Kicad

Reference manual

Copyright

Contributors

Jean-Pierre Charras, Fabrizio Tappero.

Feedback

Please direct any comments or suggestions about this document to the kicad mailing list: *https://launchpad.net/~kicad-developers*

Acknowledgments

None

Version

2013, may 31.

Note for Mac users

The KiCad support for the Apple OS X operating system is experimental.

Table of Contents

1 - Introduction

1.1 - KiCad

KiCad is an open-source software tool for the creation of electronic schematic diagrams and PCB artwork. Beneath its singular surface, kicad incorporates an elegant ensemble of the following stand-alone software tools:

- Kicad : project manager
- Eeschema : schematic editor
- CvPcb: footprint selector
- Pcbnew : circuit board layout editor
- GerbView: Gerber viewer

Two utilities are included

- Bitmap2Component : component maker for logos (Creates a schematic component or a footprint from a bitmap picture).
- PcbCalculator : A calculator that is helpfull to calculate components for regulators, track witdth versus current, transmission lines...

At the time of writing, KiCad can be considered mature enough to be used for the successful development and maintenance of complex electronic boards. Kicad does not present any board-size limitation and it can easily handle up to 16 copper layers and up to 12 technical layers. Kicad can create all the files necessary for building printed boards, Gerber files for photo-plotters, drilling files, component location files and a lot more. Being open source (GPL licensed), kicad represents the ideal tool for projects oriented towards the creation of electronic hardware with an open-source flavour.

Kicad is available for Linux, Windows and Apple OS X (experimental).

2 - Installation and configuration

2.1 - Display options

It is recommended to set your display/graphics card to use 24 or 32 bits per pixel. The 16-bit mode will work for Eeschema, but in Pcbnew the display will not function correctly under Linux.

2.2 - Initialization of the default configuration

A default configuration file named *kicad.pro* is supplied in kicad/template. It serves as a template for any new project.

The default file *kicad.pro* can be freely modified if necessary, mainly to set the list of libraries files loaded by Eeschema or Pcbnew.

Some other parameters (default text size, default line thickness, mainly for Pcbnew) are stored.

Verify you have a write acces to kicad/template/kicad.pro

Run kicad and load *kicad.pro* project.

Run Eeschema via kicad. Modify and update the Eeschema configuration

Run *Pcbnew* via kicad. Modify and update the Pcbnew configuration

2.3 - Kicad: principles of use

In order to manage a Kicad project: schematic files, printed circuit board files, supplementary libraries, manufacturing files for photo-tracing, drilling and automatic component placement files, it is recommended to create a project as follows:

- Create a working directory for the project (using KiCad or by other means).
- In this directory, use Kicad to create a project file (file with extension .pro) via the “Start a new project” icon.

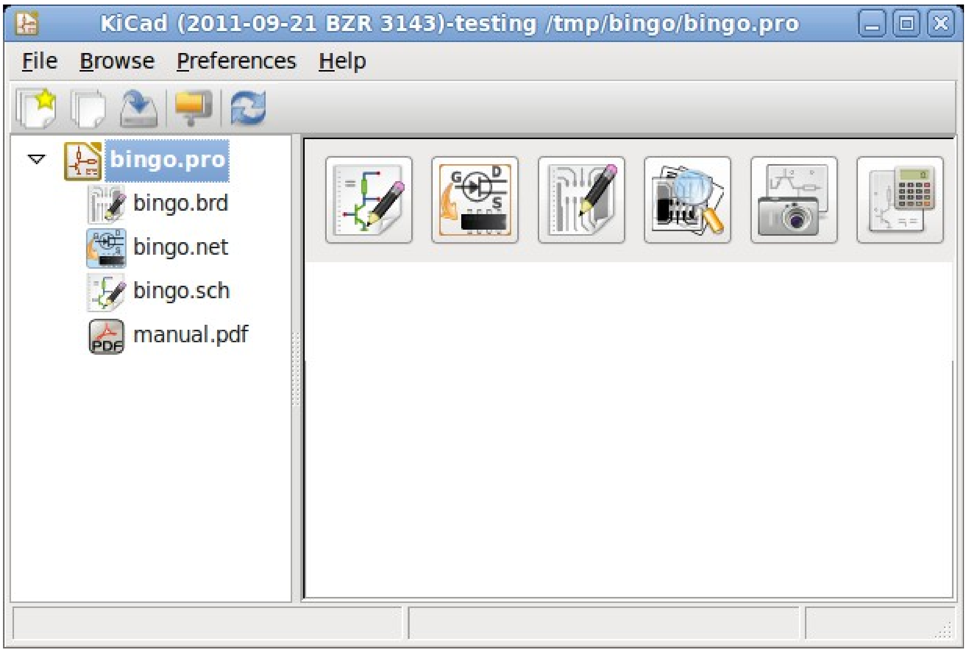
It is strongly recommended to use the same name for both project file and its directory.

Kicad creates a file with a .pro extension that maintains a number of parameters for project management (such as the filename of the principal schematic, list of libraries used in the schematics and PCBs). Default names of both principal schematic and printed circuit board files are derived from the name of the project. Thus, if a project called *example.pro* was created in a directory called *example*, the default files will be created:

example.pro	project management file.
example.sch	main schematic file.
example.brd	printed circuit board file.
example.net	netlist file.
example.xxx	various files created by the other utility programs.
example.cache.lib	cache file of libraries used in the schematic (backup of the components used)

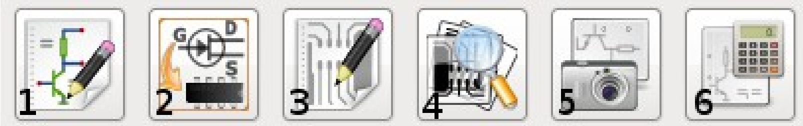
3 - Use Kicad

3.1 - Main window


KiCad (2011-09-21 BZR 3143)-testing /tmp/bingo/bingo.pro
File Browse Preferences Help
bingo.pro
bingo.brd
bingo.net
bingo.sch
manual.pdf

The main KiCad window is composed of a project tree view, a launch pane containing buttons used to run the various software tools, and a message window. The menu and the toolbar can be used to create, read and save project files.

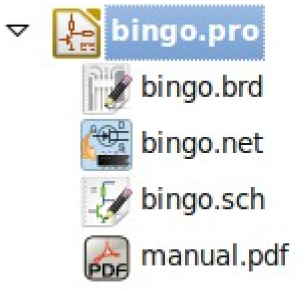
3.2 - Utility launch pane



Kicad allows you to run all stand alone software tools that come with it. The launch pane is made of the above 6 buttons that correspond to the following commands (from left to right):

1	Eeschema
2	Cvpcb
3	Pcbnew
4	Gerbview
5	Bitmap2component
6	Pcb Calculator

3.3 - Project tree view



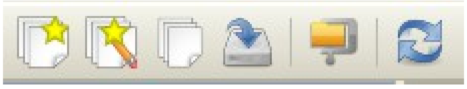
bingo.pro
bingo.brd
bingo.net
bingo.sch
manual.pdf

Double-clicking on the Eeschema icons runs the schematic editor which in this case will open the file bingo.sch.

Double-clicking on the Pcbnew icon runs the layout editor, in this case opening the file bingo.brd.

Right clicking on any of the files in the project tree allows generic files manipulation.

3.4 - Top toolbar



Kicad top toolbar allows for some basic files operation (from left to right).

	Create a project file. If the template kicad.pro is found in kicad/template, it is copied into the working directory.
	Create a project from a template.
	Open an existing project.
	Update and save the current project tree.
	Create a zip archive of the whole project. This includes schematic files, libraries, pcb, etc.
	Redraw the tree list, useful after a tree change.

4 - Using templates:

4.1 - Definitions

A template is a directory of files, which includes a directory of metadata.
The template system name (SYSNAME) is the directory name under which the template files are stored. The metadata directory (METADIR) contains pre-defined files which provide information about the template.

All files and directories in a template are copied to the new project path when a project is created using a template, except METADIR.

All files and directories which start with SYSNAME will have SYSNAME replaced by the new project file name, excluding the file extension.

4.2 - Templates:

They facilitate the easy setup of projects which have common attributes such as pre-defined board outlines, connector positions, schematic elements, design rules, etc.

4.2.1 - Metadata

A template's METADIR must contain the required files, and might optionally contain any of the optional files

4.2.2 - Required Files:

meta/info.html

Contains html formatted information about the template which is used by the user to determine if the template is what they are after. The <title> tag determines the actual name of the template that is exposed to the user for template selection.
Using html to format this document means that images can be in-lined without having to invent a new scheme.
Only a basic HTML language can be used to format this document.

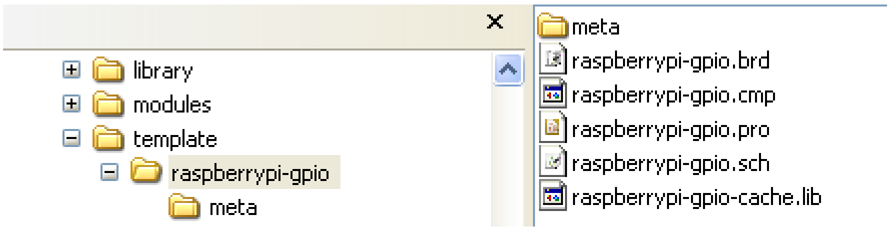
4.2.3 - Optional Files:

meta/icon.png

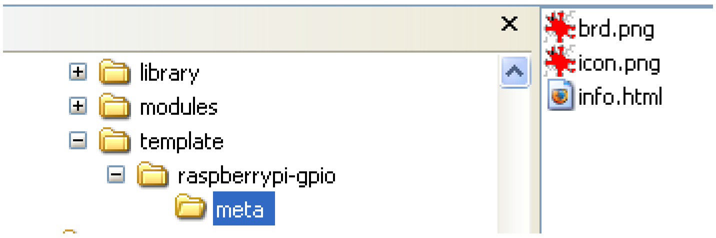
A 64 x 64 pixels PNG icon file which is used as a clickable icon in the template selection dialog.

4.2.4 - Example:

Here is a template for a raspberrypi-gpio board:



And the meta data info:



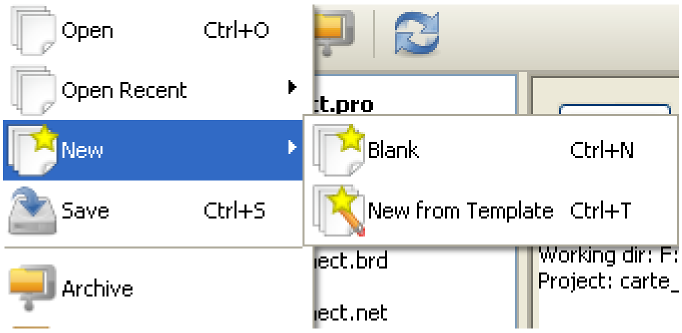
brd.png is an optional file

Here is an info.html file sample:

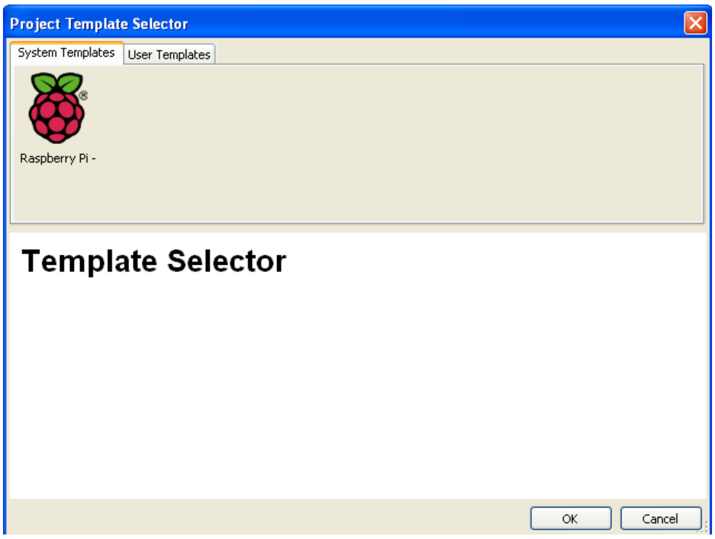
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
   <META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=windows-1252">
   <TITLE>Raspberry Pi - Expansion Board</TITLE>
   <META NAME="GENERATOR" CONTENT="LibreOffice 3.6  (Windows)">
   <META NAME="CREATED" CONTENT="0;0">
   <META NAME="CHANGED" CONTENT="20121015;19015295">
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>This project template is the basis of an expansion board for the
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM board.</A> <BR><BR>This base project includes a PCB edge defined
as the same size as the Raspberry-Pi PCB with the connectors placed
correctly to align the two boards. All IO present on the Raspberry-Pi
board is connected to the project through the 0.1&quot; expansion
headers. <BR><BR>The board outline looks like the following:
</P>
<P><IMG SRC="brd.png" NAME="brd" ALIGN=BOTTOM WIDTH=680 HEIGHT=378
BORDER=0><BR><BR><BR><BR>
</P>
<P>(c)2012 Brian Sidebotham<BR>(c)2012 Kicad Developers</P>
</BODY>
</HTML>
```

4.2.5 - Operation

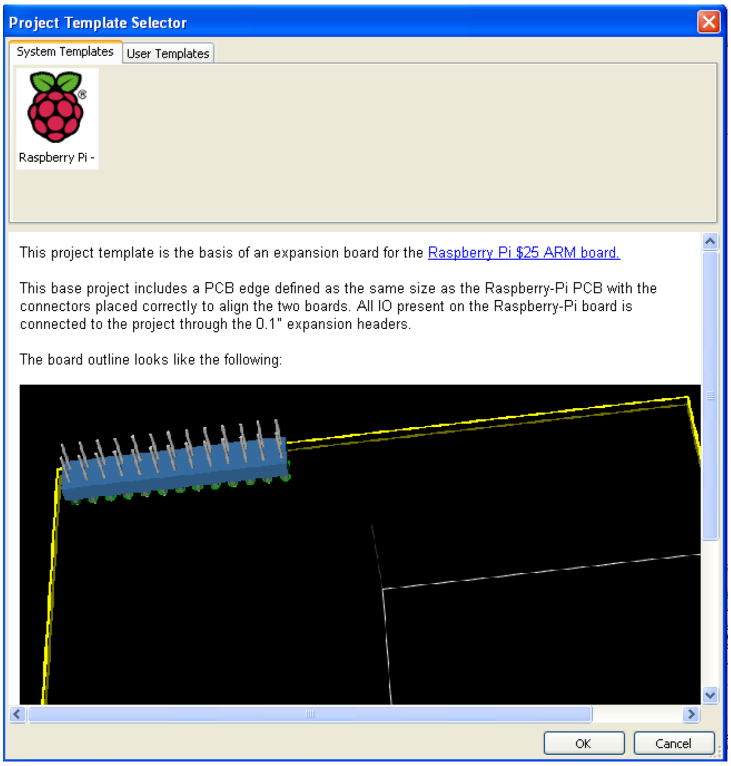
The KiCad File menu New shows 2 option:



- **Blank**
  Create a blank project by just copying template/kicad.pro to the current folder.
- **New from Template**
  Open the template selection dialog.
  The template selection dialog has a list of icons, and a display window.
  A single click on a template's icon on the left will load that templates info.html metadata file and display it in the display window.
  A click on the OK button starts the new project creation.
  The template will be copied to the new project location (excluding METADIR as mentioned earlier) and any files that match the string replacement rules will be renamed to reflect the new project's name.



After selection of a template:



This project template is the basis of an expansion board for the Raspberry Pi $25 ARM board.

This base project includes a PCB edge defined as the same size as the Raspberry-Pi PCB with the connectors placed correctly to align the two boards. All IO present on the Raspberry-Pi board is connected to the project through the 0.1" expansion headers.

The board outline looks like the following:

4.2.6 - Templates Location:

The list of available templates are gathered from the following sources:

- For sytem templates:
  `<kicad bin dir>/../share/template/`
- For user templates
  On Unix:
  `~/kicad/templates/`
  On Windows:
  `C:\Documents and Settings\username\My Documents\kicad\templates\`
  On Mac:
  `~/Documents/kicad/templates/`
- When the environment variable KICAD_PTEMPLATES is defined, there is a third page: Portable Templates, which lists templates found in KICAD_PTEMPLATES path.