KiCad

23 de mayo de 2017

Índice general

Manual de referencia

Contribuidores

Jean-Pierre Charras, Fabrizio Tappero.

Traducción

Antonio Morales <antonio1010.mr@gmail.com>, 2015

Realimentación

Por favor dirija cualquier reporte de fallo, sugerencia o nuevas versiones a:

- Acerca de la documentación de KiCad: https://github.com/KiCad/kicad-doc/issues
- Acerca del software KiCad: https://bugs.launchpad.net/kicad
- Acerca del software KiCad i18n: https://github.com/KiCad/kicad-i18n/issues

Fecha de publicación y versión del software

2015, May 21.

Capítulo 1

Introducción

1.1. KiCad

KiCad es una herramienta software open-source para la creación de diagramas electrónicos y diseño de placas de circuito impreso. Bajo su singular fachada, KiCad incorpora un elegante conjunto con las siguientes herramientas software:

- **KiCad**: Project manager
- **Eeschema**: Schematic editor and component editor
- **Pcbnew**: Circuit board layout editor and footprint editor
- **GerbView**: Gerber viewer

3 utility tools are also included:

- **Bitmap2Component**: Component maker for logos. It creates a schematic component or a footprint from a bitmap picture.
- **PcbCalculator**: A calculator that is helpful to calculate components for regulators, track width versus current, transmission lines, etc.
- **Pl Editor**: Page layout editor.

Estas herramientas están generalmente se ejecuta desde el director del proyecto, pero pueden ser también funcionan como herramientas independientes.

En este momento, KiCad se considera suficientemente maduro y puede utilizarse para el desarrollo y mantenimiento exitoso de PCB's complejas.

KiCad no presenta limitación alguna en cuanto al tamaño de la placa y puede gestionar hasta 32 capas de cobre, 14 capas técnicas y 4 capas auxiliares.

KiCad can create all the files necessary for building printed circuit boards, including:

- Archivos Gerber o fotolitos

- archivos de taladrado
- archivos de ubicación de componentes

Al ser de código abierto (licencia GPL), KiCad representa la herramienta ideal para proyectos orientados a la creación de equipos electrónicos con estilo open-source.

KiCad is available for Linux, Windows and Apple macOS.

1.2. Ficheros y directorios en KiCad

KiCad crea y utiliza archivos (y directorios) con las siguientes extensiones de archivos específicas para la edición de esquemas y placas.

Fichero del Gestor del proyecto:

*.pro	Archivo que contiene algunos parámetros para el proyecto actual, incluyendo la lista de bibliotecas de componentes.

Archivos del Editor de Esquemas

*.sch	Archivos de esquemas, que no contienen los componentes en si mismos.
*.lib	Archivos e bibliotecas de símbolos, que contienen las descripciones de los componentes: forma gráfica, pines, campos.
*.dcm	documentación de las bibliotecas de símbolos, contiene algunas descripciones de componentes: comentarios, palabras clave, referencia a las hojas de características.
*_cache.lib	archivo de caché para las bibliotecas de símbolos, que contiene una copia de los símbolos de los componentes utilizados en el esquema del proyecto proyecto.

Ficheros y directorios del Editor de placas:

*.kicad_pcb	Board file containing all info but the page layout.
*.pretty	Footprint library folders. The folder itself is the library.
*.kicad_mod	Footprint files, containing one footprint description each.
*.brd	Board file in the legacy format. Can be read, but not written, by the current board editor.
*.mod	Footprint library in the legacy format. Can be read by the footprint or the board editor, but not written.
fp-lib-table	Footprint library list (*footprint libraries table*): list of footprint libraries (various formats) which are loaded by the board or the footprint editor or CvPcb.

Ficheros comunes

*.kicad_wks	Page layout description files, for people who want a worksheet with a custom look.
*.net	Netlist file created by the schematic, and read by the board editor. This file is associated to the .cmp file, for users who prefer a separate file for the component/footprint association.

Ficheros especiales

*.cmp	Association between components used in the schematic and their footprints. It can be created by Pcbnew, and imported by Eeschema. The purpose is a back import from Pcbnew to Eeschema, for users who change footprints inside Pcbnew (for instance using *Exchange Footprints* command) and want to import these changes in schematic.

Otros ficheros

Son generados por KiCad para fabricación o documentación.

*.gbr	Gerber files, for fabrication.
*.drl	Drill files (Excellon format), for fabrication.
*.pos	Position files (ASCII format), for automatic insertion machines.
*.rpt	Report files (ASCII format), for documentation.
*.ps	Plot files (Postscript), for documentation.
*.pdf	Plot files (PDF format), for documentation.
*.svg	Plot files (SVG format), for documentation.
*.dxf	Plot files (DXF format), for documentation.
*.plt	Plot files (HPGL format), for documentation.

Capítulo 2

Instalación y configuración

2.1. Opciones de visualización

Pcbnew needs the support of OpenGL v2.1 or higher.

2.2. Instalación de la configuración por defecto

A default configuration file named **kicad.pro** is supplied in kicad/template. It serves as a template for any new project and is used to set the list of library files loaded by Eeschema. A few other parameters for Pcbnew (default text size, default line thickness, etc.) are also stored here.

Another default configuration file named **fp-lib-table** may exist. It will be used only once to create a footprint library list; otherwise the list will be created from scratch.

2.3. Modifying the default configuration

The default **kicad.pro** file can be freely modified, if desired.

Verifique que tiene acceso de escritura a kicad/template/kicad.pro

Run KiCad and load **kicad.pro** project.

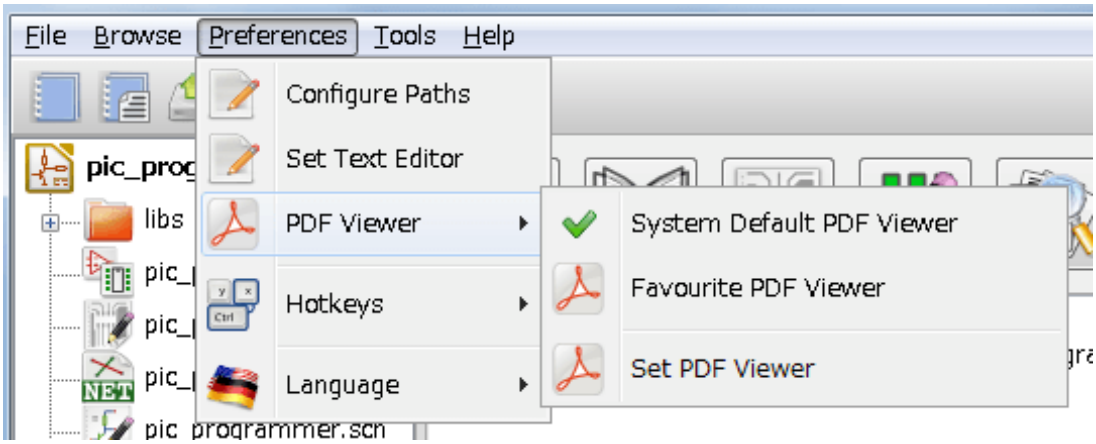
Run Eeschema via KiCad. Modify and update the Eeschema configuration, to set the list of libraries you want to use each time you create new projects.

Run Pcbnew via KiCad. Modify and update the Pcbnew configuration, especially the footprint library list. Pcbnew will create or update a library list file called **footprint library table**. There are 2 library list files (named fp-lib-table): The first (located in the user home directory) is global for all projects and the second (located in the project directory), if it exists, is specific to the project.

2.4. Initialization of external utilities

When using KiCad, choosing a text editor and a PDF viewer is useful.

These settings are accessible from the Preference menu:



2.5. Configuración de las rutas

In KiCad, one can define paths using an *environment variable*. A few environment variables are internally defined by KiCad, and can be used to define paths for libraries, 3D shapes, etc.

This is useful when absolute paths are not known or are subject to change, and also when one base path is shared by many similar items. Consider the following which may be installed in varying locations:

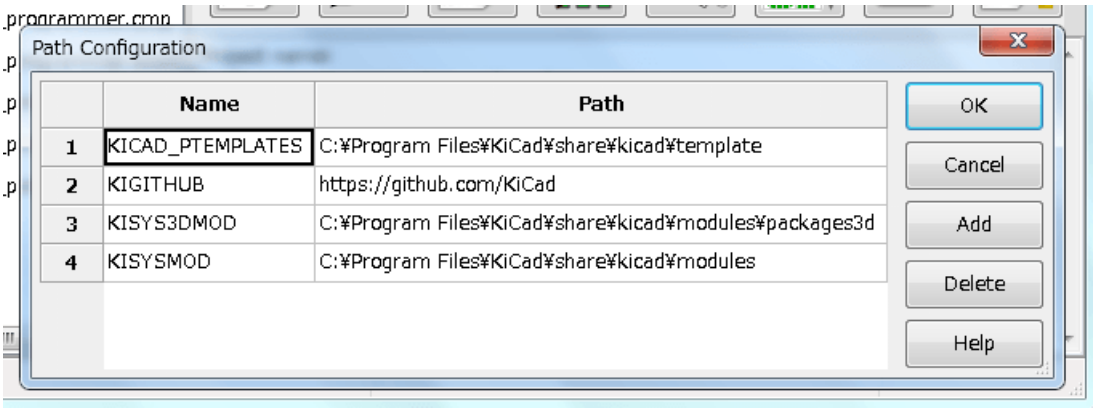
- Eeschema component libraries
- Pcbnew footprint libraries
- 3D shape files used in footprint definitions

For instance, the path to the ***connect.pretty*** footprint library, when using the **KISYSMOD** environment variable, would be ***${KISYSMOD}/connect.pretty***

This option allows you to define a path with an environment variable, and add your own environment variables to define personal paths, if needed.

KiCad environment variables:

KICAD_PTEMPLATES	Templates used during project creation. If you are using this variable, it must be defined.
KIGITHUB	Frequently used in example footprint lib tables. If you are using this variable, it must be defined.
KISYS3DMOD	Base path of 3D shapes files, and must be defined because an absolute path is not usually used.
KISYSMOD	Base path of footprint library folders, and must be defined if an absolute path is not used in footprint library names.



Note also the environment variable **KIPRJMOD** is **always** internally defined by KiCad, and is the **current project absolute path**.

For instance, ***${KIPRJMOD}/connect.pretty*** is always the ***connect.pretty*** folder (the pretty footprint library) found inside **the current project folder**.

If you modify the configuration of paths, please quit and restart KiCad to avoid any issues in path handling.

2.6. Selection of text editor

Before using a text editor to browse/edit files in the current project, you must choose the text editor you want to use.

Select ***Preferences → Set Text Editor*** to set the text editor you want to use.

2.7. Selection of PDF viewer

You may use the default PDF viewer or choose your own.

To change from the default PDF viewer use ***Preferences → PDF Viewer → Set PDF Viewer*** to choose the PDF viewer program, then select ***Preferences → PDF Viewer → Favourite PDF Viewer***.

On Linux the default PDF viewer is known to be fragile, so selecting your own PDF viewer is recommended.

2.8. KiCad principles of use

In order to manage a KiCad project of 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:

- **Crear un directorio de trabajo para el proyecto** (usando KiCad u otros medios).
- **En este directorio, utilice KiCad para crear un archivo de proyecto** (archivo con extensión .pro) a través de los iconos de opción ”Crear un nuevo proyecto” o ”Crear un nuevo proyecto desde plantilla”.

---

aviso

Use a unique directory for each KiCad project. Do not combine multiple projects into a single directory.

KiCad creates a file with a .pro extension that maintains a number of parameters for project management (such as the list of libraries used in the schematic). Default names of both main schematic file and printed circuit board file 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.kicad_pcb	Printed circuit board file.
example.net	Netlist file.
example.*	Various files created by the other utility programs.
example-cache.lib	Library file automatically created and used by the schematic editor containing a backup of the components used in the schematic.

Capítulo 3

Using KiCad manager

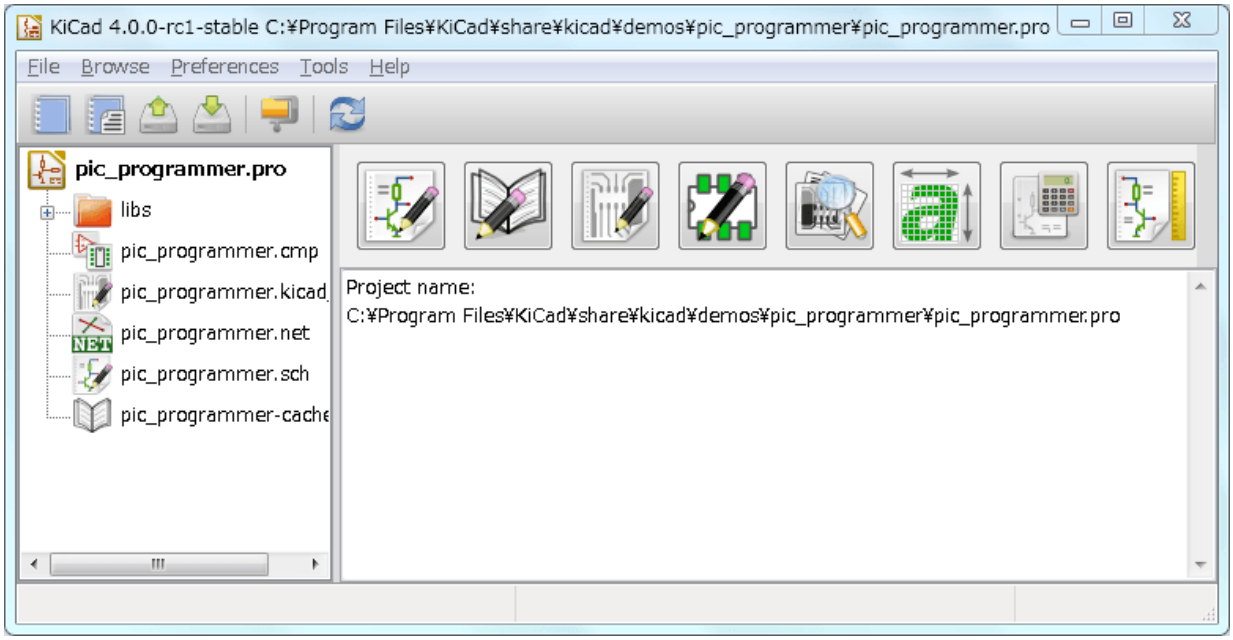
The KiCad Manager (kicad or kicad.exe) is a tool which can easily run the other tools (schematic and PCB editors, Gerber viewer and utility tools) when creating a design.

Ejecutar los otros programas desde el gestor del proyecto de KiCad tiene algunas ventajas:

- control cruzado entre el editor de esquemas y el editor de placas de circuito
- control cruzado entre el editor de esquemas y el selector de huellas (CvPcb).

However, you can only edit the current project files. When these tools are run in *stand alone* mode, you can open any file in any project but cross probing between tools can give strange results.

3.1. Ventana principal

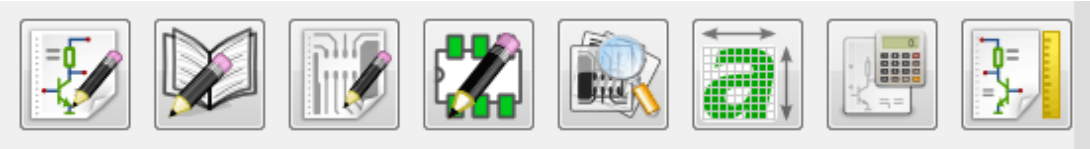


La ventana principal KiCad se compone de una vista tipo árbol del proyecto, un panel que contiene botones que sirven para ejecutar las distintas herramientas del software, y una ventana de mensaje. El menú y la barra de herramientas se pueden utilizar para crear, leer y guardar archivos de proyecto.

3.2. Panel de utilidades

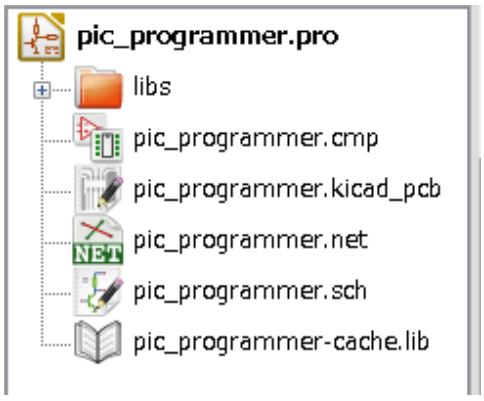
KiCad le permite ejecutar todas la herramientas software autónomas que están incluidas en el.

El panel de herramientas esta compuesto por 8 botones que corresponden a los siguientes comandos (de 1 a 8, de izquierda a derecha)



1	**Eeschema**	Schematic editor.
2	**LibEdit**	Component editor and component library manager.
3	**Pcbnew**	Board layout editor.
4	**FootprintEditor**	Footprint editor and footprint library manager.
5	**Gerbview**	Gerber file viewer. It can also display drill files.
6	**Bitmap2component**	Tool to build a footprint or a component from a B&W bitmap image to create logos.
7	**Pcb Calculator**	Tool to calculate track widths, and many other things.
8	**Pl Editor**	Page layout editor, to create/customize frame references.

3.3. Vista de árbol del proyecto



Double-clicking on the Eeschema icon runs the schematic editor, in this case opening the file **pic_programmer.sch**.

Double-clicking on the Pcbnew icon runs the layout editor, in this case opening the file **pic_programmer.kicad_pcb**.

Haciendo clic derecho sobre cualquiera de los archivos en el árbol del proyecto le permite ver las acciones genéricas de manipulación del archivo.

3.4. Barra de herramientas superior

KiCad top toolbar allows for some basic file operations:

	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.
	Rebuild and redraw the tree view, sometimes needed after a tree change.

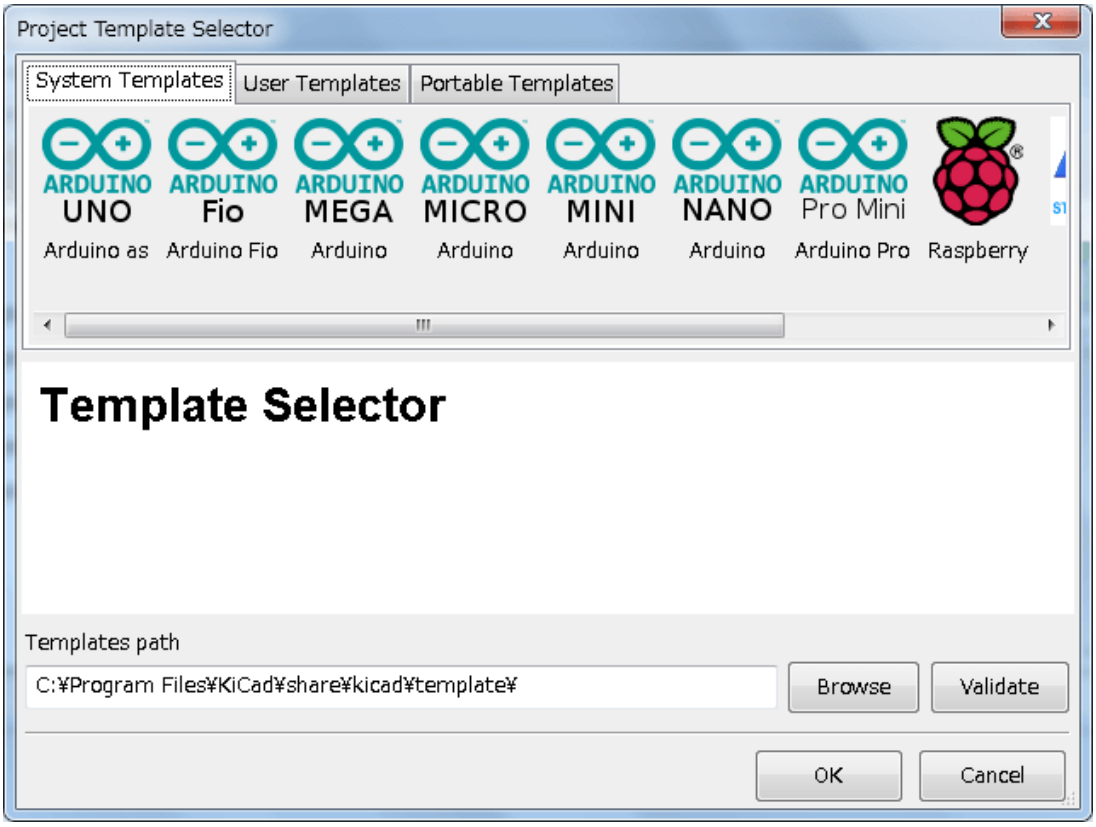
Capítulo 4

Project templates

A template facilitates the easy creation of a new project, based on a template definition. Templates may contain pre-defined board outlines, connector positions, schematic elements, design rules, etc. Complete schematics and/or PCBs used as seed files for the new project may even be included.

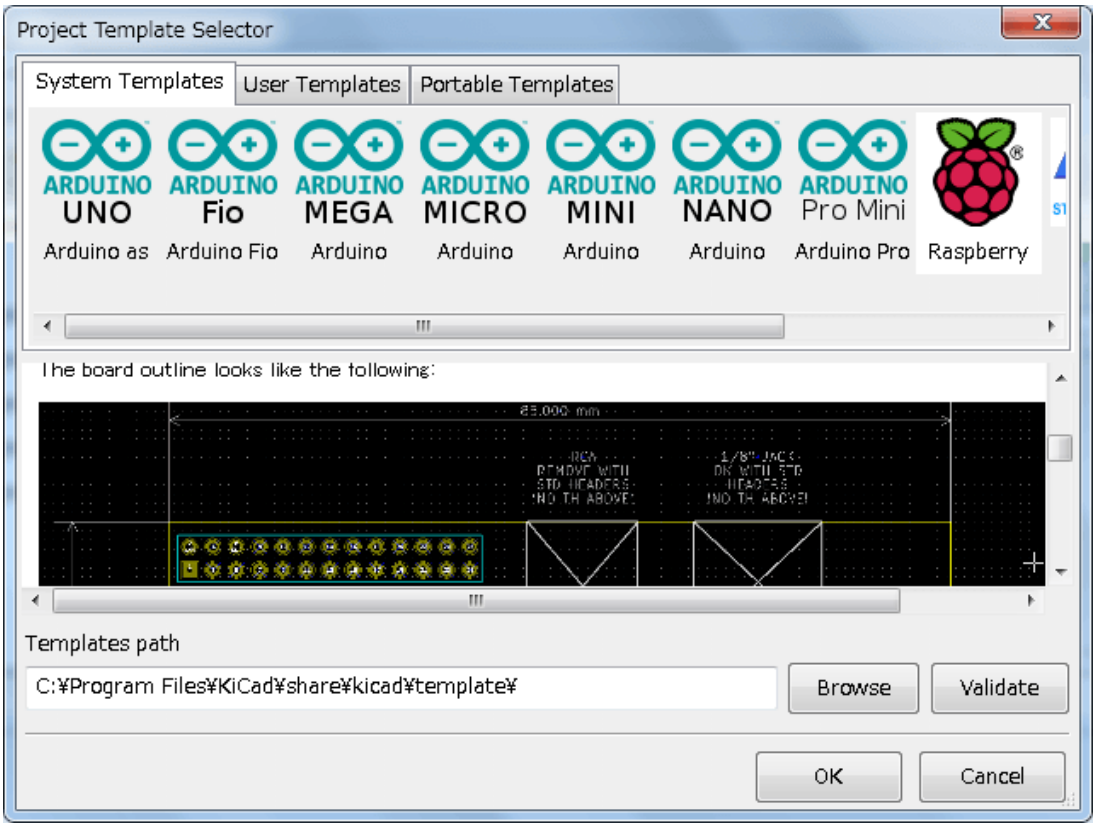
4.1. Using Templates

The ***File → New Project → New Project from Template*** menu will open the Project Template Selector dialog:



A single click on a template's icon will load that template's information, and a further click on the OK button creates the new project. The template files will be copied to the new project location and renamed to reflect the new project's name.

Después de seleccionar una plantilla:



4.2. Template Locations:

La lista de las plantillas disponibles se recopila de las siguientes fuentes:

- System templates: <kicad bin dir>/../share/kicad/template/
- User templates:
  - Unix: ~/kicad/templates/
  - Windows: C:\Documents and Settings\username\My Documents\kicad\templates
  - Mac: ~/Documents/kicad/templates/
- When the environment variable KICAD_PTEMPLATES is defined there is a third tab, Portable Templates, which lists templates found at the KICAD_PTEMPLATES path.

4.3. Creating templates

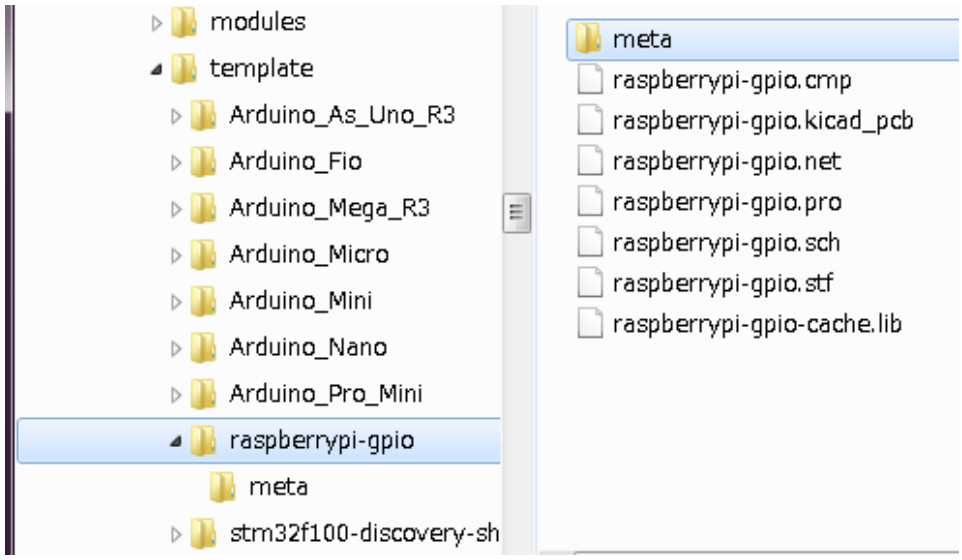
The template name is the directory name under which the template files are stored. The metadata directory, in a subdirectory named **meta**, contains files which describe the template.

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

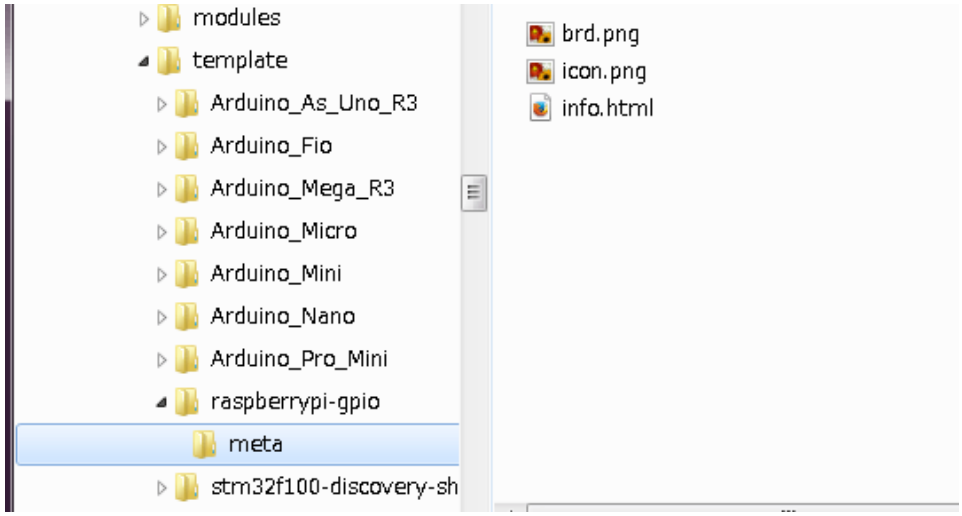
All files and directories which start with the template name will be renamed with the new project file name, excluding the file extension.

The metadata consists of one required file, and may contain optional files. All files must be created by the user using a text editor or previous KiCad project files, and placed into the required directory structure.

Here are project files for a **raspberrypi-gpio** template:



And the metadata files:



4.3.1. Required File:

meta/info.html	HTML-formatted information describing the template.

The <title> tag determines the actual name of the template that is exposed to the user for template selection. Note that the project template name will be cut off if it's too long. Due to font kerning, typically 7 or 8 characters can be displayed.

Using HTML means that images can be easily in-lined without having to invent a new scheme. Only basic HTML tags can be used in this document.

Here is a sample **info.html** file:

```
<!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>
```

4.3.2. Ficheros Opcionales:

meta/icon.png	A 64 x 64 pixel PNG icon file which is used as a clickable icon in the template selection dialog.

Any other image files used by **meta/info.html**, such as the image of the board file in the dialog above, are placed in this folder as well.