Hierarchical schematics and building blocks in KiCAD. Rev.A DRAFT-English

Dipl. Ing. Bernd Wiebus

May 26, 2010

Contents

1 Preliminary remark

I want to avoid the phrase module for building blocks at KiCAD, because in the KiCAD namespace modul means a footprint. So it could get missunderstand. But the phrase building block will meet the meaning also very well. The following informations will be related to EESchema: Build Version: (20100314 SVN-R2460)-final with Windows XP and EESchema build Version XXXXXXXXXXXXXXXXX with operating system Linux Debian Lenny. Because of the fast progress of the work of Jean-Pierre Charras and his co-autors, this information probably will be outdated.

2 How hierarchical schematics at KiCAD can used for rapid development of new schematics.

KiCAD supports hierarchical schematics. Those hierarchical schematics are originally used, to display complex schematics in a clearly arranged way, by dividing them to subsheets. Nevertheless they can used in a reversed manner to create new schematics in a very fast and efficient way.

As an Example: Many circuits use an input rectifier with capacitors, fuses and so on, often followed by a voltage regulator. to maintain a stable voltage output. If you have a big confusing schematic, it could be a good Idee,

to put this repeating groups in subsheets, called hierarchical sheets. At the original main schematic only a box with connections and a link to a subschematic remains.

If you have done this work creating a subschematic, you will easily able to reuse them when you creates new schematics, who needs a similar group of devices. So you will be very quick and save much time. Also you can reuse this group more than once at your schematic. You only put a link to the subschematic several times into your schematic. Consequently you would be able to create a main schematic, which only contains subschematics and the connections between them. Because KiCAD discriminates a device as a symbol at the schematic from a module (footprint) used at the board, the technologie would be not very impotrant. You have to draw the device at a symbol and will decide later, by using the programm CVpcb), which especial footprint and technologie you want. But for get running this smooth, you will have to pay attention to some details. Often you have to copy and rename a file by hand, due KiCAD at moment does not support working this way. Using KiCAD in this manner, is a method i stumbled accidentially about. KiCAD at moment is NOT intendet to be used that way, but perhaps in the future, there could be some tools and import/export features supporting the reusing of schematics.

3 Creating building blocks as hierarchical schematics at KiCAD.

You can get schematics for KiCAd somewhere, but a last, it has to exist. And so somebody has to do the job and create a schematic fitting to KiCAD and your requirement. Despite i assume you are well known in creating schematics for KiCAD, i will repeat somme steps in creating schematics, especiall for the reuse as building block.

A Start up KiCAD. Look at picture 1.

B Choose NEW, for creating a new projekt. Look at picture 2.

Create at new folder(named here: Versuch) a new project of your choice(named here: Versuch1.pro). The ending with .pro is essential to a KiCAD project file. Look at picture 3.

C After doing this, you will find the project Versuch1.pro with a not yet filled board Versuch1.brd assignet to the KiCAD projecttree. Look at picture 4.

D Start, to create the schematic, the programm EESchema. At startup, you probably will get some error messages, because the schematic Versuch1.sch already does not exist. Just quit them and go on. now you will get a new, empty schematic. look at picture 5. If you will save the schematic once and leave EEschema, the schematic will get created as Versuch1.sch. Additional schematics assignet to this project could be created from EESchema, if you choose at Files from the upper tool bar either NEW or SAVE ACTUAL SCHEMATIC AS.

E . Now draw the schematic as you are used at KiCAD. If you want to ust the schematic as building block, you have to create ALL connections to this schematic as hierarchical pins. I have a suspicion, that it could be VERY DANGEROUS, to use global names at building blocks. Think about using a global label GND for ground. Now think about using this building block at the same schematic several times, but different ground systems (like so often at power electronics). Then all your grunds will get tied together, and you will not see it at a glance, because it is somewhere hidden in the subschematics.

For creatin hierarchical labelse, use the button XXXXX at the right tool bar. You can see the complete building block ready to use as 317Regulator$_{B}uildingBlock_{M}odA-11042010.schatpicture6.ItisastandardvoltageregulatorwithaLM317, justasanexercisingobject.$

Look out for the file 317Regulator-BuildingBlock-ModA-11042010-cache.lib. It will be reated lately, if you save ALL schematics and leave EESchema. Perhaps it would be nice to have a button to force this. This -cache.lib is a symbol library created with all used symbols in the just closed schematic. It is important, if yo carry this bibliothek to another project, where perhaps an other library exists, whitout the symbols you have used. Restart EESchema and choose your newly created schematic. Remove ALL symbol librarys from the library list and put only the -cache.lib into the library list. Also save the path to this library as RELATIVE. (Damit KiCAD alles im entsprechenden Ordner sucht).

Now save all, also your project file. It iss essential, for migrating this building block schematic to other projects, to keep the schematic (.sch), the library cache (-cache.lib) and the projectfile (.pro) together. Ok, perhaps not the projectfile, but i am not really sure, and it would nice to keep it, if you want it some day.

4 Inserting building blocks into schematics.

4.1 Copying and renaming the files

Now you want to put the building block, you have just created, into another schematic. Here this schematic is called Test1.sch . Therefore Test1.sch has to be exist and so it has to be created at KiCAD in a well known way. Think about to use this 317regulator building block at this schematic Test1.sch three times. Therefore copy the schematic

317Regulator-BuildingBlock-ModA-11042010.sch and the symbollibrary cache 317Regulator-BuildingBlock-ModA-11042010-cache.lib into the projekt folder of your Test1 schematic or into a subfolder created for just this use. You will not need the projectfile of the building block. It would only convenient if you want to edit the original file of the building block. But you should only work with renamed copies of the original building block. Now RENAME the COPY of 317Regulator-BuildingBlock-ModA-11042010.sch and Block1-cache.lib. This renaming is for decoupling and keeping oversight. Just savety and transparency. And conserving the original file.

4.2 Creating the subschematics

You have to insert the building block three times into your main schematic. Choose from the right toolbar inserting hierarchical schematic, button XXXX. A window will pop up. Write as filename the filename of your building block, here Block1.sch. Choose as schematic name a random, at least only once appearing name. Here is used Regulator317-I. Quit with OK. Probably you will get a warning, that the block is already existing, and a question, wether its data should be read. Quit this with YES. Repeat this three times. Always choose as file name Block1.sch, but choose different schematic names, here are used Regulator317-II and Regulator317-III. Save ALL.

Add the symbol librarycache (Block1-cache.lib) to the library list. You can use a hierarchical schematic either by a left mouse doubleclick into it or by a right mouse single click into it and choosing edit schematic or just use the hierarchical subsheet browser. Button XXXX at the top tool bar. You should see the subschematic just like you would see the original building block schematic.

4.3 Recapitulation of creating the subschematic and changing the sequence

The use of building blocks is simply the use of hierarchical sheets with prefabricated schematic files. Therfore you have to link to this existing files at the point you create (or use) this subschematic. Also you have to link to the symbol cache of this building block at the library list. To change the sequence, at last linking building blocks to existing hierarchical schematics, is a little bit complex, but is needed at the case we have to edit and chance existing schematics. Therefore create the hierarchical subschematic at first with a random name of your choice. Choose SAVE ALL SCHEMATICS from the upper tool bar. This will create empty subschematics with the names you have choosen bevore. It is equal to the situation if you have an existing schematic with not empty subschematics. Now close EESchema. Erase the subschematicfiles with a filebrowser and replace them with your wanted subschematics. Rename the subschematic names fitting to your choose names. Do not forget to add the new symbol-cache or even symbol caches at the library list.

If you forgot this, you will perhaps get only questionmarks instead of device symbols.

5 Interconnect the building blocks.

At the main schematic, go to the right tool bar and choose PLACE/IMPORT HIERARCHICAL PIN. Buitton XXXX. Now klick left into a choosen subschematic box. You will get a hierarchical label as a pin, which you can move with the mouse and place somewhere. Look at picture XXXX. You can repeat this, until all hierarchical labels of the subschematic are placed inside the subschematic box. Look at picture XXXX. You can connect this hierarchical pins to other hirarchical pins or other device pins of the schematic like regular pins. Look at picture XXXX.

6 Allocating of footprints and values.

If you have created two or even more subschematics who are linked to the same building block files, this will be no problem at first. The annotation is able to keep two or more subschematics separate. Devices at the same position at the subschematic will get correct counted. Also they will be noted correct with this reference numbers to the schematic. There you cann allocate them footprints at CVpcb. Devices with different reference numbers can also allocated to different footprints/modules. But you should think about it in detail, if you want to do it, because there are several drawbacks. Because all these subschematics are linked to the same building block schematic, and there can noted only one value, all devices of different reference numbers and at the same position at the subschematic will get the same value, even if they get different footprints. It depends from your personal behaviour about value and footprint at the BOM, wheter this is good or bad. Different footprints with different technologies but same VALUE propertys should be also different by the VALUE. As an example, VALUE could not only be 10k, but more 10k/0805 or 10k/THT/0,3W/RM10m. The consequentest and cleanest solution of this problem (if you consider this as a problem) would be to make a copy of the building block with a different name, if there is only a small difference between this two subschematics, may at value or at footprint. Then link different subschematics to diffenent building block files. Now the subschematics will not only differ by value or footprint, but also by the building block, and therefore you can allocate different values to different devices at different files.

7 Example Project

The example project is located in the folder Experimentalprojekt23052010. Inside this folder is a subfolder named BuildingBlocksExperimental, which contains the original building blocks. They are named VoltageRegulatorBuildingBlock.sch, VoltageRegulatorBuildingBlock.pro, VoltageRegulatorBuildingBlock-cache.lib, VoltageDetectorBuildingBlock.sch, VoltageDetectorBuildingBlock.pro and VoltageDetectorBuildingBlock-cache lib. The main schematic is in the Experimentalprojekt23052010 as UnderVoltageDetector24V-2Group Experimental.sch. The original building block VoltageRegulatorBuildingBlock.sch is renamed as 12VVoltageRegulatorBuildingBlock.sch and 5VVoltageRegulatorBuilding-

Block.sch. UnderVoltageDetectorBuildingBlock.sch is a hierarcical subschematic containing 5VVoltageRegulatorBuildingBlock.sch and VoltageDetectorBuildingBlock.sch, just showing, that this all can done in a cascaded way. So the main schematic UnderVoltageDetector24V-2Group Experimental.sch. It is made by 12VVoltageRegulatorBuildingBlock.sch and two UnderVoltageDetectorBuildingBlock.sch, but UnderVoltageDetectorBuildingBlock.sch itsself is made by 5VVoltageRegulatorBuildingBlock.sch and VoltageDetectorBuildingBlock.sch. Also there sometimes are additional devices.

8 DANGER

Because this all are features not yet intended by KiCAD, be careful by doing this. Work only with copys of the original files. Copying and renaming manually is not very dapper, but the only way to do it. But you will get in trouble if you make mistakes. Especial if you manage it to make a subschematic containing itsself as a subschematic. :O)

Do not blame me for errors, its all experimental.

Hope for KiCAD tools in managing this in a save way.

Bild 1: Nach dem ersten Start von KiCAD.

Bild 2: Anlegen eines neuen Projektes.

Bild 3: Namensnennung des neuen Projekts.

Bild 4: Nach dem anlegen des Projekts.

Autor: Bernd Wiebus dl1eic - at 26. Mai 2010. Uedem/Germany
GNU
Dipl. Ing. Bernd Wiebus
Weezer Str. 5
47589 Uedem
Germany
Tel. 02825-9399977
Tel. 0162-6157950 (mob.)
e-mail: bernd.wiebus@gmx.de
dl1eic@darc.de