July 25, 2011

# MATLAB Support Package for Velleman PCSGU250 User's Guide

## Contents

## Introduction

The MATLAB Support Package for Velleman PCSGU250 (the Support Package) is a MATLAB interface to the functions of the Velleman PCSGU250 oscilloscope and function generator.

Package Requirements

- Windows 2000 or later (32-bit or 64-bit)
- 32-bit[1] MATLAB R2009a or later (which can be installed on a 64-bit system)
- USB driver for the PCSGU250 (instructions are included below)
- PCSGU250 SDK (automatically downloaded during setup)

[1] This restriction is because the PCSGU250 provides an external interface through 32-bit DLLs. 64-bit Windows applications cannot access 32-bit DLLs, so 64-bit MATLAB would be unable to access the PCSGU250.

# Installation Instructions

## Step 1: Install the PCSGU250 USB Driver

The PCSGU250 USB drivers do not come with an installer so they will have to be installed manually. In Windows, this process can be difficult. The following instructions were written using Windows 7 but should be general enough for any version of Windows. If there are any problems, there try searching the Internet for guides about manually installing drivers (search for "manually installing windows drivers" or similar).

IF YOU HAVE THE PCSGU250 CD:

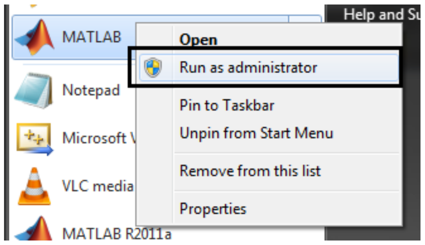
1. The CD should come with an instruction guide. Follow the guide's instructions to install USB drivers.

OTHERWISE:

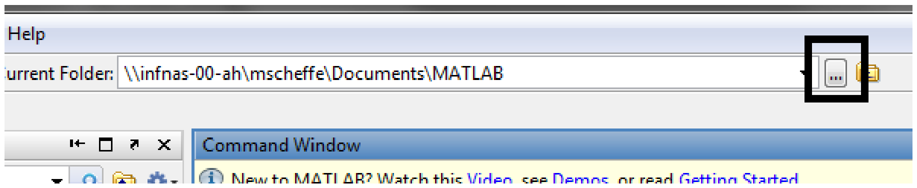
1. If you already have drivers (ex. if you have the CD), skip to step 3. If you don't, go to this webpage and click the appropriate link ("drivers for Windows Vista and Windows 7") to download USB drivers.
2. Unzip the USB drivers to a convenient location (for example, the C:\ directory).
3. Plug the PCSGU250 into the computer.
4. Open Device Manager.
5. Find the PCSGU250 (look under Unknown or Other Devices).
6. Right click on the PCSGU250 and click on "Update Driver Software..."
7. Click on "Browse my computer for driver software"
8. Click on "Browse" and navigate to the USB driver folder (PCSGU250D on the CD, whatever folder they were unzipped to for downloaded drivers).
9. If Windows displays an error, click "Install this driver software anyway".
10. Installation should be complete.

## Step 2: Run setup

1. Run MATLAB as an administrator by right clicking on the MATLAB icon and selecting “Run as administrator”.



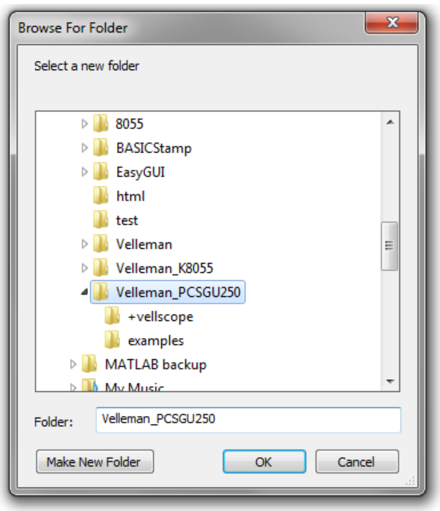
2. Click on the ellipsis (...) to the right of the “Current Folder” prompt at the top of the screen



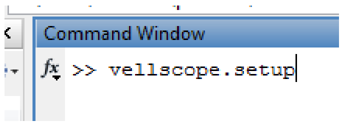
\\infnas-00-ah\mscheffe\Documents\MATLAB

New to MATLAB? Watch this Video, see Demos, or read Getting Started

3. Navigate to this folder (“\Velleman_PCSGU250\”) and click “OK”.

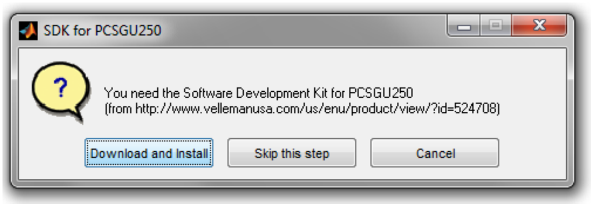


4. In the Command Window, type “vellscope.setup” without the quote marks



```
>> vellscope.setup
```

5. The following dialog box should pop up. Click "Download and Install".



SDK for PCSGU250

You need the Software Development Kit for PCSGU250
(from http://www.vellemanusa.com/us/enu/product/view/?id=524708)

6. Setup will automatically try to download the SDK from this webpage. This may take some time. If an error occurs, see "Troubleshooting Setup" in the "Troubleshooting" section of this guide. Note that this will work even if the SDK has been downloaded to the computer previously.
7. If the Command Window displays a message similar to the following, setup has completed successfully. Setup may output warnings as well; these do not necessarily mean setup has failed.

```
>> vellscope.setup
# Installed Velleman Software Development Kit (SDK) to C:\Velleman\PCSGU250_DLL
# Added Velleman package to MATLAB path
# Added 'C:\Users\mscheffe\perforce\mscheffe.sandbox\Velleman PCSGU250' to MATLAB path
You can now navigate to any other directory and use the Velleman package.
See this example to get started
```

## Getting Started

The MATLAB Support Package for Velleman PCSGU250 is a MATLAB package that uses MATLAB's support for object oriented (OO) programming (For more information: MATLAB Packages, OO Programming in MATLAB). Users should be familiar with the MATLAB interface and the basics of MATLAB OO concepts (creating and destroying objects, calling methods, and accessing properties).

## Creating a pclab Object

The functionality of the PCSGU250 is accessible through the MATLAB class pclab. To create a pclab object called lab, type the following in the Command Window:

```
lab = vellscope.pclab
```

This will load the PCSGU250 library and bring up the PCSGU250 GUI. The PCSGU250's functionality will then be accessible through the methods and properties of lab. The rest of this guide assumes that lab is a pclab object that has already been created.

## Accessing Functionality

The PCSGU250 is accessible through pclab's methods and properties. In general, properties are used to change settings (ex. volts per division, trigger edge setting, function generator frequency), while methods are used to perform actions (ex. turn on the scope, get data from the scope, turn on the function generator). A full list of properties and methods can be found in the Reference section later in this guide.

### Using the Scope

To turn on the scope, type the following:

```
lab.startScope
```

This will make the PCSGU250 start collecting data from both its input channels. Alternatively, to make the scope collect just one screen's worth of data, type the following:

```
lab.startScope('Single')
```

This will wait for a screen of data to be generated and then stop (generally this should be used in conjunction with a trigger).

To change the scope settings, change lab's properties. For example, to change the volts per division for Channel 1 to 3V, type the following:

```
lab.VoltsPerDiv_Ch1 = '3V';
```

Note that the value is a string, not a number. This is the case for all properties that allow only certain specific values. These values are listed in the help text for each property (See "Accessing Help") and in the error that appears when a property is assigned an incorrect value. Properties that take a range of values (for example, YPosition_Ch1 takes any number between -128 and 127) use numeric values instead.

To get data from the scope, use readChannel. readChannel takes a channel number as an arguments and outputs a 1x4096 double containing the last set of voltage data collected on the scope. For example, to read data from channel 1 and store it in a variable called data, type:

```
data = lab.readChannel(1);
```

One thing to note is that this does not simply grab data from the GUI viewing screen. Instead, it reads all data from the most recently collected set, including data that is only visible in the GUI by scrolling the viewing screen. This also means that the first point won't be the point that triggered the scope; the trigger point is instead at about entry 1000 of the resulting vector. See readChannel's help for more information.

pclab has a property, DataReady, that indicates whether the PCSGU250 has new data to output. Use this to automatically determine whether MATLAB should read data from the PCSGU250.

To use the PCSGU250 trigger, type the following:

```
lab.Trigger = 'on';
```

The trigger settings can be changed just like the scope settings by changing lab's trigger properties.

To stop the scope, type:

```
lab.stopScope;
```

### Using the Function Generator

Unlike the scope, the function generator does not update settings in real time. Therefore, it's recommended that properties are set before starting the generator.

The most important property for the function generator is FgenSource. This determines which class of function the generator will use. FgenSource can take on three values: 'standard', 'library', and 'sweep'. The standard functions are a sine wave, triangle wave, and square wave; which is used is determined by FgenFunction.  A library function as an arbitrary waveform defined by a library (*.lib) file, which is specified by FgenLibFile. A sweep is a sinusoidal sweep between two frequencies over a period of time.

More information on what properties control the function generator can be found in the property help texts (see "Accessing Help"). To see more information on library files and arbitrary waveforms, see "Creating a Library File".

Once properties have been set, type the following to start the function generator:

```
lab.startGen
```

This will start the generator using the specified settings. If settings change, use startGen to update the outputted waveform. To stop the function generator, type:

```
lab.stopGen
```

## Cleaning Up

It's good practice to delete objects when they're no longer in use. This frees up memory and can prevent hard-to-trace errors. To delete lab, type:

```
delete(lab);
```

Never try to delete a pclab object by deleting the PCSGU250 GUI. This will cause the SDK to become unresponsive. Trying to call a pclab method will hang up MATLAB as it waits for the SDK to respond. To get rid of the GUI, use pclab's hide method instead.

If the GUI is deleted accidentally, just delete the lab object and create it again. This will restart the SDK and prevent any issues.

## Accessing Help

The Support Package is fully documented with help text for all of pclab's methods and properties, as well as for all other included files. To access help, use MATLAB's help command. For example, to access help for pclab, pclab's show method, pclab's TriggerEdge property, and the setup function, type:

```
help vellscope.pclab
help vellscope.pclab.show
help vellscope.pclab.TriggerEdge
help vellscope.setup
```

Note that the package name (vellscope) is necessary. Trying to call "help pclab" will cause an error.

Help can also be accessed using MATLAB's doc function by replacing "help" with "doc" (ex. "doc vellscope.pclab"). This will display help in a separate window and allow browsing of help for related items.

Help for pclab's properties will also show valid values for those properties. For properties that are strings, these values are case sensitive; for example, TriggerEdge could be set to 'positive' but trying to set it to 'Positive' would cause an error.

To access a method list with help links, click on the "Method List" link that appears at object creation, or call pclab's showMethods static method. To access a property list, click on "Property List" or call showProperties.

## Tips for Automating Operation

One powerful feature of the Support Package is that operation of the PCSGU250 can be automated to various degrees using MATLAB scripting. For example, it's relatively easy to write a MATLAB script that will start up the GUI, change settings, start the scope and collect data automatically. However, there are a few things to be aware of.

Method calls take some time to propagate to the PCSGU250 and take effect. This means that calling two methods immediately after each other may mean that the second method doesn't "see" the effects of the first method. For example, look at the following lines of code:

```
lab = vellscope.pclab;
startScope;             %start the scope
data = readChannel(1); %read data
delete(lab);
```

Now, more likely than not, data will be full of zeros, which is an extremely unlikely result in the real world. This is because, while the scope has started, it probably hasn't collected any data yet. Therefore, when MATLAB asks the SDK for data, it outputs an empty data set. In this case, one should use pclab's DataReady property to see if there is data to collect before collecting it. Here's another example, where a user is trying to see the output from a function generator using the scope:

```
lab = vellscope.pclab;
lab.startScope; %start the scope
lab.startGen;   %start the function generator

while(~lab.DataReady) end   %wait for data
data = lab.readChannel(1);  %read data
delete(lab);
```

This code uses DataReady to make sure the scope is collecting data before importing that data to MATLAB. However, this code is unlikely to see the function generator output, even if the hardware is hooked up correctly. The solution here is to use the MATLAB pause command to give the function generator time to start outputting. The following code should work correctly:

```
lab = vellscope.pclab;
lab.startScope; %start the scope
lab.startGen;   %start the function generator

pause(1); %wait for scope and FG to start
data = lab.readChannel(1);  %read data
delete(lab);
```

A 1s pause gives both the scope and function generator plenty of time to start up. In general, when code works but doesn't return expected results, try adding pause functions between interdependent method calls.

In general, the following timing limitations apply to pclab:

- The SDK will take several seconds to load; MATLAB will automatically wait for this to happen before any method calls are made.
- The scope takes about 200ms to start collecting data.
- The function generator takes about 500ms before starting output (very rough estimate, a 500ms pause is pretty safe, a 1s pause is very safe).
- Changing scope properties will take effect at the next scope refresh
- Changing function generator properties will take effect at the next startGen call

## Reference Guide and Advanced Topics

### Method List

Optional arguments are listed in *italics*.

| | |
|---|---|
| `show()` | Shows the PCSGU250 GUI |
| `hide()` | Hides the PCSGU250 GUI |
| `startScope(`*`single`*`)` | Starts the scope (optionally in single mode) |
| `stopScope()` | Stops the scope |
| `readChannel(channel)` | Reads data from the specified channel |
| `startGen()` | Start function generator output |
| `stopGen()` | Stop function generator output |

### Property List

| **Scope Properties** | |
|---|---|
| `VoltsPerDiv_Ch1` | Volts per division for channel 1 |
| `VoltsPerDiv_Ch2` | Volts per division for channel 2 |
| `TimePerDiv` | Time per division for the GUI output |
| `Trigger` | Whether the trigger is on (values are 'on' or 'off') |
| `TriggerEdge` | Which edge the trigger will capture ('positive' or 'negative') |
| `TriggerSource` | The trigger source channel ('ch1' or 'ch2') |
| `TriggerLevel` | The relative trigger level (0-255) |
| `Y_Position_Ch1` | The y offset of channel 1 (-128 – 127) |
| `Y_Position_Ch2` | The y offset of channel 2 (-128 – 127) |
| `Coupling_Ch1` | The coupling type of channel 1 ('AC', 'DC', 'GND') |
| `Coupling_Ch2` | The coupling type of channel 2 ('AC', 'DC', 'GND') |
| `DataReady` | Indicates that there is data to read |
| | |
| **Function Generator Properties** | |
| `FgenSource` | The type of function to use ('standard', 'library', 'sweep') |
| `FgenFunction` | The standard function to use ('sine', 'triangle', 'square') |
| `FgenFrequency` | The output frequency of the function generator |
| `FgenAmplitude` | The peak-to-peak amplitude of the output waveform |
| `FgenOffset` | The voltage offset of the output waveform |
| `FgenLibFile` | The library file to use in library mode |
| `SweepStartFreq` | The start frequency for sinusoidal sweeps |
| `SweepStopFreq` | The end frequency for sinusoidal sweeps |
| `SweepTime` | The period of each frequency sweep |

### Demo Mode

The PCSGU250 can be run in Demo Mode. This will not connect to any PCSGU250 devices, but still allow access to a "dummy" PCSGU250. To turn Demo Mode on and off, go to Options→Hardware Settings in the PCSGU250 GUI. Note that creating a pclab object without a PCSGU250 connected will cause the SDK to launch in demo mode until it's turned off manually.

## Creating a Library File

The support package provides a function called makeLibFile that allows for easy creation of the library files used in the function generator. makeLibFile takes two arguments: a data vector and a filename. The data vector is a MATLAB vector (ie. 1xn or nx1 matrix) that contains relative voltage data for one period of the waveform. This data vector can be any length; however, longer is almost always better. The PCSGU250 SDK will automatically linearly interpolate the result, meaning that a shorter vector means a less accurate waveform. "Smooth" waveforms, such as sine waves, should have more data points to ensure accurate output. Depending on the application, this could mean hundreds or even thousands of points (which is usually easy to generate using MATLAB).

Note that the library file only determines the shape of the waveform. The actual voltage and frequency (in this case, how many times the waveform repeats per second) are determined by FgenAmplitude and FgenFrequency. Also note that FgenAmplitude is peak-to-peak voltage, not magnitude, and is measured based on a function that reaches both the positive and negative sides of that amplitude. This means that the maximum absolute value (ie. "biggest" number, positive or negative) of the output waveform will be equal to FgenAmplitude/2, no matter what that waveform looks like.

The filename can be any Windows filename and should not have an extension. This is the same filename that is used in setting pclab's FgenLibFile setting. Example filenames are 'unitpulse' and 'Sawtooth Wave'. Library files are stored in the lib directory in the PCSGU250 SDK directory.

For an example of how to create a library file, see makeSampleLibs in the examples folder. This script creates two simple waveform library files: an AM (amplitude modulation) wave and a noisy sine wave.

## Viewing the SDK Location

The location where pclab expects to find the PCSGU250 SDK is stored as a MATLAB preference. To access this location, use the Prefs class. To create a Prefs object:

```
    pref = vellscope.Prefs
```

This will load the SDK location and display it as a property of pref called SDKDir. This property can be retrieved and set like any other class property. Note that setting the SDK location will automatically check the location and perform some setup steps. Only change the SDK location using this method if there are two or more SDK directories on the computer and Setup chose the wrong one.

# Troubleshooting

Before consulting this section, make sure to read the rest of this guide. Also, make sure that it's the Support Package, not user code, that's causing errors. Feel free to contact classroom-resources@mathworks.com with any questions or comments on the MATLAB Support Package for Velleman PCSGU250.

## Troubleshooting Setup

```
??? Unable to download the SDK. Check your internet connection.
If you are connected to the internet and still see this error,
see README.pdf for troubleshooting information.
```

Setup was unable to download the SDK. Check that the computer is connected to the Internet and that the Velleman website is up by navigating to the following webpage: http://www.vellemanusa.com/us/enu/home/. If this website is accessible but setup still fails, see "Manually Installing the PCSGU250 SDK" below.

```
??? Unable to unzip the SDK
See README.pdf for troubleshooting instructions.
```

Setup was able to download the SDK but was unable to unzip it. See "Manually Installing the PCSGU250 SDK" below.

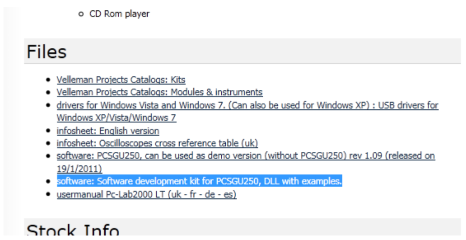
```
??? Unable to copy FASTTime32.dll
See README.pdf for troubleshooting instructions.

??? Unable to create lib folder
See README.pdf for troubleshooting instructions.
```

Setup was unable to make necessary modifications to the SDK's file structure. See "Manually Modifying the PCSGU250 SDK" below.

### Manually Installing the PCSGU250 SDK

1. Go to this webpage.
2. Click the link: "software: Software development kit for PCSGU250, DLL with examples" under Files.



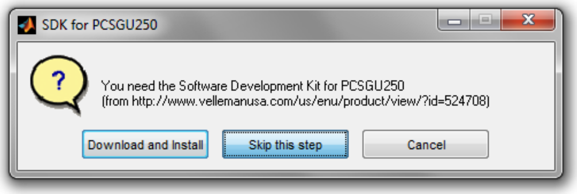
- CD Rom player

Files

- Velleman Projects Catalogs: Kits
- Velleman Projects Catalogs: Modules & instruments
- drivers for Windows Vista and Windows 7. (Can also be used for Windows XP) : USB drivers for Windows XP/Vista/Windows 7
- infosheet: English version
- infosheet: Oscilloscopes cross reference table (uk)
- software: PCSGU250, can be used as demo version (without PCSGU250) rev 1.09 (released on 19/1/2011)
- software: Software development kit for PCSGU250, DLL with examples.
- usermanual Pc-Lab2000 LT (uk - fr - de - es)

Stock Info

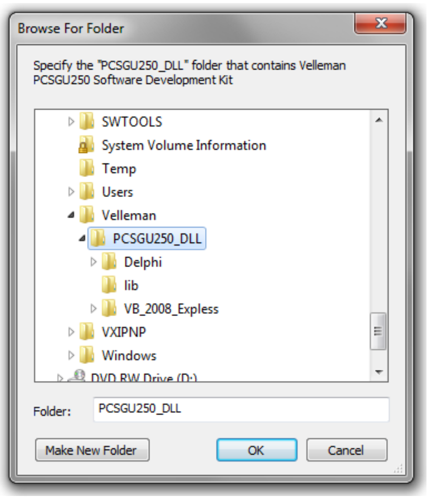
3. Download pcsgu250_dll_rev1.zip to a convenient location.
4. Unzip pcsgu250_dll to a convenient location.
5. Run setup again (see "Step 2: Run setup", step 4).
6. Click "Skip this Step".



SDK for PCSGU250

You need the Software Development Kit for PCSGU250
(from http://www.vellemanusa.com/us/enu/product/view/?id=524708)

7. The following file dialog will pop up. Navigate to and select the folder "PCSGU250_DLL" from the unzipped SDK.



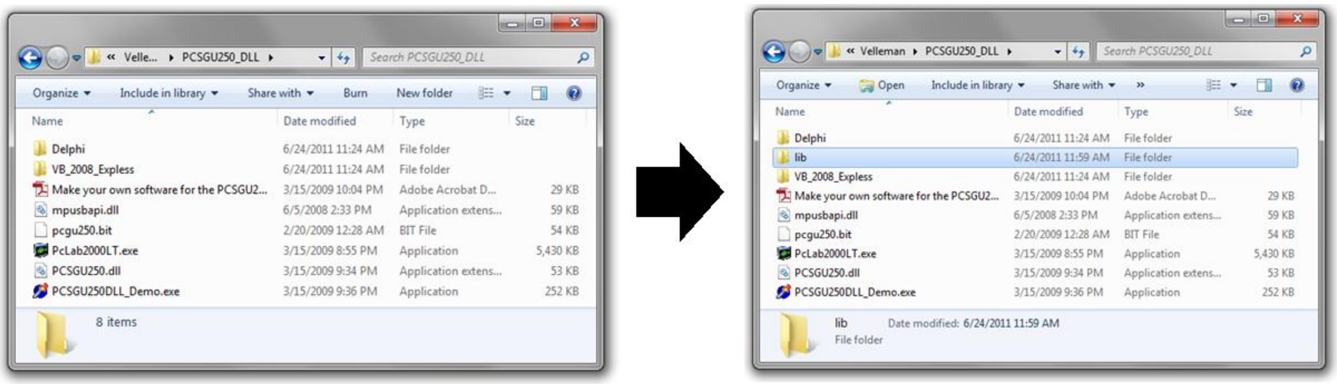
Specify the "PCSGU250_DLL" folder that contains Velleman PCSGU250 Software Development Kit

Folder: PCSGU250_DLL

8. Setup should complete normally. If an error listing missing files appears, check that the correct folder was specified and that the files listed aren't missing from the PCSGU250_DLL directory. If they are, try downloading the SDK again. For other errors, see "Troubleshooting Setup".

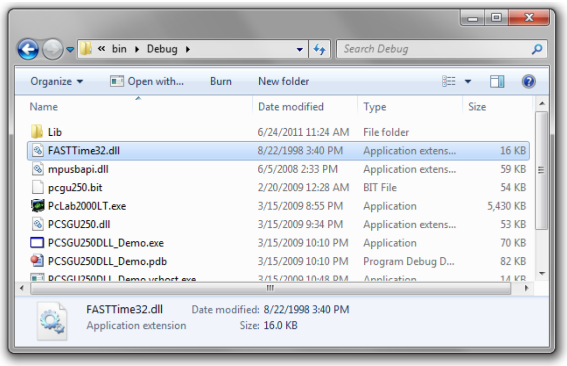
## Manually Modifying the PCSGU250 SDK

1. Navigate to the folder "PCSGU250_DLL". If setup downloaded the SDK, it will be located at "C:\Velleman\ PCSGU250_DLL". If the SDK was downloaded manually, this will be wherever the SDK was unzipped.
2. Create a folder called "lib" in the PCSGU250_DLL directory.


« Velle... › PCSGU250_DLL ›
Search PCSGU250_DLL
Organize
Include in library
Share with
Burn
New folder
8 items
« Velleman › PCSGU250_DLL ›
Open
lib Date modified: 6/24/2011 11:59 AM
File folder

| Name | Date modified | Type | Size |
|---|---|---|---|
| Delphi | 6/24/2011 11:24 AM | File folder | |
| lib | 6/24/2011 11:59 AM | File folder | |
| VB_2008_Expless | 6/24/2011 11:24 AM | File folder | |
| Make your own software for the PCSGU2... | 3/15/2009 10:04 PM | Adobe Acrobat D... | 29 KB |
| mpusbapi.dll | 6/5/2008 2:33 PM | Application extens... | 59 KB |
| pcgu250.bit | 2/20/2009 12:28 AM | BIT File | 54 KB |
| PcLab2000LT.exe | 3/15/2009 8:55 PM | Application | 5,430 KB |
| PCSGU250.dll | 3/15/2009 9:34 PM | Application extens... | 53 KB |
| PCSGU250DLL_Demo.exe | 3/15/2009 9:36 PM | Application | 252 KB |

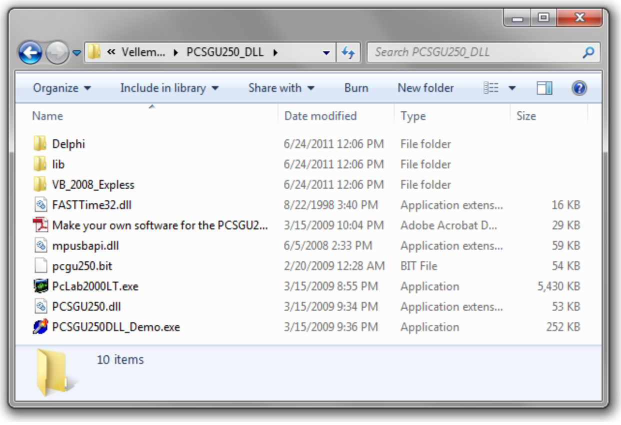
3. Navigate to "...\PCSGU250_DLL\VB_2008_Expless\PCSGU250DLL_Demo\bin\Debug".
4. Copy the file "FASTTime32.dll" (Ctrl+C).



| Name | Date modified | Type | Size |
|---|---|---|---|
| Lib | 6/24/2011 11:24 AM | File folder | |
| FASTTime32.dll | 8/22/1998 3:40 PM | Application extens... | 16 KB |
| mpusbapi.dll | 6/5/2008 2:33 PM | Application extens... | 59 KB |
| pcgu250.bit | 2/20/2009 12:28 AM | BIT File | 54 KB |
| PcLab2000LT.exe | 3/15/2009 8:55 PM | Application | 5,430 KB |
| PCSGU250.dll | 3/15/2009 9:34 PM | Application extens... | 53 KB |
| PCSGU250DLL_Demo.exe | 3/15/2009 10:10 PM | Application | 70 KB |
| PCSGU250DLL_Demo.pdb | 3/15/2009 10:10 PM | Program Debug D... | 82 KB |
| PCSGU250DLL_Demo.vshost.exe | 3/15/2009 10:48 PM | Application | 14 KB |

FASTTime32.dll Date modified: 8/22/1998 3:40 PM
Application extension Size: 16.0 KB

5. Paste (Ctrl+V) the file in the PCSGU250_DLL directory.
6. Modifications are now complete. PCSGU250_DLL should now look like this:



| Name | Date modified | Type | Size |
|---|---|---|---|
| Delphi | 6/24/2011 12:06 PM | File folder | |
| lib | 6/24/2011 12:06 PM | File folder | |
| VB_2008_Expless | 6/24/2011 12:06 PM | File folder | |
| FASTTime32.dll | 8/22/1998 3:40 PM | Application extens... | 16 KB |
| Make your own software for the PCSGU2... | 3/15/2009 10:04 PM | Adobe Acrobat D... | 29 KB |
| mpusbapi.dll | 6/5/2008 2:33 PM | Application extens... | 59 KB |
| pcgu250.bit | 2/20/2009 12:28 AM | BIT File | 54 KB |
| PcLab2000LT.exe | 3/15/2009 8:55 PM | Application | 5,430 KB |
| PCSGU250.dll | 3/15/2009 9:34 PM | Application extens... | 53 KB |
| PCSGU250DLL_Demo.exe | 3/15/2009 9:36 PM | Application | 252 KB |

10 items

7. Follow steps 5-8 of "Manually Installing the PCSGU250 SDK" to run setup again.

## Other Troubleshooting Topics

### MATLAB is hanging up

MATLAB will hang up if it tries to access an unresponsive SDK. The SDK seems to become unresponsive when the GUI is closed manually, either by closing its window or otherwise terminating the PcLab2000LT.exe process. If MATLAB hangs up like this (ie. if it still seems to be working but is constantly busy), kill the MATLAB process using Task Manager and restart MATLAB.

### pclab is in demo mode even though I have a PCSGU250 connected

The PCSGU250 GUI (and by extension, the SDK) will remember whether it was last started in demo mode or not. This means that if it was in demo mode for whatever reason (such as by creating a pclab object without having a PCSGU250 connected), it will stay in demo mode until it is manually changed back. To change it back, go to Options→Hardware Settings in the PCSGU250 GUI with a PCSGU250 connected to the computer. If the GUI says there is no hardware response, check that the PCSGU250 is connected properly. If it is, try reinstalling the USB drivers (see Setup).

### My MATLAB scripts are giving me unexpected results

See "Tips for Automating Operation".