a high-fidelity helicopter “add-on” for Microsoft® Flight Simulator X® (FSX.) It can be purchased online from www.dodosim.com. For more information on the aircraft and the unique capabilities it brings to helicopter flight in FSX, please visit the site and download the manual.

When running, *the Dodosim 206 FSX* replaces much of the default helicopter behaviour within FSX and utilises its own internal custom variables to drive the on-screen aircraft gauges, such as pressure, temperatures, engine and rotor speeds. Because of this, third-party utilities that make use of Inter-Process Connect (IPC) facilities, (such as FSUIPC and SimConnect), will not be able to access the correct gauge values in order to display external software or hardware gauges.

The set of tools provided in *the DodoSim 206 FSX Hardware Utilities* makes these custom variables accessible to the outside world for the purpose of driving SimKits gauges, (or other gauges that utilise FSUIPC to access FSX's data), and hardware annunciator lighting panels.

Care must be taken to observe and understand the installation and set-up procedure to ensure the tools operate correctly.

1.1 CONTENTS

The following tools are provided in this package:

- 1) ***DodoSim206FSXExportIPC*** - This executable runs constantly whilst the DodoSim 206 FSX is loaded and provides all the analogue gauge variables through FSUIPC's data interface. It was written to allow SimKits USB gauges to access the correct gauge display values when using the SimKits FSUIPC software (USBFSUIPCLink.)
- 2) ***DodoSim206FSXExportIPC_Test*** - This executable is used only when initially configuring and calibrating the SimKits USB gauges.
- 3) ***DodoSim206FSXDataExport*** – This (Dynamic Link Library) DLL is provided to make the annunciator and turbine over-temp light states available to custom-written hardware drivers via dll function calls.

1.2 PREREQUISITES

These tools require the following software to be installed in order to function:

- Microsoft® Flight Simulator X® SP2, Acceleration or Gold
- *DodoSim 206 FSX* (build V1.05 or later.) Contact support@dodosim.com if your current version pre-dates this for the latest files. (Your purchase information may be requested.)

DodoSim206FSXExportIPC and *DodoSim206FSXExportIPC_Test* additionally require FSUIPC (version 4.416 or later) to be installed. It can be downloaded from <http://www.schiratti.com/dowson.html>. Follow FSUIPC's own installation and set-up instructions and ensure that FSUIPC has added a new menu entry “Addons->FSUIPC” in FSX before attempting to use these two utilities.

2 USING THE TOOLS

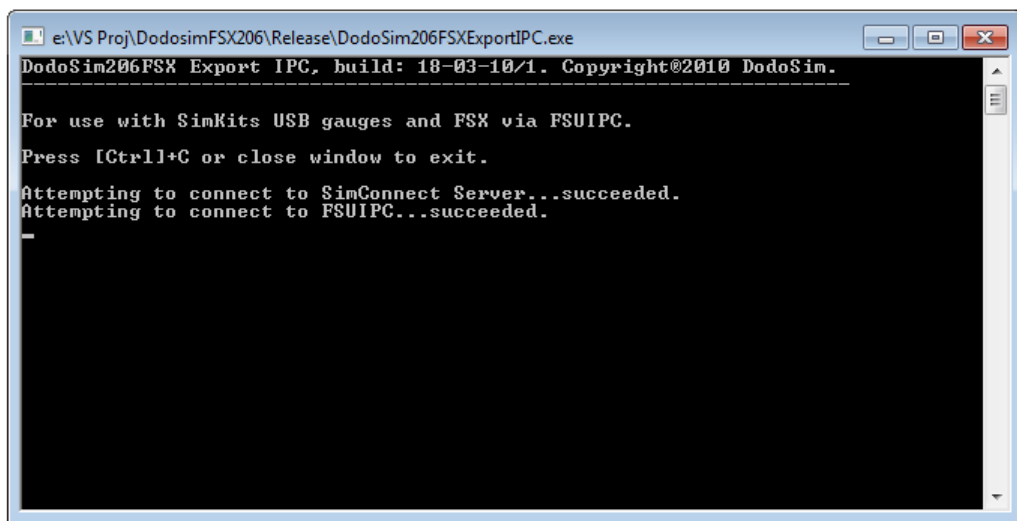
2.1 DODOSIM206FSXEXPORTIPC

2.1.1 Description

The *DodoSim206FSXExportIPC.exe* file is an executable that, when running, connects to the *DodoSim 206 FSX* helicopter via SimConnect (FSX's own inter-process communications methodology) and FSUIPC, providing a bridge between the custom variables in the *DodoSim 206 FSX* helicopter and FSUIPC, from which other software and hardware drivers can extract gauge display values.

2.1.2 Usage

Simply run the *DodoSim206FSXExportIPC.exe* file. It will open a text window that should look like this assuming FSX is running and FSUIPC is installed correctly:



When the sim is running the program will automatically fetch the gauge display values from FSX via SimConnect and populate FSUIPC's data tables to be read by other software and hardware drivers that utilise FSUIPC.

In addition to replacing FSX data values with custom DodoSim variables, the *DodoSim206FSXExportIPC.exe* also makes all the annunciator lights and Turbine Over-Temp (TOT) light states available via FSUIPC packed into a binary bit pattern available at Offset 0x2F28 (Concorde nose visor %.) By reading the 8 byte data from this offset, each light state can be extracted as an individual bit for driving lights via Phidgets interfaces or similar. The bits are arranged as follows:

BIT	Annunciator / Light
0	ENG_OUT
1	ROTOR_LOW
2	FUEL_LOW
3	GENERATOR_FAIL
4	BATTERY_WARM
5	BATTERY_HOT

6	FUEL_PUMP_FAIL
7	TRANSMISSION_PRESSURE_FAIL
8	TRANSMISSION_TEMPERATURE_FAIL
9	ENGINE_CHIP
10	TAIL_ROTOR_CHIP
11	TRANSMISSION_CHIP
12	FUEL_FILTER_FAIL
13	BAGGAGE_DOOR
14	SIMCONNECT_FAIL
15	SPARE_1 (below SC FAIL)
16	SPARE_2 (right of SC FAIL)
17	SPARE_3 (left of GEN FAIL)
18	SPARE_4 (right of FUEL LOW)
19	SPARE_5 (below GEN FAIL)
20	TURBINE_OVER_TEMP_LIGHT

2.2 DODOSIM206FSXEXPORTIPC_TEST

2.2.1 Description

The *DodoSim206FSXExportIPC_Test.exe* file is an executable that is used only during the initial testing and calibration of SimKits gauges in conjunction with the *DodoSim 206 FSX* helicopter. It runs instead of the *DodoSim 206 FSX* helicopter and replicates its gauge display values, which it can drive to minimum, middle and maximum values, so that hardware gauges can be calibrated easier.

2.2.2 Usage

- 1) Firstly, start FSX and load any aircraft **other than** the *DodoSim 206 FSX* helicopter. (This is because the *DodoSim206ExportIPC_Test* program creates the client data areas within FSX that otherwise the helicopter would do, and if run simultaneously could cause a conflict.)
- 2) Load a flight and then **ensure the aircraft electrical power is on**. (Use the main battery switch.) the following example uses the Cessna 172, but any default aircraft will suffice.

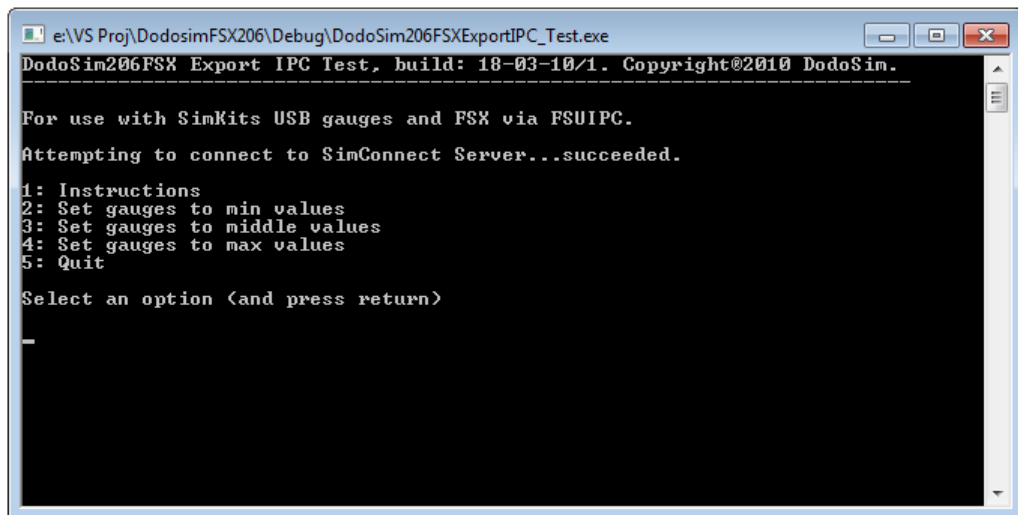


Battery switch on and electrical power evident

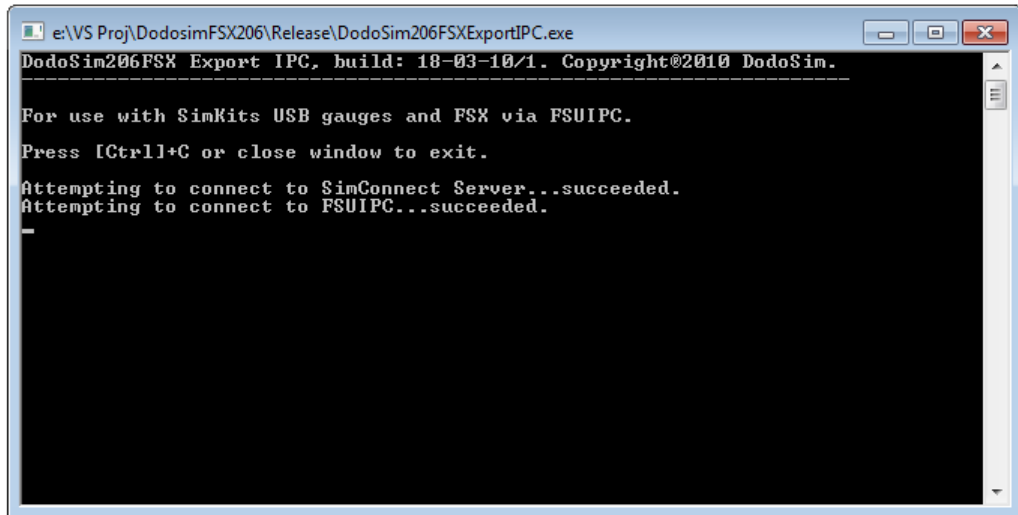
Important note: Battery power is required during set-up because some values do not present correctly when read via FSUIPC is power is off.

- 3) Run the *DodoSim206FSXExportIPC_Test.exe* program. It should open a window like this:

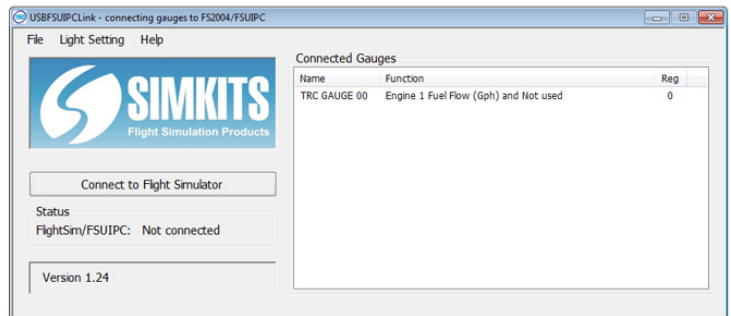
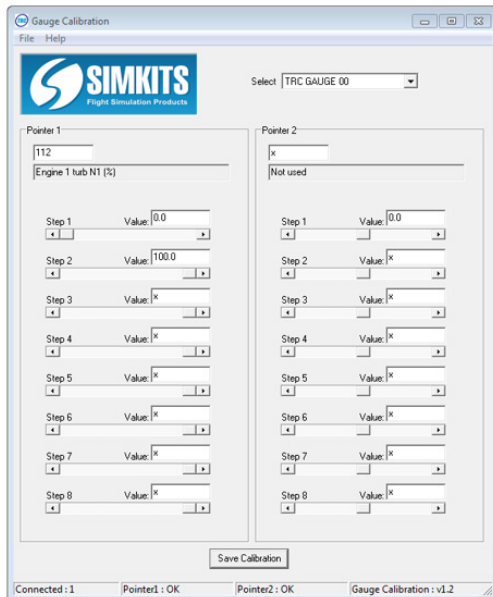
(Note that the program will pause with "Attempting to connect to SimConnect Server..." if it is started before FSX is running and a flight is loaded.)



- 4) Before selecting any options in the *DodoSim206FSXExportIPC_Test.exe* window, start the *DodoSim206FSXExportIPC.exe* program. It should open the expect window as previously shown:



- 5) Run the SimKits *USBFSUIPCLink* and *USBGaugeCalibration* programs. They should look something like this:



You should have an entry in the *USBFSUIPCLink*'s "Connected Gauges" window for each of your connected USB gauges.

Important note: Though SimKits provide an FSX specific version of their software, *USBFSXLink*, this is not used for the *DodoSim 206 FSX* since it uses SimConnect directly to access simulation variables and does not therefore provide us with the method of masking FS variables with DodoSim custom ones that FSUIPC does.

- 6) On the *DodoSim206FSXExportIPC_Test.exe* program, select option 1 for instructions. It will present this list:

```

e:\VS Proj\DodosimFSX206\Debug\DodoSim206FSXExportIPC_Test.exe

Start FSX, load an aircraft OTHER than the DodoSim 206 FSX.
Ensure 'DodoSim206FSXExportIPC.exe' is running.
Turn on electrical power.
Calibrate each gauge in turn using the SimKits function number list below.
Calibrate to the low, mid and high values below, entering the numbers displayed
in the step values in the Calibration Tool.
After calibration is complete, close this application but leave 'DodoSim206FSXExportIPC.exe' running.
Use only the FSUIPC compatible SimKits driver software with FSX.

Use these SimKits function numbers for each gauge:

busLoadGaugePercent = 108
engineOilGaugePressurePSI = 103
engineOilGaugeTemperatureC = 102
fuelPressureGaugePSI = 115
n1GaugePercent = 112
n2GaugePercent = 113
nRGaugePercent = 90
torqueGaugePercent = 110
transmissionOilGaugePressurePSI = 91
transmissionOilGaugeTemperatureC = 92
turbineTemperatureGaugeC = 114

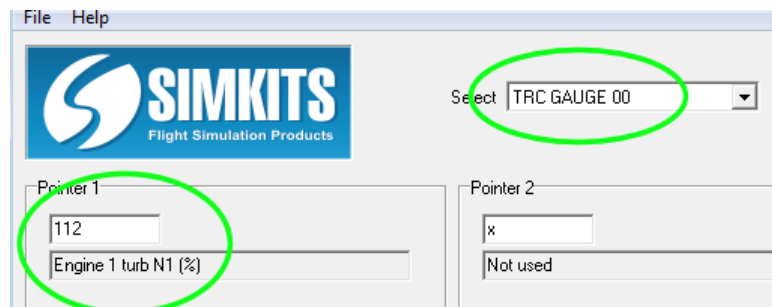
1: Instructions
2: Set gauges to min values
3: Set gauges to middle values
4: Set gauges to max values
5: Quit

Select an option <and press return>

```

Hint: Note that you may need to scroll up or enlarge the window to see all the instructions text.

You will see that the instructions contain a list of all supported gauges and the SimKits parameter number to use to drive the needle. Configure each of your SimKits gauges to use these values accordingly, i.e.



- 7) To calibrate each gauge through its range of values for the *DodoSim 206 FSX*, (which may cover a greater range than for standard aircraft), select each option in *DodoSim206FSXExportIPC_Test.exe* for minimum values (option 2), middle value (option 3) and maximum value (option 4) and use the sliders in the *USBGaugeCalibration* program to set the ranges.

Pressing either of the three options will present a list of the values at the gauge's respective position, i.e. Pressing option 4 for maximum values will present this screen:

You can see from this that, for example, the maximum value the *DodoSim 206 FSX* can drive the N1 to is 112%. (Normally aircraft in FSX do not drive gauges beyond 100%. The *DodoSim 206 FSX* is different in that the pilot can overspeed the engine and rotor systems through misuse.)


```

e:\VS Proj\DodosimFSX206\Debug\DodoSim206FSXExportIPC_Test.exe

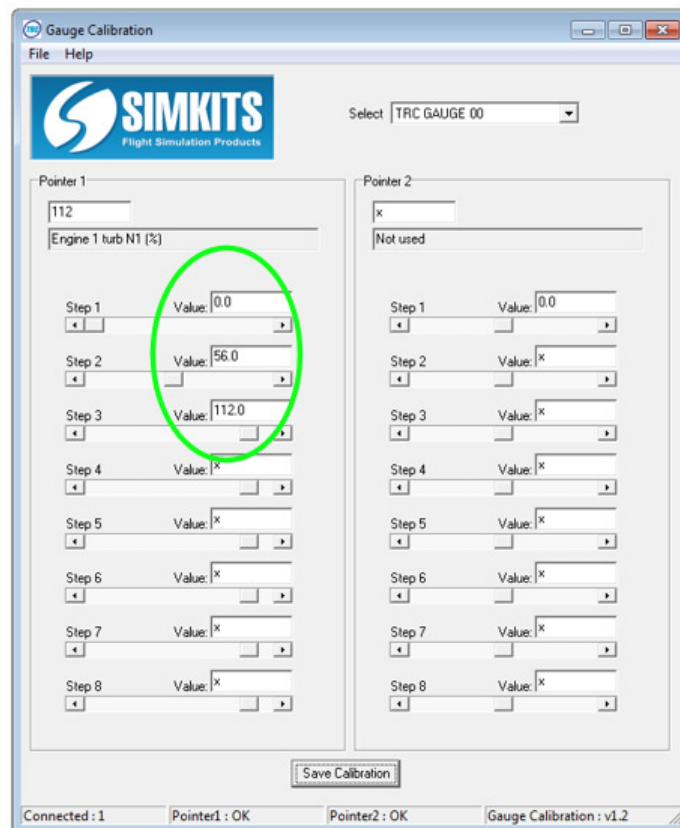
Setting gauges to max:
busLoadGaugePercent = 110.0
engineOilGaugePressurePSI = 150.0
engineOilGaugeTemperatureC = 150.0
fuelPressureGaugePSI = 35.0
n1GaugePercent = 112.0
n2GaugePercent = 130.0
nRgaugePercent = 130.0
torqueGaugePercent = 130.0
transmissionOilGaugePressurePSI = 150.0
transmissionOilGaugeTemperatureC = 150.0
turbineTemperatureGaugeC = 1000.0

1: Instructions
2: Set gauges to min values
3: Set gauges to middle values
4: Set gauges to max values
5: Quit

Select an option <and press return>

```

Therefore, the appropriate setting in the *USBGaugeCalibration* program would be as such, where the slider has been adjusted until the needle reaches the 112% mark on the dial face:



(Note that actual slider positions may vary from the example.)

- 8) Once you are confident your gauges are all calibrated, you can save your calibration, close the *USBGaugeCalibration* program and select option 5 in *DodoSim20FSX6ExportIPC_Test* to exit the program and close the window.
- 9) You can now fly the *DodoSim 206 FSX* and utilise your SimKits gauges, so long as you remember that you must always also have the *DodoSim206FSXExportIPC.exe* program running in conjunction with the SimKits *USBFSUIPLink* program.

2.3 DODOSIM206FSXDATAEXPORT

2.3.1 Description

DodoSim206FSXDataExport.exe is a DLL file which software writers can use to obtain the state of all annunciator lights specific to the *DodoSim 206 FSX* helicopter. It connects to the helicopter via SimConnect and fetches the client data area. Exported functions provide a means of fetching individual values from the DLL.

Though the DLL itself is written in C, the exported functions adhere to the standard DLL API and should therefore be accessible via any suitable programming language supporting the dynamic loading of DLLs. Library (.lib) and header (.h) files are provided for static linking should that be required by the programmer.

2.3.2 Usage

- 1) Refer to your compiler's documentation for linking and using DLLs.
- 2) The following exported functions are provided:

General:

```
double GetDLLVersion(void); // Get DLL version
```

Annunciators and Warning Lights:

```
int GetEngOutLightState(void);

int GetRotorLowLightState(void);

int GetFuelLowLightState(void);

int GetGenFailLightState(void);

int GetBatteryTempLightState(void);

int GetBatteryHotLightState(void);

int GetFuelPumpLightState(void);

int GetTransOilPressLightState(void);

int GetTransOilTempLightState(void);

int GetEngineChipLightState(void);

int GetTRChipLightState(void);

int GetTransChipLightState(void);

int GetAFFuelFilterLightState(void);

int GetBaggageDoorLightState(void);

int GetSCFailLightState(void);

int GetSpare1LightState(void); // below SC FAil

int GetSpare2LightState(void); // right of SC FAIL
```

```
int GetSpare3LightState(void); // left of GEN FAIL

int GetSpare4LightState(void); // right of FUEL LOW

int GetSpare5LightState(void); // below GEN FAIL

int GetTurbineOverTempLightState(void);
```

- 3) All DLL functions are exported using the standard C calling format, i.e.

```
extern "C" __declspec (dllexport)
```

Alternatively, the DLL can be opened at run-time and local function pointers mapped and used via (for example of Engine Out light in C):

```
typedef int (WINAPI *GETLIGHT) (void);
HMODULE hLibrary = LoadLibrary(TEXT("DodoSim206FSXDataExport.dll"));
GetEngOutLightState=(GETLIGHT)GetProcAddress(hLibrary, "GetEngOutLightState");
..
..
int l = GetEngOutLightState();
```

- 4) Light states return either 0 for off, or 1 for lit.

3 SUPPORT

These utilities are provided “as is” and no warranty is implied or suitability for use assured. DodoSim is not obliged to provide support or coding advice with regards to their use, but questions may be posted to technical support via www.dodosim.com and may be answered if a question is readily available.

To utilise the *DodoSim206FSXDataExport.dll* library you would be expected to be proficient in your chosen programming language and familiar with opening and handling Dynamic Link Libraries. DodoSim cannot offer tuition on writing software.

DodoSim cannot offer support on either installation or operation of FSUIPC or SimKits gauges and software.