PLECS

Tutorial

Custom Libraries in PLECS Standalone

How to create and update user defined component libraries and model references

Tutorial Version 1.0

www.plexim.com

- Request a PLECS trial license
- Check the PLECS documentation

1 Introduction

The aim of this exercise is to build a PLECS model using custom components from a user-defined library. Libraries enable you to ensure that the custom components or masked subsystems used in your circuit are always up-to-date. Or, the other way round, if you are developing your own custom components you can use a library to ensure that changes you make to your component models are automatically propagated to a user's circuit upon loading. This exercise also explores the model reference feature, available starting with PLECS 4.5.

In this tutorial you will learn the following:

- How to create a new user library and add a custom component to it
- How to structure a library
- How to update a library component
- How to break a library reference and customize it
- How to create a library within a library
- How to create a model reference

Before you begin ensure that you have the reference files that you can compare with your own models at each stage of the exercise.

2 Create a New Library in PLECS

Any model file in PLECS Standalone can be used as a library file. In this section, we will create a new library file for the PV panel custom component developed in the previous exercise, titled "Creating Custom Components with the PLECS Subsystem Block". This custom component is provided in the attached reference files as `pv_panel_lib.plecs`.

To create a new component library in PLECS Blockset, the PLECS Library block from PLECS Extras library must be copied into a Simulink model or library. Then, the Simulink model must be named (i.e. saved) before components are copied from the component library. For more details, refer to the tutorial video titled "*Creating a User Library Component in PLECS Blockset*" on the Plexim website.

Your Task:

1 Locate the PLECS model `pv_panel_lib.plecs` in the reference files.

2 Open the **PLECS Preferences** dialog and go to the **Libraries** tab.

3 Add a new search path by clicking on the **+** icon on the upper-left side. Browse to the directory that contains the file `pv_panel_lib.plecs` and click **Open**. The search path should now be configured.

4 Configure the **User libraries** by clicking on the **+** icon on the lower-left side. Select the file `pv_panel_lib.plecs` and click **Open**. A new user library should now be created.

5 Modify the **Description** of the newly created user library as desired, e.g, to "PV Panel Library".

6 Click **Apply** to apply the changes and **OK** to close the dialog.

The newly created library "PV Panel Library" should now be visible in the PLECS Library Browser, as shown in Fig. 1. This new user library can now be used similar to the standard library components.

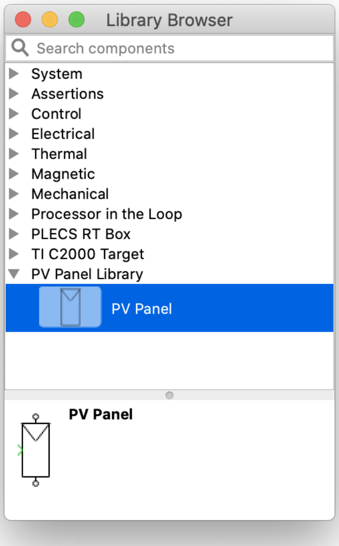


Figure 1: PLECS Library Browser with newly added user library

Note: If the checkbox **Show library link indicators** within the **PLECS Preferences + Libraries** dialog is checked, PLECS will display a small hollow curved arrow (↗) in the lower-left corner of each component that links back to a library, as seen in Fig. 4. A right-click on this link indicator opens a context menu that allows you to break the library link or show the original component in the library browser.

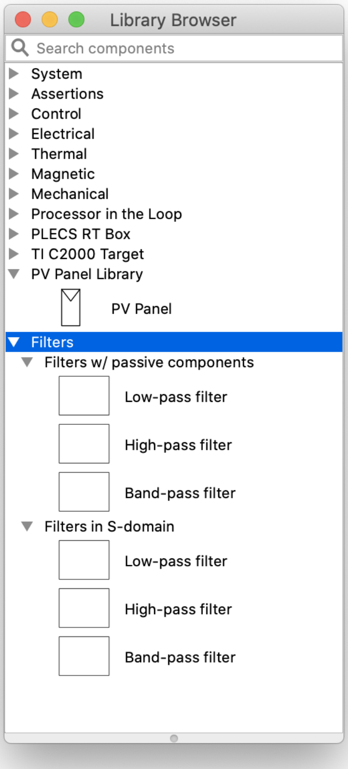
Structuring a Library

Subsystems can be used to create different sections inside a custom library. For example, in Fig. 2, a structured library called Filters is created by using two separate subsystems, each containing different custom components. This structured library model is provided in the attached reference files as `filters_lib.plecs`.

3 Creating a Library Reference

The next step is to use this library component in a PLECS model. In this section, we will use the PV Panel Library to build a full-bridge single-phase inverter.

When we copy a library component, either into a circuit schematic or a component library, PLECS automatically creates a reference component rather than a full copy. You can modify the parameters of the reference component but you cannot mask it or, if it is already masked, edit the mask. You can recognize a library reference by the string “(link)” displayed next to the mask type in the dialog box, the string “Link” displayed in the title bar of the underlying schematic windows, or with PLECS 4.5 or newer, by showing library link indicators, as seen in Fig. 4.



Library Browser

Search components

- System
- Assertions
- Control
- Electrical
- Thermal
- Magnetic
- Mechanical
- Processor in the Loop
- PLECS RT Box
- TI C2000 Target
- PV Panel Library
    - PV Panel
- Filters
    - Filters w/ passive components
        - Low-pass filter
        - High-pass filter
        - Band-pass filter
    - Filters in S-domain
        - Low-pass filter
        - High-pass filter
        - Band-pass filter

Figure 2: PLECS Library Browser with a structured user library, "Filters"

Your Task:

1. From the reference files, open the PLECS model `component_libraries_start.plecs`. The model contains the circuit shown in Fig. 3 within the subsystem labeled "Full-Bridge Inverter".
2. Drag and drop the PV Panel block from the custom library onto the schematic window.
3. Complete the PV inverter circuit shown in Fig. 4, using blocks found in the Passive Components category to build the output filter and the load.
4. Complete the controller circuit to generate pulse width modulation (PWM) signals, as shown in Fig. 4.
5. Set the parameter values as displayed in Fig. 4; leave all the other component parameters as their default values.
6. Complete the model by adding two Probe blocks to monitor the voltage, current and power measurements of the PV inverter and the load.
7. Open the Simulation Parameters window and set the simulation **Time Span** to 0.4 and change the **Solver** to `RADAU`. Leave the other simulation parameters as their default values.

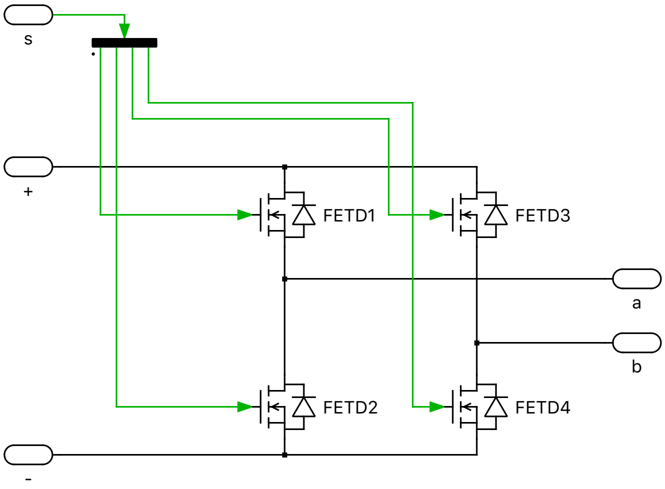


Figure 3: Single-phase full-bridge inverter subsystem

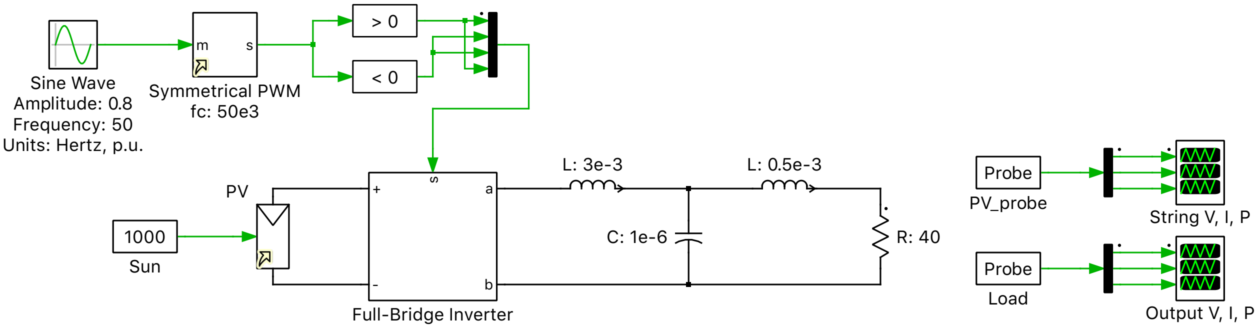


Figure 4: Complete PV inverter circuit

At this stage, your model should be the same as the reference model, `component_libraries_1.plecs`.

Note: The reference component links to the library component by its full path in effect at the time the copy is made. If PLECS is unable to resolve a library reference it highlights the reference component and issues an error message. To fix an unresolved library reference, delete the reference component and make a new copy of the library component.

4 Updating a Library Reference

Library references are resolved upon loading a circuit. Afterwards, any changes that are made to a referenced library component are automatically propagated to the referencing components after starting the simulation.

In this section, we will update the PV Panel user library component.

Your Task:

1 Locate and open the PV Panel user library component file `pv_panel_lib.plecs` in the reference files.

2 Select the subsystem, then choose **Edit mask...** from the **Subsystem** submenu of the Edit menu or the block's context menu by right-clicking.

3 From the **Dialog** tab, add a new parameter by clicking on the **+** icon on the upper-left side; leave the `prompt` as default, click **OK** to apply the changes and close the dialog, and save the model.

Now when you go back to the model that you built in Section 3, and open the PV Panel subsystem, you should see the new parameter tab listed in the block parameters dialog

5 Breaking a Library Reference

It is possible to break the link between a library reference and the library component. The reference then becomes a simple copy of the library component; changes to the library component would no longer affect the copy or vice-versa.

Your Task:

1 Reopen the model that you built in Section 3. Select the PV Panel subsystem, then choose **Break library link** from the **Subsystem** submenu of the Edit menu (or the component's context menu). The link between the PV Panel subsystem in the model and its library component should now be broken.

2 Select the PV Panel subsystem again, then choose **Unprotect** from the **Subsystem** submenu (or the block's context menu). This should allow you to make changes to the subsystem.

3 Then choose **Edit mask...** from the **Subsystem** submenu of the PV Panel subsystem (or its context menu), and delete the new parameter you added in Section 4 from the **Dialog** tab, then click **OK** to apply the changes and close the dialog.

If you drag and drop the PV Panel block from the custom library onto the schematic window, and compare the block parameters with the existing subsystem, you will notice that the changes you made did not affect the library component.
At this stage, your model should be the same as the reference model, `component_libraries_2.plecs`.

Note: It is also possible to break the links to all user-defined libraries in a model, for example when sending a model to another PLECS user who does not have all the library files that the model depends on. This can be done by selecting **Break all external links...** from the **Edit** menu. Library links to the PLECS component library are not affected by this functionality.

6 Library Within a Library

It is possible to create a new library component that contains an existing custom library component. For example, a new custom component titled “Solar Power Generation System” is built using the PV Panel user library component, as shown in Fig. 5. This component can then be structured to be within the same section as the original PV Panel user library component, as shown in Fig. 6. The updated custom component model is provided in the attached reference files as `pv_lib_final.plecs`.

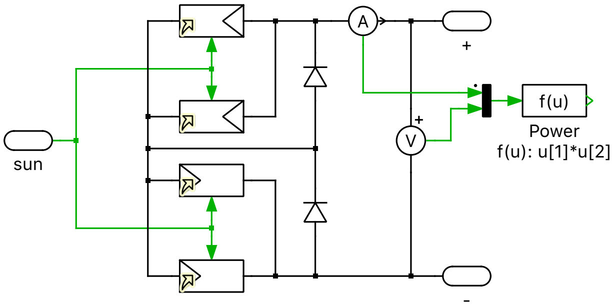
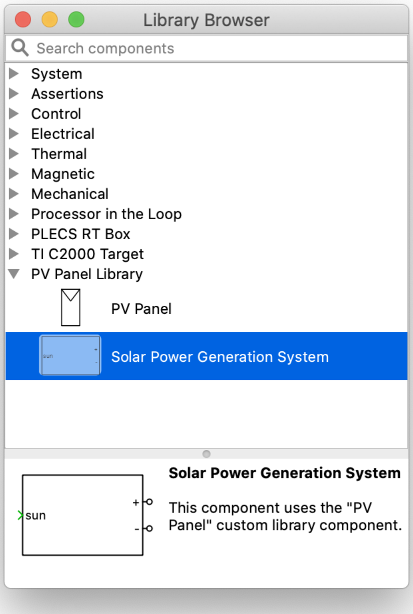


Figure 5: Solar Power Generation System component using the PV Panel component



Solar Power Generation System

This component uses the "PV Panel" custom library component.

Figure 6: PLECS Library Browser with the updated user library containing a library component inside

In this section, we will build a new library component that contains an existing custom library component.

Your Task:

1 Open the PLECS model titled `pv_panel_lib.plecs` located in the reference files, that you used in Section 2.

2 Drag and drop the Subsystem component onto the schematic window.

3 Within the schematic of this subsystem component, build the circuit shown in Fig. 5, using the PV Panel library component.

4 Double-click on all the PV Panel components, and set the "Number of modules" parameter as "n".

5 Go back to the parent subsystem, rename it as desired, then choose **Create mask...** from the **Subsystem** submenu of the Edit menu (or the component's context menu).

6 From the **Dialog** tab, configure the **Dialog parameters** by clicking on the **+** icon.

7 In the parameter window enter "Number of modules per each panel" and in the Variable box, give this parameter the variable name "n".

Probe Signals

In order to monitor a signal from within a library component subsystem, you should define the custom component signals of interest under the **Probes** tab of the Mask Editor dialog, as shown in Fig. 7. Adding custom probe signals to a subsystem mask makes it possible to directly monitor the key values inside a subsystem using the PLECS Probe block. A step-by-step explanation is provided in the tutorial titled "Creating Custom Components with the PLECS Subsystem Block".

As an example, to monitor the DC voltage inside the Solar Power Generation System component, a mask signal should be defined for this component by linking the internal Voltmeter and selecting the Voltage signal, as shown in Fig. 7. Then, when this component is copied into a new model, the block itself can be probed such that one can monitor the DC voltage without opening up the subsystem. Any quantity that is measured or calculated inside the subsystem can be added to a custom probe signal list in the subsystem mask. In this example, the probe signals that are added to the Solar Power Generation System subsystem mask are output voltage, current and power.

Your Task:

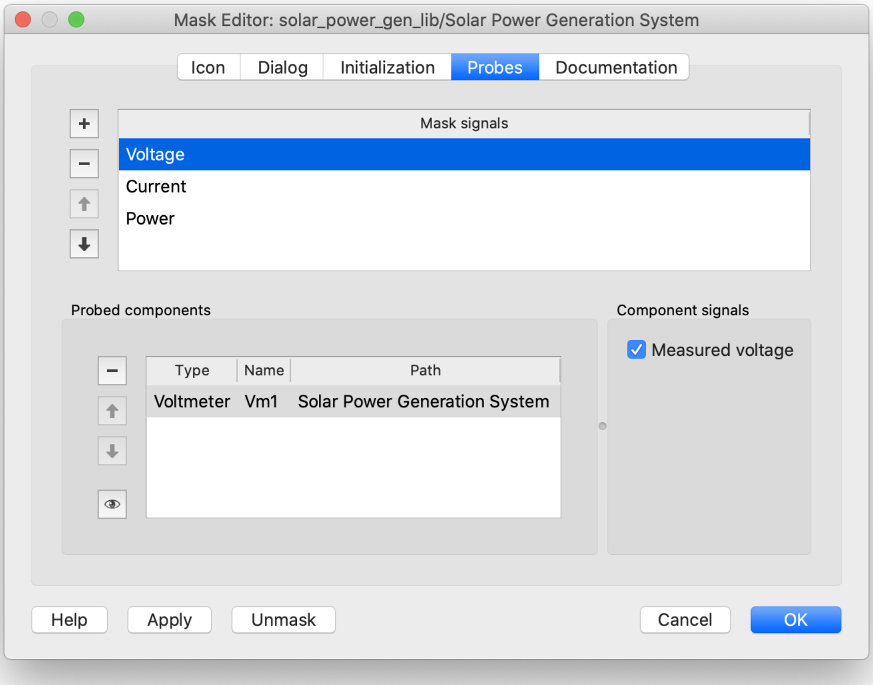
8 From the **Probes** tab, define the mask signals as shown in Fig. 7. Then click **OK** to apply the changes and close the dialog.

9 Save the model. The newly created component should now appear in the same section of the library as the PV Panel component, as shown in Fig. 6.

At this stage, your model should be the same as the reference model, `pv_lib_final.plecs`. And the newly created library should now be visible in the PLECS Library Browser, as shown in Fig. 6.

Your Task:

10 Next, update the model that you built in Section 5, by replacing the PV Panel library component, with the newly created Solar Power Generation System component.



Mask Editor: solar_power_gen_lib/Solar Power Generation System

Mask signals

- Voltage
- Current
- Power

Probed components

Type	Name	Path
Voltmeter	Vm1	Solar Power Generation System

Component signals

☑ Measured voltage

Figure 7: Create probe signals using the Mask Editor

At this stage, your model should be the same as the reference model, `component_libraries_3.plecs`.

Note:

- When you make any changes to a lower-level library component, for example, the PV Panel component, you must restart your PLECS session for these changes to reflect in the higher-level library component, in this case the Solar Power Generation System component.
- In a self-referencing library, such as the the Solar Power Generation System component, make sure that the referenced component, in this case the PV Panel component, is created first (so that it will appear first in the model parser). If the referenced component is created later (then it appears later in the model parser), then PLECS will not be able to find it, and therefore will generate an error.
- Instead of creating a structured library, one could also create a new model (and therefore a new library section) with the Solar Power Generation System component.

7 Model Referencing

Starting with PLECS 4.5, there is a Model Reference component. Note that this block is included only in the PLECS Standalone component library.

A Model Reference allows you to reference a subsystem from another model so that when the original subsystem is changed, the reference is updated automatically. This concept is similar to library links, with the difference that the referenced model file path can be specified relative to the referencing model.

To define a model reference, double-click on the Model Reference block to open the Model Reference dialog, then drag the subsystem that you want to reference into the dialog and release your mouse once a small solid curved arrow appears. You can also drag the subsystem of interest directly onto the Model Reference block in the schematic. The labels **Model file** and **Subsystem path** will be updated to reflect the subsystem that is referenced. By default, the referenced model file is specified with a path relative to the referencing model, but you can change this with the **Model file reference selector**.

When you click **OK** to apply the settings, the Model Reference block is replaced with a copy of the referenced subsystem that is marked with a small solid curved arrow (⬈) in the lower-left corner of the component icon. A right-click on this arrow opens a context menu that allows you to break the link, to edit the model reference, or to show the original subsystem.

In this section, we will create a Model Reference to the subsystem labeled “Full-Bridge Inverter”, shown in Fig. 4.

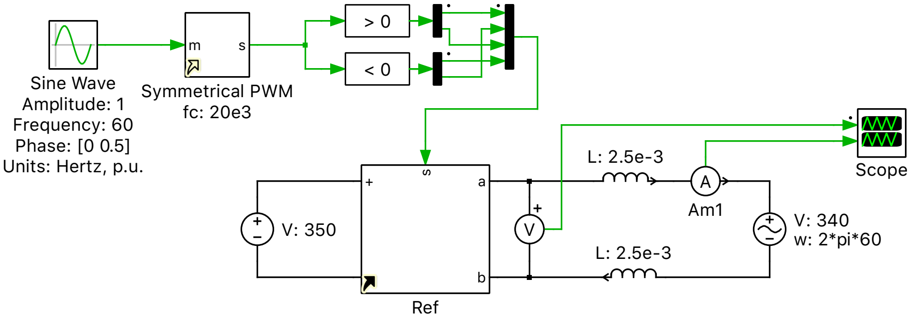


Figure 8: A circuit using the Model Reference block

Your Task:

1. Open a new PLECS model.
2. Complete the circuit shown in Fig. 8, except for the subsystem labeled “Ref”; we will create a model reference for this subsystem.
3. Drag and drop a Model Reference block from the System component library onto the schematic window.
4. Associate the Full-Bridge Inverter subsystem from the model that you built in Section 6 with the Model Reference block. A Model Reference parameter window should pop-up.
5. Modify the **Model file reference** parameter if desired. If the new PLECS model is in the same folder as the PLECS model with the parent subsystem, then choose `relative path`. Otherwise, choose `absolute path`.
6. Click **OK** to apply the changes and close the dialog.
7. Complete the electrical circuit and save the model.

At this stage, your model should be the same as the reference model, `component_libraries_4.plecs`.

Note:

- To view the source subsystem associated with the Model Reference block, right-click on the small solid curved arrow (↗), and click on **Show source component**; or select **Edit model reference**, and click on the eye (👁) icon.
- Breaking a model reference is similar to breaking a library reference, as defined in Section 5.

8 Conclusion

In this tutorial you built a PLECS model using custom components from a user-defined library. Libraries enable you to ensure that the custom components or masked subsystems used in your circuit are always up-to-date. Or, the other way round, if you are developing your own custom components you can use a library to ensure that changes you make to your component models are automatically propagated to a user's circuit upon loading. This exercise also explored the model reference feature, available starting with PLECS 4.5.

Revision History:

Tutorial Version 1.0	First release

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	`info@plexim.com`	Email
	`http://www.plexim.com`	Web

PLECS Tutorial