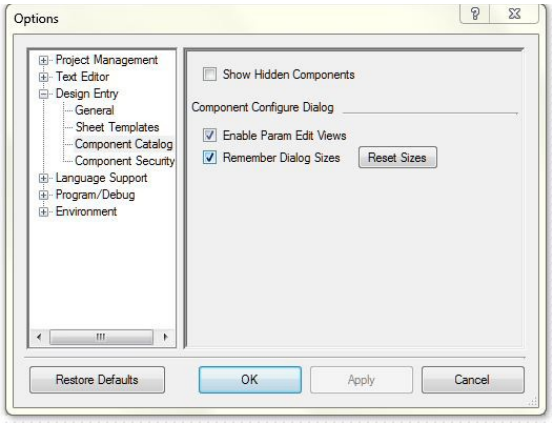
There are secrets and there are hidden secrets....

The first nut I had to crack was to have the number of I/O-pins for the row or column to be adjustable. Well, I like to read manuals but I could not get any clue out of Component Author Guide. So I asked for some help and I was directed to Brad Budlong at Cypress who instantly helped me. It turned out that there seemingly was no written documentation but a handful of videos each about 20 minutes long where some essential information is revealed. I remembered to have those videos already seen, but that was at a time I was not yet so deep involved into PSoCs and they seemed to be too complicate for me to grasp.

I suggest you to take your time and watch all three videos, get your development kits ready and follow the instructions and examples.

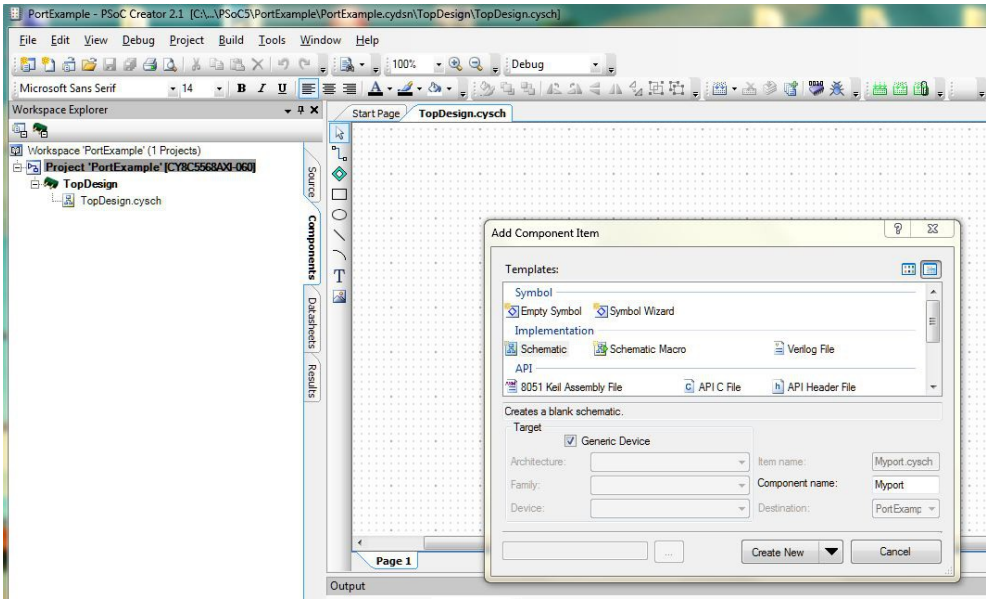
Here http://www.cypress.com/?id=1162&page=2 you'll find video numbers 110, 111 and 112 which are related to component creation and give you additional information not contained in the manual.

Brads advices were simple and easy to follow: first of all in the options dialog I had to set a property



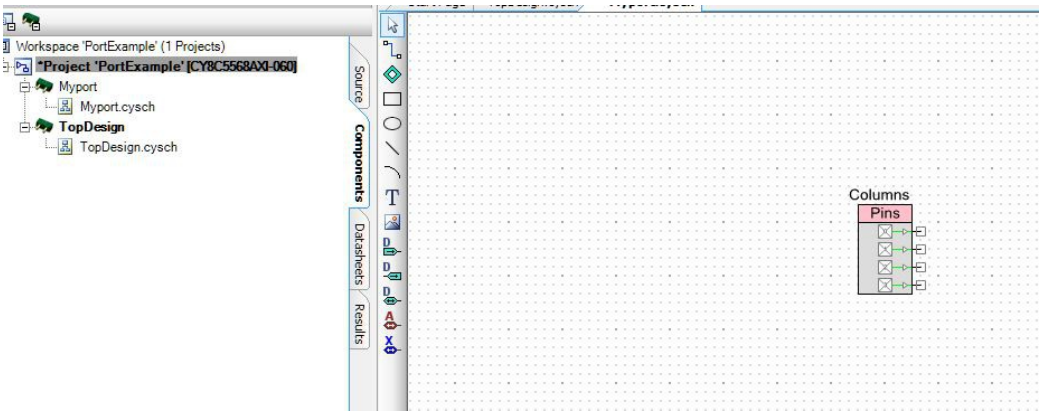
“Enable Param Edit Views” has to be checked.

The next step to create a new component is to switch to component view in the “Workspace Explorer” window. Click on that tab, right click on the project and select “Add Component Item”

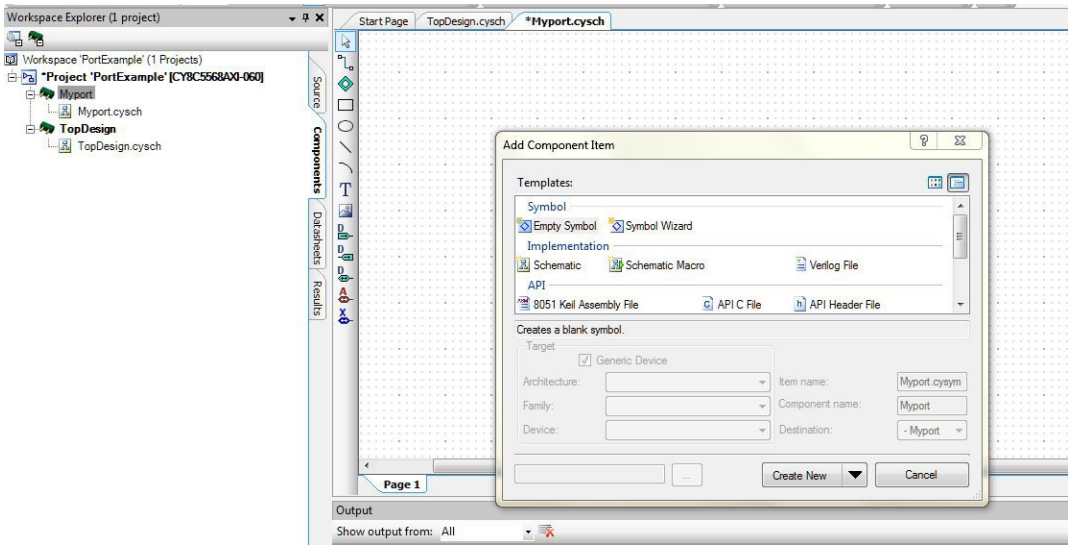


In the pop-up window select Implementation – Schematic and give your component a meaningful name (Myport). Finally click on “Create New” since that is exactly what we want to.

We get a smaller than usual sheet-window where we can now place all our needed components from which we are building our new one. Since I want to show you something special just drop an output pin on the schematic and configure it so that it has 4 pins and has a meaningful name. Should now look something like this

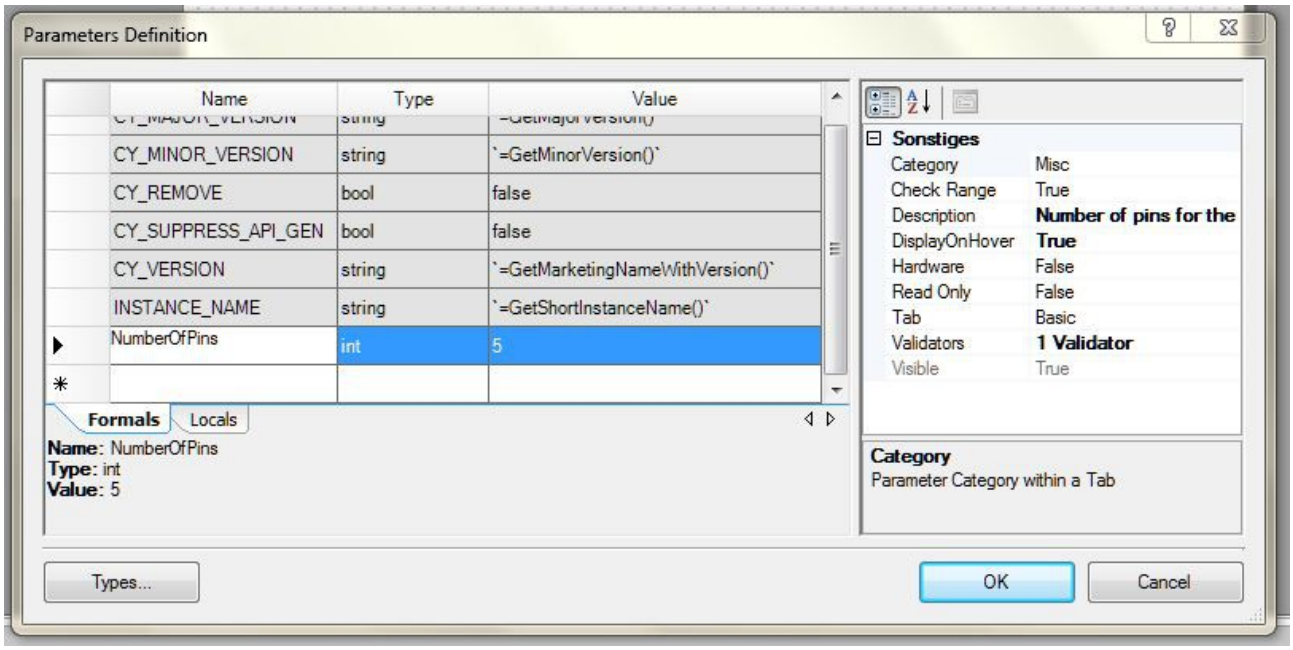


We now have to create a visual symbol for our component which is shown when we drop it on our project schematic. Right-click on our “Myport” component in the workspace explorer, select “Add Component Item” and now select “Empty Symbol”.



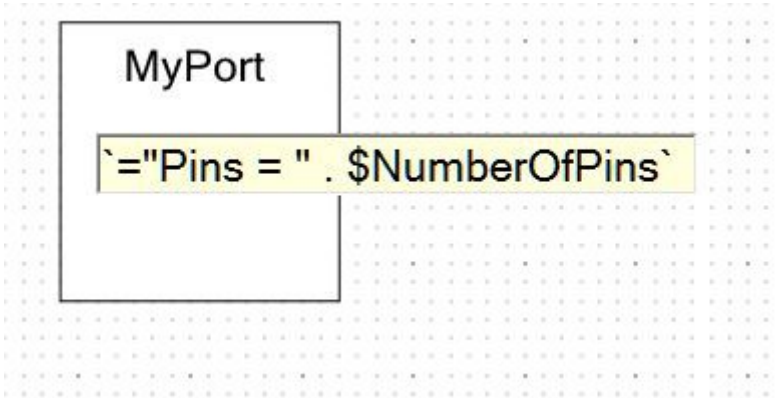
Click on “Create New” and you get an empty sheet again with a positioning – cross in the middle. Draw a rectangular around and put a headline on it.

Now we are going to give our component a “Parameter”: right-click on the sheet and select “Symbol Parameters”



Enter a new parameter name, add a description, a validator... all this description you can find in the “Component Author Guide” (from Help → Documentation) chapter 3.
So this is nothing really new, you always will have to go this way when creating new components.
Save your changes and re-open your component.cysch-file
Configure the pin-component, right-click on the Pins-tab. In the list select “Expression View”
This functionality is not documented (yet), you only find this in the videos!
Now comes the trick: Under “NumPins” we now put our parameter name preceded by a dollar sign “$NumberOfPins” (without the quotes). This will expand to the value we give our parameter later on. Save that.
Now we go back to our symbol and put another text-field right on it. Write into the text-field
`="Pins = " . $NumberOfPins`
Care for the backward single quotes, they are not accepted immediately and need another key press.

Save that.



Now you can switch back to the “Source” view by clicking on that tab in the Workspace Explorer window. In your component-window under “Default” you'll find your newly created component. Place that on your sheet and configure it. We do not have yet generated any APIs, but we can build that project already. Check the pins requirement of the component.

This was merely an exercise to share with you what I've found out / was told by Brad. The parameter substitution is an essential way of propagating the settings of a high-level component down to its lowest-level components.

I used this knowledge to create my KeyPad component and to specify the number of pins for the row and column matrix connections needed.

… to be continued

Bob