

# **How to Create Footprints in OrCAD Layout**

**Department of Electrical & Computer Engineering**

**Michigan State University**

**Prepared by: John Kelley**

**Revision: 4/13/00**

## **Executive Summary**

This application note outlines a process through which footprints may be created using OrCAD Layout software available in the Electrical and Computer Engineering Department at Michigan State University. This document is a companion application note to the application note “How to Create a Printed Circuit Board (PCB),” which outlines the general procedure for creating a fully functional PCB. This application note describes the process using the example of creating an optical sensor footprint for use in an optical sensor circuit PCB.

## **Table of Contents**

<b><u>Focus</u></b>	<b><u>Page</u></b>
1. Procedure	2
2. Appendix	6

## 1. Procedure

A good way to begin learning OrCAD Layout is through the Layout toolbar menu path *Help→Learning Layout*. This tutorial does a reasonably good job of explaining the basics of Layout. In the following portions of this application note, it has been assumed that readers have a basic familiarity with Layout.

Let's begin with a generic Layout board. Start the Layout software and choose *File→New*. Next it will ask you which template file you would like to use. Click on Cancel. This will let you begin with a blank board. Choose *File→Library Manager*. When you create a new footprint, you have two basic options, either you start with nothing, or you edit an existing footprint. To edit an existing footprint, select that footprint from the list on the left side of the screen. The footprint will appear in the work area to the right. After you have completed your edits then select the "Save As..." button to save the edited footprint with a new name in your custom library.

To help explain the process of creating a new footprint, I will guide you through the process I used to create the footprint for an optical sensor component I used in my optosensor demonstration circuit.

First, I chose the "Create New Footprint..." button from the Library Manager screen and named my footprint "HOA2003". Then I zoomed out to get a larger drawing area. Next I choose the "Save as..." button and then the "Create New Library..." button. I created a new directory called "library" that was separate from the directory where my layout and capture files were stored. I named my new Library file "Opto" and saved my footprint in it with a name of "HO2003". After that, I had a new library called "Opto" containing the footprint "HOA2003".

Using the physical dimension shown in Fig 2, I began by removing the default pin and drawing the outline of the sensor. With the Pin Tool selected, I clicked on the pin and pressed delete to erase the default pin. Down in the far lower left corner of the screen, you will see your cursor location readout that is formatted as [horizontal distance, vertical distance]. This cursor readout is measured from the origin of the drawing space, which is shown as a circle/crosshairs.

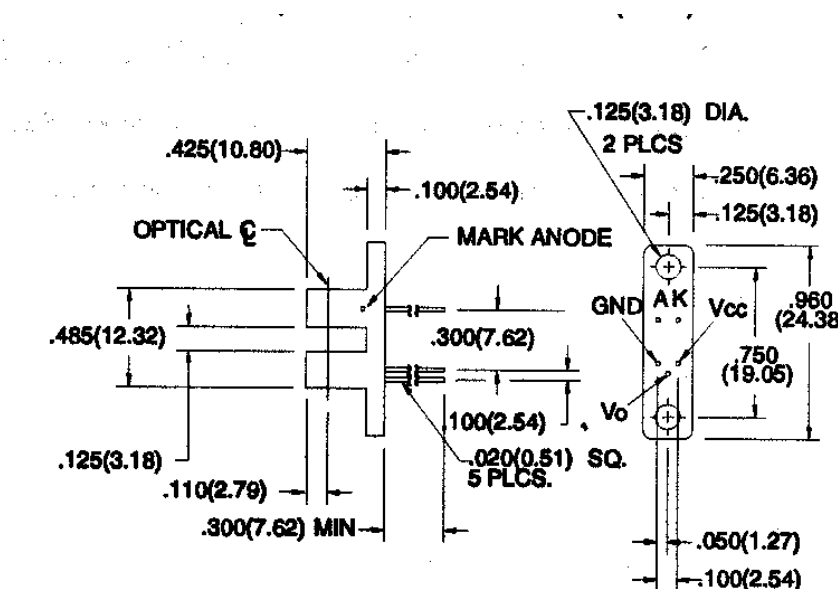


Fig 2: HOA2003 Transmissive Optoschmitt Sensor physical dimensions.

If you place the cursor directly overtop of this origin the cursor locations should read [0,0]. I used this origin as the lower left corner of my footprint. To draw the outline of the sensor, I selected the Obstacle Tool, right clicked anywhere in the drawing space and selected “New...” I then clicked and held the left mouse button at the origin of the screen and dragged the cursor until it read [950,250] or .95 inches in the horizontal direction, and 0.25 inches in the vertical direction. Note that I went to 950 instead of 960 as shown on Fig 2. I did this for ease of drawing and because I knew the .01 was not important in my application. Next, I double clicked on the outline I had just drawn. This brought up a dialogue box where I chose the Obstacle Type to be Place outline and the Obstacle Layer to be the Global Layer and then I chose OK. To end the command, I right clicked in the drawing space and selected the End Command.

Next, I selected the Text Tool and removed all of the text except the green HOA2003. This green ASYTOP layer did not appear on my final PCB. I left it there as a reference while I worked on the PCB in Layout. However, I wanted it smaller; so, I double clicked on it and brought up its properties. I then set the Line Width to 4 and the Text Height to 30. I then chose OK and moved the text to the middle of the footprint outline. At this time my footprint looked like the image depicted in Fig 3.

Next, I put the five pins in the center. I began by selecting the Pin Tool, right clicking in the drawing space and selecting New... Then I left clicked to place the first pin close to where it

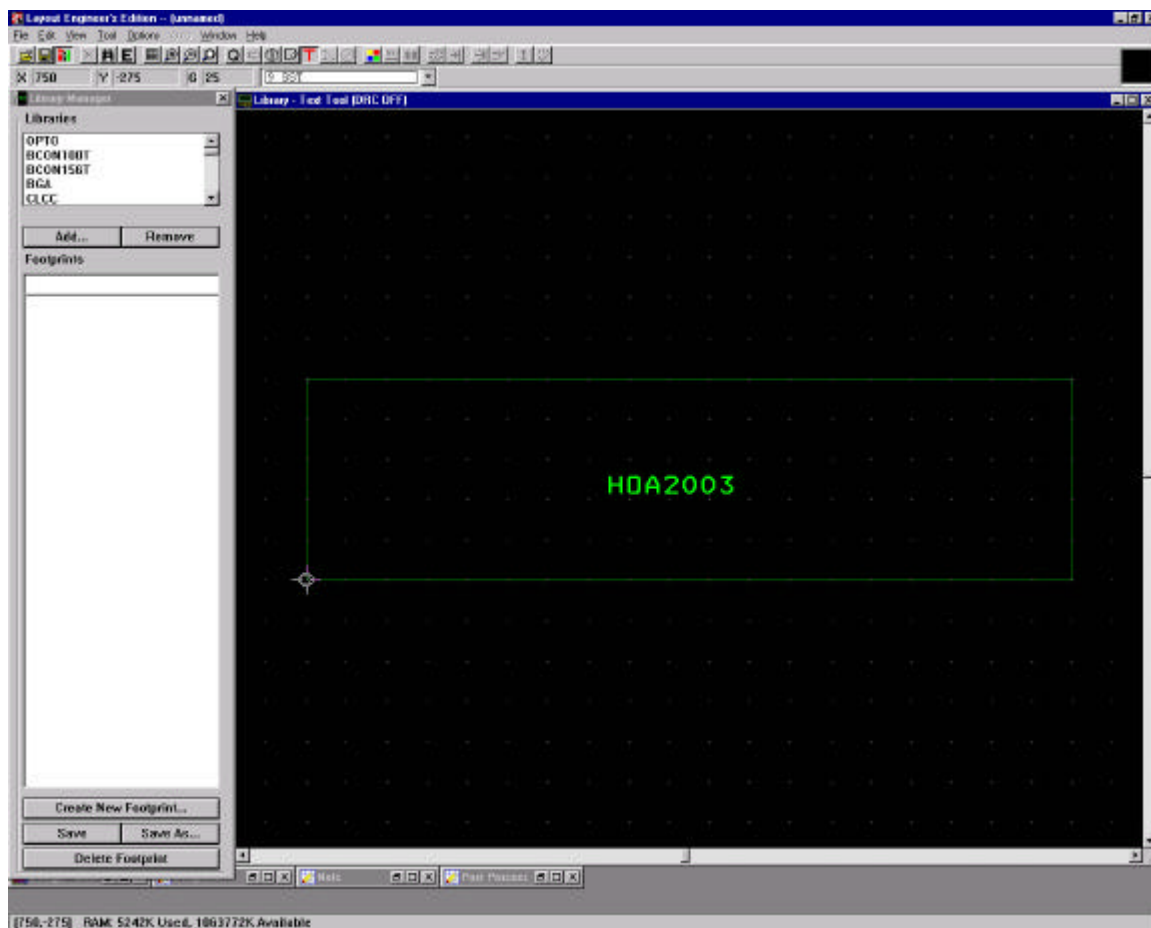


Fig 3: Partially completed Sensor footprint.

belonged. I repeated this process until I had 5 generic pins inside the footprint outline. I did not like the default names for the pins, so I double clicked on the first pin to bring up its property sheet and changed Pad Name to A. I repeated this for the remaining pins, naming them K, GND, VCC and VO.

The characteristics of any pin in Layout are defined by something called the “padstack.” I knew from working with this sensor that the pins are the general size pins such as you would use in a protoboard. First, I double clicked on one of the pins to bring up a “properties” dialog box, where I could see the name of the padstack that was currently being used. Doing this, I saw that it was T1. Next, I selected *View→Database Spreadsheets→Padstacks*. This brought up a spreadsheet containing all of the padstack information. I did not like the name T1, so first I double clicked on T1 and brought up its properties and renamed it PIN. Looking at the data for PIN, I saw that for each layer PIN was defined to have a certain shape, width, height, x offset, and y offset. For most cases, I left the x offset and y offset at 0. Hence, these two parameters were of little interest for this application. What was of interest though was the shape and size used in each layer. The layers that make it a through-hole pin are the DRLDWG and DRILL layers. These two layers defined that this pin would have a round hole drilled through the board of such and such width and height. (Note: for round holes width = height, thus I referred to it generically as size) In this case, the drill size is 38, or .038 inches. However, in talking with Brian in the ECE Shop, I learned that their standard drill size for a DIP or similar component pins was 40 not 38. (Note that the drill size for components such as resistors and capacitors is also 40.) So, I changed all the DRILL and DRILLDWG layers that say 38 to 40 by double clicking on the 38 and entering the new value of 40. The other important part of the padstack for our uses was the size of the TOP and BOTTOM layers. These layer settings defined the amount of copper that would be around the drilled hole. These layers had a default value of 62. That meant that around the drilled hole would be an area of copper of size 62 on the top and the bottom. In working with the ECE shop equipment, I knew that a larger copper area would be better to facilitate soldering. Thus, I double clicked on the size columns in BOT and then TOP and changed the value to 95. Those four layers were the only ones of interest for me in doing a two-sided, through-hole board. Leaving the other layers with the default settings may not cause any problems, but it would represent extra information that might clutter the display and get in the way. Thus, I double clicked on the Pad Shape column of the unused layers and changed the layers to “undefined.”

Now that the pins themselves were correct, I merely had to move them so that they were in the correct location. To determine the correct location, I began by finding the lengthwise or horizontal distance by referring back to Fig 2. I saw that I needed 0.3 inches between the two rows of two pins. I also knew that I wanted these for pins to be symmetrical about the vertical axis of the footprint symbol. My footprint symbol was 950 in length, thus the horizontal distance for pins A and K is  $950/2 - 300/2 = 325$  and the horizontal distance for pins GND and VCC was  $950/2 + 300/2 = 625$ . Next, I noted that Vo pin was 0.1 inches beyond GND and VCC. Thus, Vo was 725 milli-inches from the origin. Now, for the widthwise or vertical measure, I again referred back to Fig 2. I saw that there was 0.1 inches in between A and K and 0.1 inches in between GND and VCC. These were now symmetrical about the horizontal axis. I knew that the total width of the symbol was 250. Thus A and GND were at a vertical height of  $250/2 + 100/2 = 175$ . By the same reasoning, K and VCC were at a vertical height of  $250/2 - 100/2 = 75$ . Vo was right on the centerline, or  $250/2 = 125$ . In summary, the locations of the pins were chosen as follows:

**A=(325, 175); K=(325, 75); GND=(625, 175); VCC=(625, 75); Vo=(725, 125 )**

Now that I had the location, I selected my pin A and tried to move it to (325, 175). However, I discovered that the snap (or grid) was set so coarsely that I could not place the cursor at (325, 175). After I looked it up in the help index, I discovered that the way to change this parameter setting was to choose *Options*→*System Settings*.... This opened a dialog box where I changed the grid spacings. For moving the pins, I wanted to change the Place grid specifications. Looking at my coordinates, I realized that 25 would allow me to reach all of my coordinates, so I changed the setting to 25.

Once again I selected my pin A and this time, moved it to (325, 175). Then in succession, I selected and moved each of the remaining pins to the correct coordinates.

This completed my symbol. This new footprint symbol is illustrated in Fig 4. Now this footprint may be used in my layouts.

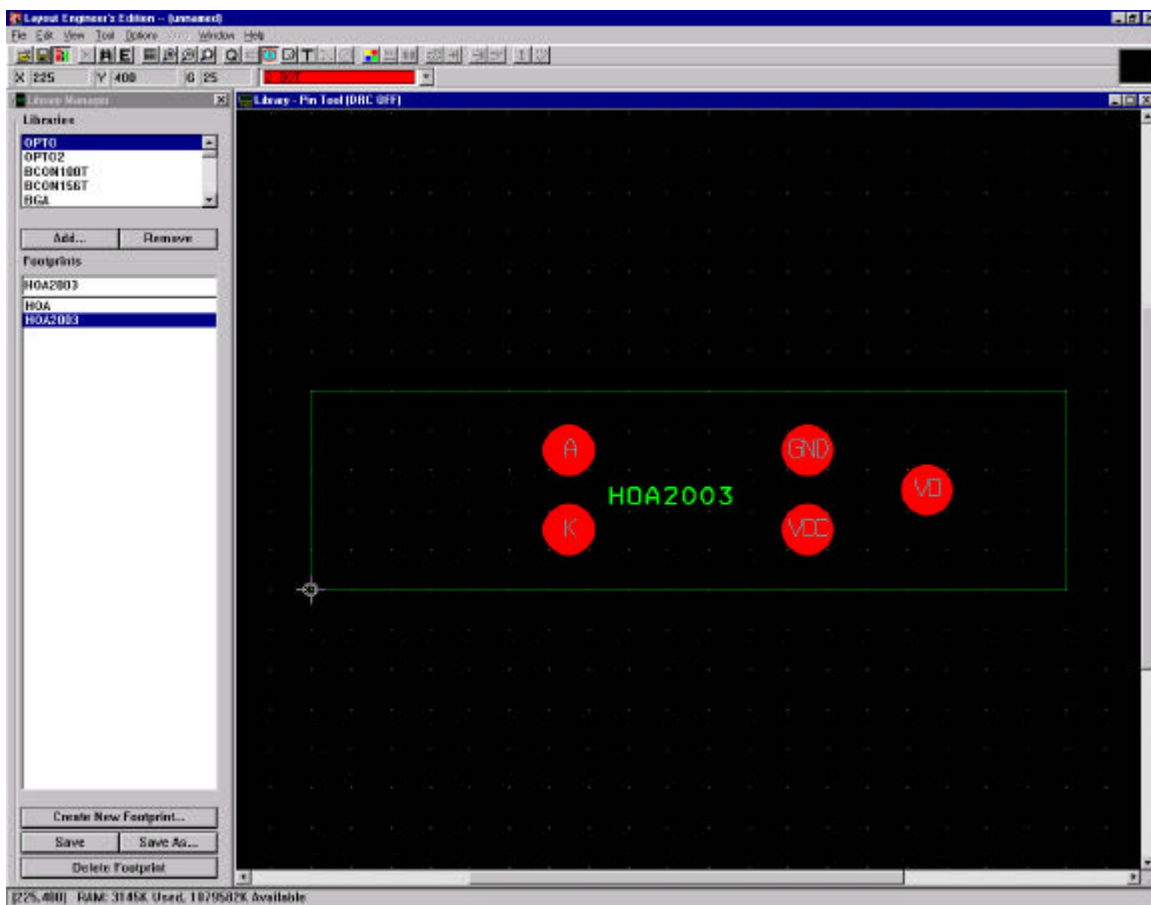


Fig 4: Completed Sensor footprint.

## 2. Appendix

### 2.1 – Frequently Asked Questions

#### *What is a padstack?*

- A padstack is a collection of information defining how a pin connects to the PCB. This padstack includes information about every layer in the board. For example, a dip chip would have a padstack defined for its pins. This padstack would define the size of the copper area around the pin on the top and the bottom of the PCB. It would also define the size of the hole to be drilled through the board. ( Note: for more information on padstacks, refer to application note an\_pcb\_2 – “How to Create Footprints in OrCAD Layout”. )

#### *What is a footprint?*

- A footprint is a collection of information defining how a part or component will attach and be used on the PCB. For example, a dip chip footprint would define where each of its pin holes are to be drilled relative to each other.

#### *What is a pin?*

- Pin has several meanings. When I say pin, I generally mean “any connection to the PCB that may serve as a termination pt for a route”.

#### *What is a Gerber file?*

- A Gerber file is a file that contains the information from Layout necessary for the prototyping machine to mill, drill and cut the PCB. You have a Gerber file for each Layer of information about the board. For example, you have a TOP Gerber file which defines how to mill, drill, and cut the top of the board.

### 2.2 - Other sources of help

OrCAD menu item *Help*→*Help Topics...*

OrCAD Homepage : “<http://www.OrCAD.com/>”

OrCAD Design Network : “<http://www.OrCAD.com/odn/>”

Aspect Online : “<http://OrCAD.aspectonline.com/>”