



USB-PTS Interface Owner's Manual



SpinCore Technologies, Inc.
<http://www.spincore.com>

USB-PTS Interface

Congratulations and *thank you* for choosing a design from SpinCore Technologies, Inc.

We appreciate your business!

At SpinCore, we aim to fully support the needs of our customers. If you are in need of assistance, please contact us and we will strive to provide the necessary support.

© 2009-2019 SpinCore Technologies, Inc. All rights reserved.

SpinCore Technologies, Inc. reserves the right to make changes to the product(s) or information herein without notice. SpinCore, and the SpinCore Technologies, Inc. logos are trademarks of SpinCore Technologies, Inc. All other trademarks are the property of their respective owners.

SpinCore Technologies, Inc. makes every effort to verify the correct operation of the equipment. This equipment version is not intended for use in a system in which the failure of a SpinCore device will threaten the safety of equipment or person(s).

Table of Contents

<u>I. Introduction.....</u>	<u>4</u>
Product Overview.....	4
Power Options.....	4
<u>II. Installation.....</u>	<u>5</u>
Installing Your USB-PTS.....	5
<u>III. Programming.....</u>	<u>5</u>
USB-PTS GUI Controller Program.....	5
Example Executable Console Programs.....	7
C/C++ Programming Using SpinAPI.....	8
USB-PTS LabVIEW GUI.....	8
<u>IV. Related Products and Accessories.....</u>	<u>10</u>
<u>V. Contact Information.....</u>	<u>11</u>
<u>VI. Document Information.....</u>	<u>11</u>

I. Introduction

Product Overview

The USB-PTS Interface board allows for USB control of PTS (*Programmed Test Sources*) Frequency Synthesizers from a Windows, Linux or Macintosh platform. The USB-PTS Interface board plugs directly into the "Remote Control" input connector of the PTS device, and the USB cable provides the communication link between the PTS and your PC.

An easy to use Graphical User Interface (GUI) program, called the SpinCore USB-PTS Controller, is provided along with simple executable console programs and their source code. The board is controlled through the use of SpinCore's Application Programming Interface (SpinAPI) functions and can be used to create custom control programs.

A block diagram of the USB-PTS Interface is shown in Figure 1. The USB connectivity is through the standard, type-B, USB connector. The main interface chip, implemented using programmable logic, performs the necessary conversion of the incoming byte stream into the required decade logic of the PTS.

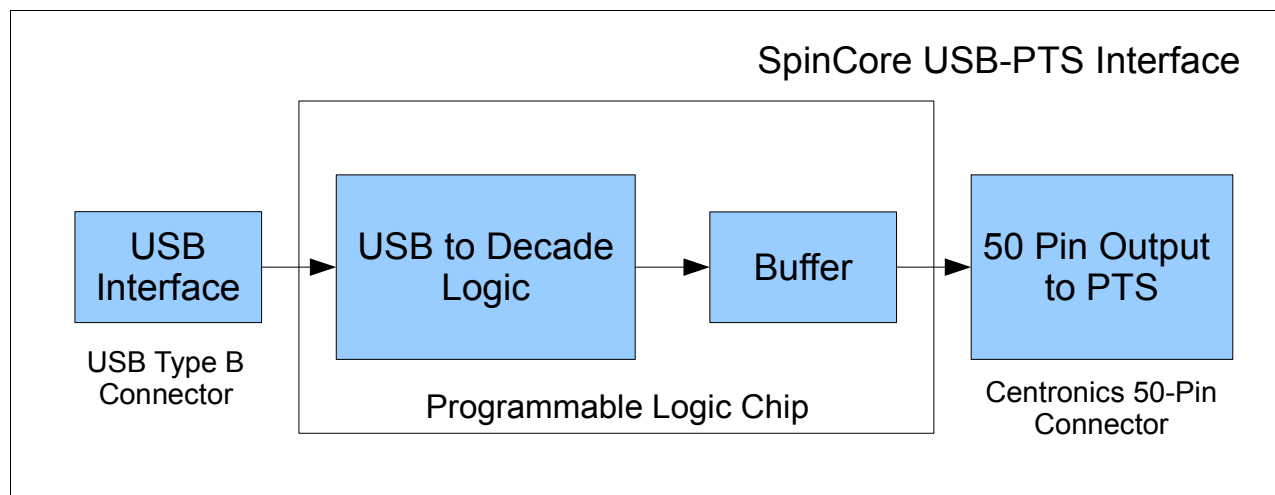


Figure 1: SpinCore USB-PTS Interface Block Diagram.

The USB-PTS device relies exclusively on data being sent from the host computer. No data is stored on the board, and thus no variable timing latencies are introduced by the device. Since the PTS update data rate is controlled solely by the host computer, delays could be introduced by the latencies of the host's operating system. For deterministic operation, please contact SpinCore for information on developing a new version of the USB-PTS that supports on-board memory.

Power Options

The user has the option to power the USB-PTS by USB or by the PTS device. To choose a power option, place a jumper on the three pin power header next to the USB connector. A diagram of the three pin header is shown below in Figure 2 on the next page. To select power from USB, place the included jumper on the USB and 5.0V pins. To select power from the PTS, place the jumper on the PTS and 5.0V pins. The power source will need to be supplied by the user.

USB-PTS Interface

WARNING: NEVER CONNECT USB AND PTS PINS ON THE POWER HEADER TOGETHER

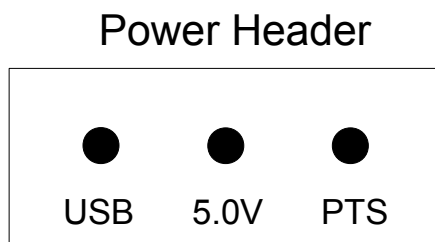


Figure 2: Power Header Connector Diagram.

II. Installation

Installing Your USB-PTS

To properly use the USB-PTS Interface, you must follow these three steps:

1. Install the latest version of the SpinAPI Driver Suite which can be downloaded from <http://www.spincore.com/support/spinapi/>.
2. Upon installation of the SpinAPI Driver Suite plug the USB-PTS Interface board into the PTS firmly.
3. Plug one end of the USB cable into the USB-PTS Interface board and the other end into the host computer and follow the installation prompts if displayed. When the device is recognized and the installation prompts are finished, you should consult the next section for information about programming your board.

Note: After the installation of the USB-PTS is complete, the Device Manager will show an “unknown device” in the USB Serial Bus Controllers. This has no affect on the installation or performance of the USB-PTS itself and can be ignored.

III. Programming

There are several different options for controlling the USB-PTS. The first and most straightforward option is to use the provided USB-PTS GUI Controller Program. The second method is to use the sample executable console programs for Windows users. The third and most flexible option is to create your own programs using the C/C++ functions from the SpinAPI package. A fourth option is to use the USB-PTS LabVIEW GUI. All four methods are described in more detail in the following four sections.

USB-PTS GUI Controller Program

The SpinCore USB-PTS GUI Controller Program is included with the latest SpinAPI package. Assuming the default installation under Windows, select Windows “Start” > All Programs > SpinCore > Examples > USB-PTS. You should run the USB-PTS_Controller.exe from Windows or USB-PTS_Controller.tcl from Linux or Macintosh. After opening the USB-PTS Controller Program, you can select your PTS model and control its frequencies.

USB-PTS Interface

There are two separate features of the USB-PTS Controller - Single Frequency Generation and Frequency Sweep Generation. To set your PTS to output one given frequency, use the Single Frequency Generation option and press "Set Synthesizer". To create a frequency sweep, set your Starting Frequency, Ending Frequency, Increment, Phase and Time Delay and press "Run Sweep". This program will only run through the sweep one time. For a way to run sweeps continuously, see the example executable console programs in the next section.

Figure 3, below, presents the Graphical Interface of the program. Note that due to inherent operating system latencies, specified delays in frequency sweeps are not perfectly accurate and delays shorter than 1 ms are typically not attainable. Also, for frequency sweeps, the GUI interface gives control to the actual executable sweep program at run time and the user does not regain control until the sweep has ended.

Note: Depending on your input for the sweep parameters, the program could take several minutes to complete without allowing user control. This is normal and not an error in the program, so be careful when setting the parameters as you will not have control throughout the duration of the sweep. If there is an error regarding missing files, make sure there are the appropriate .exe files in the includes folder. To generate .exe files, compile the .c files. More information on Windows compilation can be found [here](#).

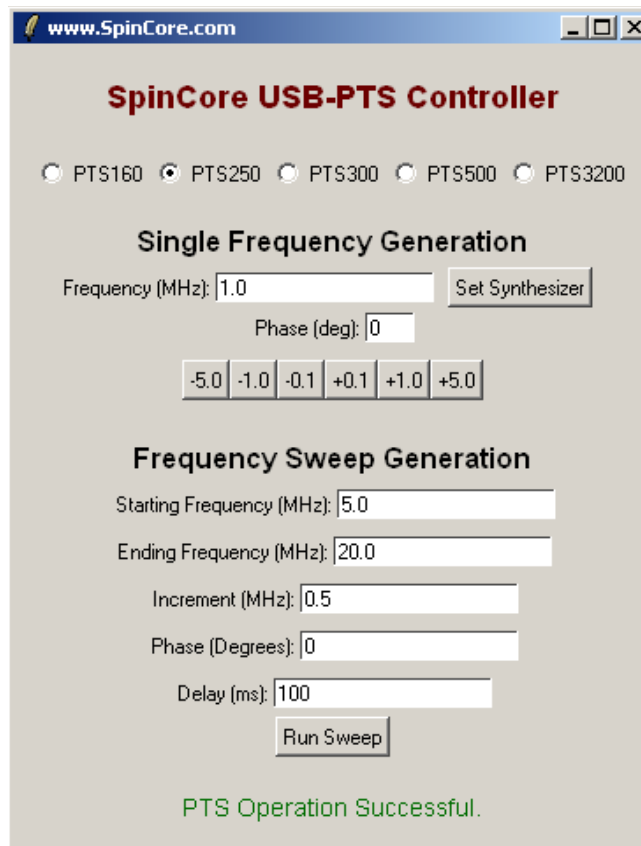
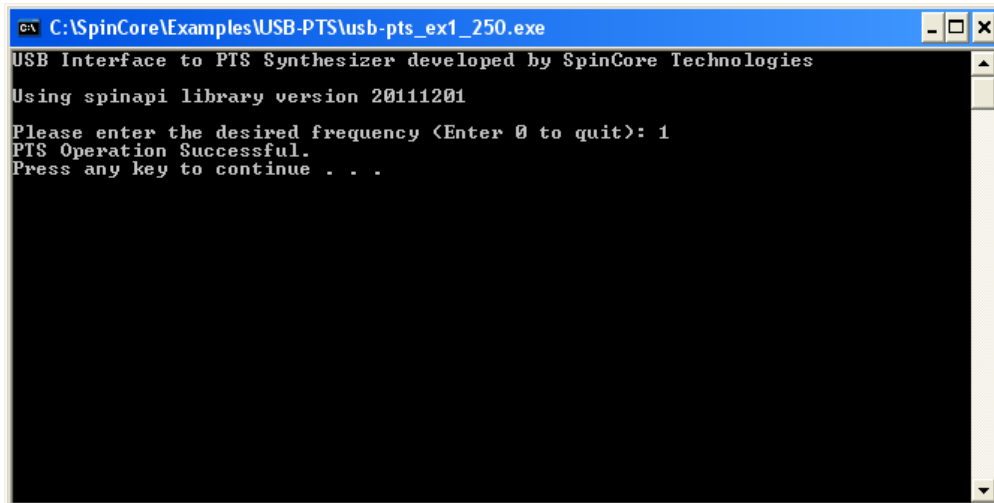


Figure 3: USB-PTS Controller Program GUI. In this example a single 1.0 MHz frequency can be generated or a sweep from 5 MHz to 20 MHz with an increment of 0.5 MHz can be run.

USB-PTS Interface

Example Executable Console Programs

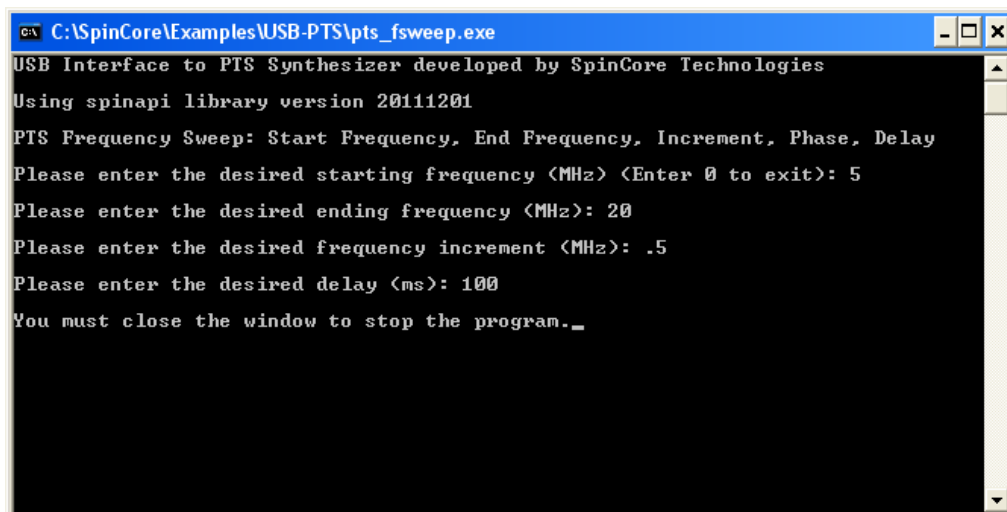
To use one of the example executable console programs, run one of the .exe files in the USB-PTS directory of SpinAPI Examples. To generate a single frequency from your PTS, run the appropriate “usb-pts_ex1” program. The end number of each “usb-pts_ex1” program corresponds to the number of the PTS model. For example, open up the program “usb-pts_ex1_250.exe” if you are using the PTS-250. It will prompt you for a frequency in MHz (megahertz). This program will allow you to continue to enter different frequencies until you choose to quit by entering a 0 for frequency. A screen shot of this program can be seen in Figure 4 below.



```
C:\SpinCore\Examples\USB-PTS\usb-pts_ex1_250.exe
USB Interface to PTS Synthesizer developed by SpinCore Technologies
Using spinapi library version 20111201
Please enter the desired frequency <Enter 0 to quit>: 1
PTS Operation Successful.
Press any key to continue . . .
```

Figure 4: Example Executable Console Program Screen Capture. This example shows a 1 MHz frequency.

To run a continuous frequency sweep, open up the program “pts_fsweep.exe”. It will prompt you to enter the following parameters: Starting Frequency, Ending Frequency, Increment, and Time Delay. When you have entered them all, your PTS should now be sweeping through the desired frequencies. In order to quit this program, exit out of the Command Prompt Window. A screen shot of this program can be seen in Figure 5 below.



```
C:\SpinCore\Examples\USB-PTS\pts_fsweep.exe
USB Interface to PTS Synthesizer developed by SpinCore Technologies
Using spinapi library version 20111201
PTS Frequency Sweep: Start Frequency, End Frequency, Increment, Phase, Delay
Please enter the desired starting frequency <MHz> <Enter 0 to exit>: 5
Please enter the desired ending frequency <MHz>: 20
Please enter the desired frequency increment <MHz>: .5
Please enter the desired delay <ms>: 100
You must close the window to stop the program._
```

Figure 5: Example sweep program. This example sweeps from 5 to 20 MHz with an increment of .5 MHz.

C/C++ Programming Using SpinAPI

The most dynamic and flexible way to control the frequencies of your PTS is with C/C++ using the SpinAPI Suite. The GUI based approaches to controlling the board are designed for ease of use and have limited functionality for a specific end-use application. Coding in C/C++ allows for more flexibility in creating various, more advanced, control programs for the PTS. The instructions to compile on Windows can be found at http://www.spincore.com/support/spinapi/Windows_Help.shtml. After configuring the compiler, changing one of our example programs and recompiling the executable file for use with your USB-PTS board is as easy as clicking “Rebuild All”.

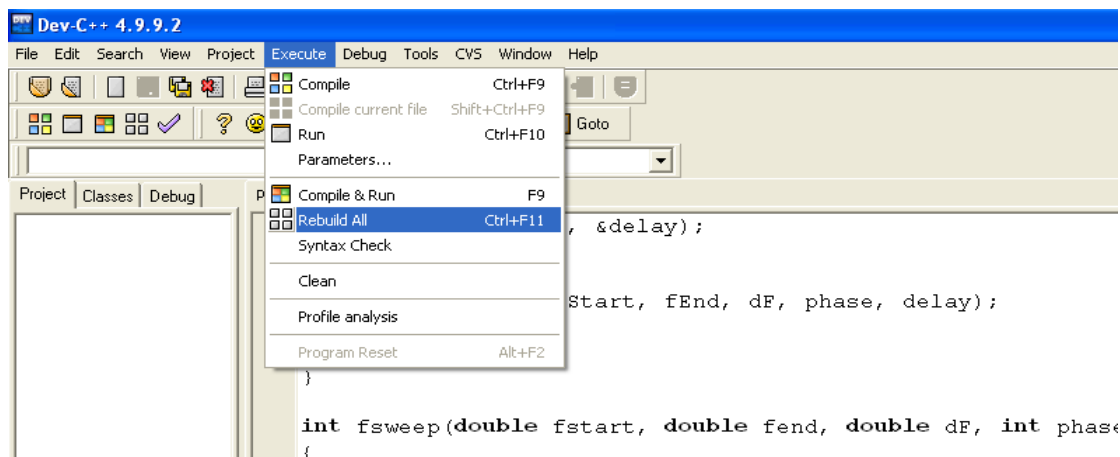


Figure 6: C/C++ Example Compiler Screen shot.

To use the SpinAPI add the line `#include "spinapi.h"` at the top of your program. The function used to control the parameters of the USB-PTS is:

```
int set_pts(double maxFreq, int is160, int is3200, int allowPhase,  
            int noPTS, double frequency, int phase);
```

Using the `set_pts()` function you can easily control the PTS by adjusting the frequency and phase parameters above. For more information details for programming in C, consult the example programs which can be found in Windows “Start” > All Programs > SpinCore > Examples > USB-PTS.

USB-PTS LabVIEW GUI

There are two separate features of the USB-PTS LabVIEW Program - Single Frequency Generation and Frequency Sweep Generation. To set your PTS to output one given frequency, select the Single Frequency Generation tab, enter in the desired frequency and phase, and press “Set Synthesizer”. To create a frequency sweep, select the Frequency Sweep Generation tab, set your Starting Frequency, Ending Frequency, Increment, Phase and Time Delay and press “Run Sweep”. This program will only run through the sweep one time.

Figure 7, on the next page, presents the Graphical Interface of the program. Note that due to inherent operating system latencies, specified delays in frequency sweeps are not perfectly accurate and delays shorter than 1 ms are typically not attainable. Also, for frequency sweeps, the GUI interface gives control to the actual executable sweep program at run time and the user does not regain control until the sweep has ended.

USB-PTS Interface

Note: Depending on your input for the sweep parameters, the program could take several minutes to complete without allowing user control. This is normal and not an error in the program, so be careful when setting the parameters as you will not have control throughout the duration of the sweep.

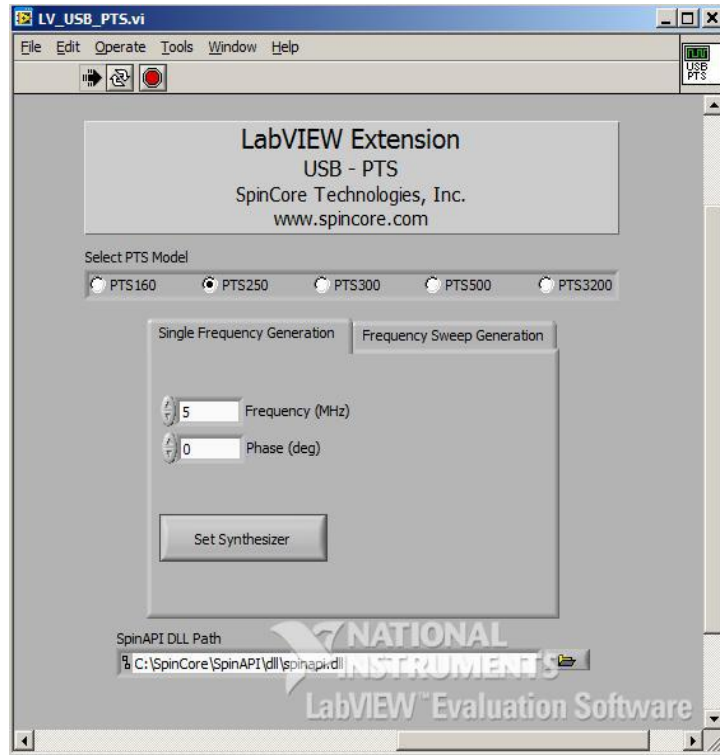


Figure 7: USB-PTS LabVIEW Program GUI. In this example a single 5.0 MHz frequency with a 0 degree phase can be generated.

You can find more information about the LabVIEW interface and download the LabVIEW interface here: <http://www.spincore.com/support/PBLV/USBPTS.shtml>

IV. Related Products and Accessories

SpinCore can develop new hardware that improves upon the USB-PTS Interface with various options, including the following:

- Support for additional PTS models.
- Hardware trigger/update.
- USB 2.0.
- On-board SRAM for deterministic, real-time operation.

Please inquire with SpinCore Technologies through our contact form, which is available at <http://spincore.com/contact.shtml>, for more information.

V. Contact Information

SpinCore Technologies, Inc.
4631 NW 53rd Avenue, SUITE 103
Gainesville, FL 32653
USA

Telephone (USA):	352-271-7383
Website:	http://www.spincore.com
Web Contact Form:	http://spincore.com/contact.shtml

VI. Document Information

Revision History Available at SpinCore.