



kiCad



KiCad

KiCad

13. Januar 2018

Inhaltsverzeichnis

1	Einleitung	1
1.1	KiCad	1
1.2	KiCad Dateien und Ordner	2
2	Installation und Konfiguration	4
2.1	Anzeige Optionen/Bedingungen	4
2.2	Vorbereitung der Standard Konfiguration	4
2.3	Vorbereitung der Standard Konfiguration	4
2.4	Vorbereitung einiger Optionen und Hilfsmittel	5
2.5	Konfiguration der Pfade	5
2.6	Voreinstellung des Texteditors	6
2.7	Voreinstellung des PDF-Betrachter	6
2.8	Prinzipien der KiCad Benutzung	6
3	Benutzung des KiCad Projektmanagers	8
3.1	Kicad Hauptfenster	8
3.2	Schnellstart-Panel der Hilfstools	9
3.3	Die Projektübersicht	9
3.4	Obere Toolbar	9
4	Projekt Vorlagen:	11
4.1	Benutzung von Vorlagen	11
4.2	Ablageorte von Vorlagen:	12
4.3	Erstellen von Vorlagen	13
4.3.1	Benötigte Dateien:	15
4.3.2	Optionale Dateien:	16

Referenz Handbuch

Copyright

Dieses Dokument ist geschützt © 2010-2015 durch deren Beitragende, welche nachfolgend aufgeführt sind. Sie können es nach den Bedingungen der GNU General Public License (<https://www.gnu.org/licenses/gpl.html>), Version 3 oder später, oder der Creative Commons Attribution License (<https://creativecommons.org/licenses/by/3.0/>), Version 3.0 oder später verteilen oder verändern.

Alle Markenrechtsnamen in diesem Guide gehören den rechtmäßigen Eigentümern.

Beitragende

Jean-Pierre Charras, Fabrizio Tappero.

Übersetzung

Carsten Schönert <c.schoenert@t-online.de>, 2016

Feedback

Bitte alle Bug Reports, Vorschläge oder neue Versionen an folgende Adressen richten:

- KiCad Dokumentation: <https://github.com/KiCad/kicad-doc/issues>
- KiCad Software: <https://bugs.launchpad.net/kicad>
- KiCad Software i18n Übersetzung: <https://github.com/KiCad/kicad-i18n/issues>

Datum der Veröffentlichung und Softwareversion

21.05.2015

Kapitel 1

Einleitung

1.1 KiCad

KiCad ist ein Open Source Softwaretool zum Erstellen von elektronischen Schaltplänen und PCB Layouts. Durch seine einheitliche Schnittstellentechnik verbindet KiCad elegant die einzelnen folgenden eigenständigen Software Tools:

- **KiCad** : Projektmanager
- **Eeschema** : Schaltplan- und Bauteileditor
- **Pcbnew** : Designtool für gedruckte Leiterplatten und Footprints
- **GerbView** : Betrachter für Gerberdateien

3 weitere Hilfetools:

- **Bitmap2Component**: Bauteilerstellung für Logos. Erstellt aus einem Bitmap Bild ein Bauteil oder einen Footprint.
- **PcbCalculator**: Ein Kalkulator, der hilfreich ist, wenn man Spannungsregler, Leiterbahnen, Übertragungsleitungen oder andere Elemente berechnen muss.
- **PI Editor**: Seiten Layout Editor.

Diese Tools werden üblicher Weise aus dem Projekt Manager heraus gestartet, können aber auch als eigenständige Programme gestartet werden.

Zum Zeitpunkt des Schreibens dieser Referenz kann KiCad als ausreichend ausgereift betrachtet werden, um erfolgreich auch komplexe elektronische Schaltungen und Leiterplatten entwickeln und managen zu können.

KiCad besitzt keine Begrenzungen bei der Platinengröße, und es kann bis zu 32 Kupfer Lagen, 14 technische Lagen und 4 zusätzliche Lagen handhaben.

KiCad kann alle Dateien erstellen, die notwendig sind, um Platinen produzieren zu lassen:

- Gerber Dateien für Foto-Plotter
-

- Bohrdateien
- Bauteilpositionsdateien

Durch die Open Source Ausrichtung (GPL Lizenz) repräsentiert KiCad sich als das ideale Tool für fortschrittlich ausgerichtete Projekte, die elektronische Hardware erstellen.

KiCad ist erhältlich für Linux, Windows und Apple Mac OS.

1.2 KiCad Dateien und Ordner

KiCad benutzt und erstellt folgende spezifischen Dateierweiterungen (und Ordner) für die Schaltplan und Leiterplattenbearbeitung.

Projektmanagerdatei:

*.pro	Kleine Datei, die wenige Parameter für das aktuelle Projekt enthält, inklusive der Liste der Bauteilbibliotheken.
-------	---

Schaltplan Editor Dateien:

*.sch	Schaltplandateien, welche nicht die Bauteile selbst enthalten.
*.lib	Schaltplan Bauteil Bibliotheksdateien, enthalten die Bauteileigenschaften: grafische Formen, Pins, Felder.
*.dcm	Schaltplan Bauteil Bibliotheken Dokumentation, enthalten einige Bauteil Beschreibungen: Kommentare, Schlüsselwörter, Referenzen zu Datenblättern.
*_cache.lib	Schaltplan Bauteil Bibliothek Cache Datei, beinhaltet Kopien der benutzen Bauteile im Schaltplan vom Projekt.

Leiterplatteneditor Dateien und Ordner:

*.kicad_pcb	Leiterplattendatei, die alle Infos des Platinenlayouts enthält.
*.pretty	Footprint Bibliotheks-Ordner. Der Ordner selbst ist die Bibliothek.
*.kicad_mod	Footprint Dateien, beinhalten einen Footprint mit der jeweiligen Beschreibung.
*.brd	Platinendatei im alten Format. Kann vom Footprint oder Platinen Editor gelesen, aber nicht geschrieben werden.
*.mod	Footprint Bibliothek im aktuellen Format. Kann vom Platinen oder Footprint Editor gelesen werden, aber nicht geschrieben.
fp-lib-table	Liste der Footprint Bibliotheken (<i>Footprint Bibliothekstabelle</i>): Liste der Footprint Bibliotheken (verschiedene Formate), welche vom Platinen oder Footprint Editor oder auch CvPcb geladen wird.

Gemeinsam genutzte Dateien:

*.kicad_wks	Die Seitenbeschreibungsdateien, für Benutzer, die ein Arbeitsblatt mit einem angepassten Aussehen benutzen wollen.
-------------	--

*.net	Netzliste, die vom Schaltplaneditor erstellt und vom Platinen Editor gelesen wird. Diese Datei ist mit der .cmp Datei verknüpft, wenn Benutzer eine separate Datei für die Bauteil Footprint Verknüpfung bevorzugen.
-------	--

Spezielle Dateien:

*.cmp	Verbindung zwischen den Bauteilen, die im Schaltplan benutzt werden und deren Footprints. Diese können durch Pcbnew erstellt und durch Eeschema importiert werden. Die Idee hinter dieser Art der Zuordnung liegt in deren gegebenen Möglichkeit des Rückimports von Pcbnew nach Eeschema, die von manchen Benutzern angewendet wird die Footprints innerhalb von Pcbnew ändern (z.B. durch Benutzen des Kommandos <i>Austausch Footprint</i>) und diese Änderungen in den Schaltplan übernehmen wollen.
-------	---

Andere Dateien:

Diese Dateien werden von KiCad erstellt für die Fabrikation oder auch Dokumentation.

*.gbr	Gerber Dateien für die Produktion der Platine.
*.drl	Bohrdateien (Excellon Format) für die Produktion der Platine.
*.pos	Positionierungsdateien (ASCII Format), für Maschinen, die die automatische Bestückung der Platine durchführen.
*.rpt	Report Dateien (ASCII Format), für Dokumentation.
*.ps	Plot Dateien (Postscript), für Dokumentation.
*.pdf	Plot Dateien (PDF Format), für Dokumentation.
*.svg	Plot Dateien (SVG Format), für Dokumentation.
*.dxf	Plot Dateien (DXF Format), für Dokumentation.
*.plt	Plot Dateien (HPGL Format), für Dokumentation.

Kapitel 2

Installation und Konfiguration

2.1 Anzeige Optionen/Bedingungen

Pcbnew benötigt Unterstützung durch OpenGL v2.1 oder höher.

2.2 Vorbereitung der Standard Konfiguration

Eine Konfigurationsdatei mit allgemeinen Standards, **kicad.pro** genannt, wird im Vorlagenordner mit ausgeliefert. Es beinhaltet eine Vorlage für jedes neues Projekt und beschreibt die Liste der Bibliotheken, die durch Eeschema geladen werden. Es sind einige weitere Parameter für Pcbnew (wie Standardtextgröße, Standard Stärke der Leiterzüge usw.) ebenfalls in ihr gespeichert.

Wenn eine andere Standard Konfigurationsdatei, **fp-lib-table** genannt existiert, dann wird die Datei einmalig benutzt, um die Liste mit den Footprint Bibliotheken zu erstellen. (Oder, wenn nicht vorhanden, wird diese Datei komplett neu erstellt.)

2.3 Vorbereitung der Standard Konfiguration

Wenn nötig, kann die Datei **kicad.pro** nach Bedarf und Anforderungen angepasst werden.

Stellen Sie sicher, dass Sie Schreibberechtigungen auf `kicad/template/kicad.pro` besitzen.

Starten und Laden Sie das Projekt **kicad.pro**.

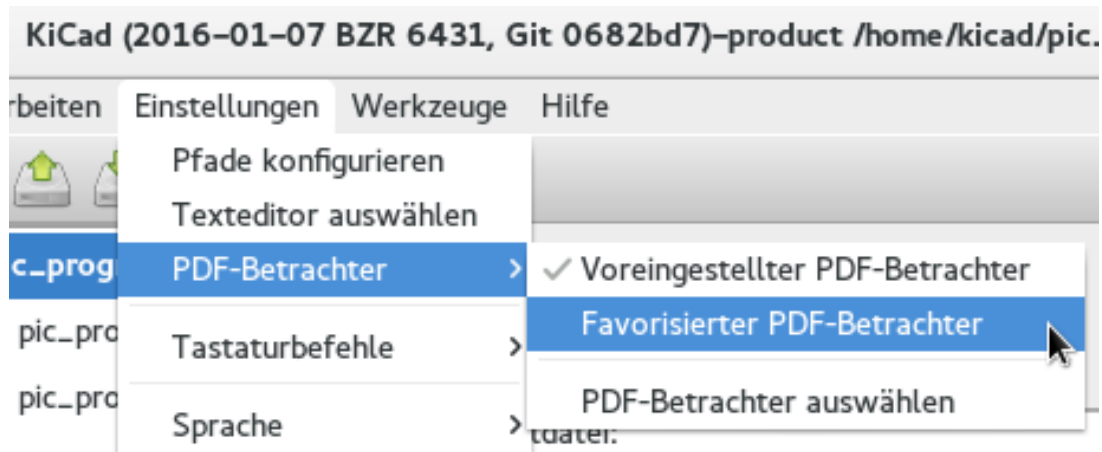
Starten Sie Eschema via dem KiCad Projekt Manager. Verändern und aktualisieren Sie die Eeschema Konfiguration, und vorzugsweise die Liste der Bibliotheken, die Sie jedes mal benutzen wollen, wenn Sie ein neues Projekt erstellen.

Starten Sie Pcbnew via KiCad Projekt Manager. Modifizieren und aktualisieren Sie die Pcbnew Konfiguration, speziell die Liste der Footprint Bibliotheken. Pcbnew wird die Bibliotheksliste, **Footprint Bibliothek-Tabelle** genannt, erstellen oder aktualisieren. Es gibt zwei Dateien dieser Listen mit den Bibliotheken, jeweils `_fp-lib-table_` genannt. Die erste (im Heimatverzeichnis des Benutzers liegend) ist global für alle Projekte gültig. Die zweite Datei (wenn vorhanden und im Projektverzeichnis abgelegt) ist nur für das Projekt gültig.

2.4 Vorbereitung einiger Optionen und Hilfsmittel

Wenn KiCad benutzt wird, ist ein Texteditor und ein PDF Betrachter meistens hilfreich. Diese einzurichten, ist daher eine gute Idee.

Die Einstellungen sind erreichbar über das Menü Einstellungen:



2.5 Konfiguration der Pfade

In KiCad können manche Pfade durch *Umgebungsvariablen* definiert werden. Einige der Umgebungsvariablen werden intern durch KiCad gesetzt, und können benutzt werden, um Pfade zu definieren (für Bibliotheken, 3D-Formen, und mehr).

Dies ist besonders hilfreich, wenn die absoluten Pfade nicht bekannt sind oder auch Gegenstand der notwendigen Veränderung sind, und auch, wenn ein Basispfad zahlreiche ähnliche Elemente enthält. Betrachten wir folgendes, was in unterschiedlichen Stellen installiert sein könnte:

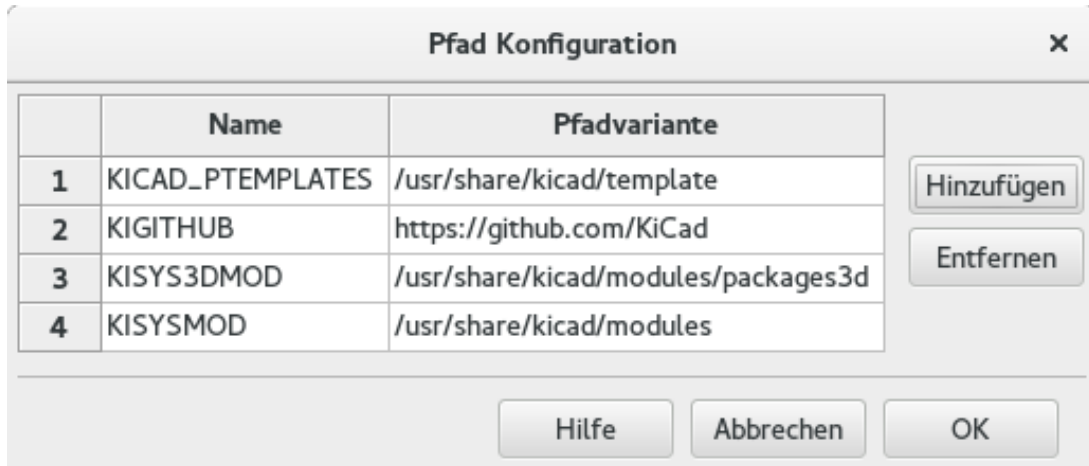
- Eeschema Bauteil Bibliotheken
- Pcbnew Footprint Bibliotheken
- 3D-Formen die in den Footprint Definitionen benutzt werden

Zum Beispiel, der volle (absolute) Pfad der Footprint Bibliothek **connect.pretty** wird gebildet, indem die Umgebungsvariable **KISYSMOD** benutzt wird, um den vollen Pfad zur Bibliothek zu definieren: `${KISYSMOD}/connect.pretty`

Diese Option erlaubt es Ihnen, sowohl einen Pfad durch eine Umgebungsvariable zu setzen als auch, wenn nötig, eigene Umgebungsvariablen zu definieren, um eigene Pfadangaben zu benutzen.

KiCad Umgebungsvariablen:*

KICAD_PTEMPLATES	Vorlagen, die während der Projekt Erstellung benutzt werden. Wenn Sie diese Variable benutzen wollen, muss diese definiert sein.
KIGITHUB	Häufig benutzt in Beispielen der Footprint Bibliothek-Tabellen. Wenn Sie diese Variable benutzen, muss diese definiert sein.
KISYS3DMOD	Standard Pfad für 3D-Formen, muss definiert sein, da ein absoluter Pfad üblicherweise nicht benutzt wird.
KISYSMOD	Standard Pfad für den Ordner mit den Footprint Bibliotheken, muss definiert werden, wenn kein absoluter Pfad in den Footprint Bibliotheken Namen benutzt wird.



Beachten Sie ebenso die Umgebungsvariable **KIPRJMOD**, die **immer** intern durch KiCad definiert ist und den **aktuellen absoluten Projekt Pfad** abbildet.

Zum Beispiel, $\${KIPRJMOD}/connect.pretty$ ist immer der Ordner *connect.pretty* (die pretty Footprint Bibliothek) der **innerhalb des aktuellen Projektes** liegt.

Wenn Sie die Konfiguration von Pfaden verändern, beenden und starten Sie KiCad neu, um mögliche Fehler bei der Pfad Benutzung zu verhindern.

2.6 Voreinstellung des Texteditors

Bevor Sie einen Texteditor benutzen können, um im aktuellen Projekt Dateien betrachten oder auch ändern zu können, müssen Sie einen Editor auswählen.

Wählen Sie *Einstellungen* → *Texteditor auswählen*, um den gewünschten Texteditor festzulegen.

2.7 Voreinstellung des PDF-Betrachter

Sie können den vom System eingestellten PDF-Betrachter als Standard PDF-Betrachter einstellen.

Um den Standard PDF-Betrachter zu ändern, benutzen Sie *Einstellungen* → *PDF-Betrachter* → *PDF-Betrachter auswählen*, um einen anderen PDF-Betrachter auszuwählen, und aktivieren diese Auswahl durch Auswahl des Menüpunktes *Einstellungen* → *PDF-Betrachter* → *Favorisierter PDF-Betrachter*.

Sollte unter Linux der Standard PDF-Betrachter manchmal Schwierigkeiten bereiten, dann kann die Auswahl und Benutzung eines anderen PDF-Betrachters sinnvoll sein.

2.8 Prinzipien der KiCad Benutzung

Für die Koordination von Schaltplandateien, Platinendateien, zusätzliche Bibliotheken, Fabrikationsdateien für Foto-Tracing, Bohrungen und Dateien für automatische Bauteilpositionierung empfiehlt es sich, Projekte folgendermaßen zu erstellen:

- **Erstellen Sie ein Arbeitsverzeichnis für das Projekt** (entweder durch KiCad selbst oder durch andere Tools).

- **Erstellen Sie eine Projektdatei durch KiCad in diesem Verzeichnis** (Datei mit der Erweiterung .pro) durch "Neues Projekt → Neues Projekt" oder durch "Neues Projekt → Projekt aus einer Vorlage".



Warnung

Benutzen Sie einzelne eindeutige Verzeichnisse für jedes KiCad Projekt. Kombinieren Sie **nicht** mehrere Projekte innerhalb eines Verzeichnisses.

KiCad erstellt eine Datei mit der Erweiterung .pro, welche eine Anzahl an Parametern für die Projektverwaltung, wie zum Beispiel die Liste der benutzten Bibliotheken enthält. Der Name der Schaltplandatei als auch der Platinendatei wird aus dem Projektnamen abgeleitet. Wenn also ein Projekt **example.pro** in einem Verzeichnis **example** erstellt worden ist, werden folgende Dateien durch KiCad erstellt:

example.pro	Projekt Management Datei.
example.sch	Hauptschaltplan Datei.
example.kicad_pcb	Platinen Datei.
example.net	Netzlisten Datei.
example.*	Weitere verschiedene Dateien durch andere Tools erstellt.
example-cache.lib	Bibliotheksdatei, die automatisch erstellt und vom Schaltplaneditor benutzt wird. (Diese Datei enthält Backups der Bauteile, die im Schaltplan benutzt werden).

Kapitel 3

Benutzung des KiCad Projektmanagers

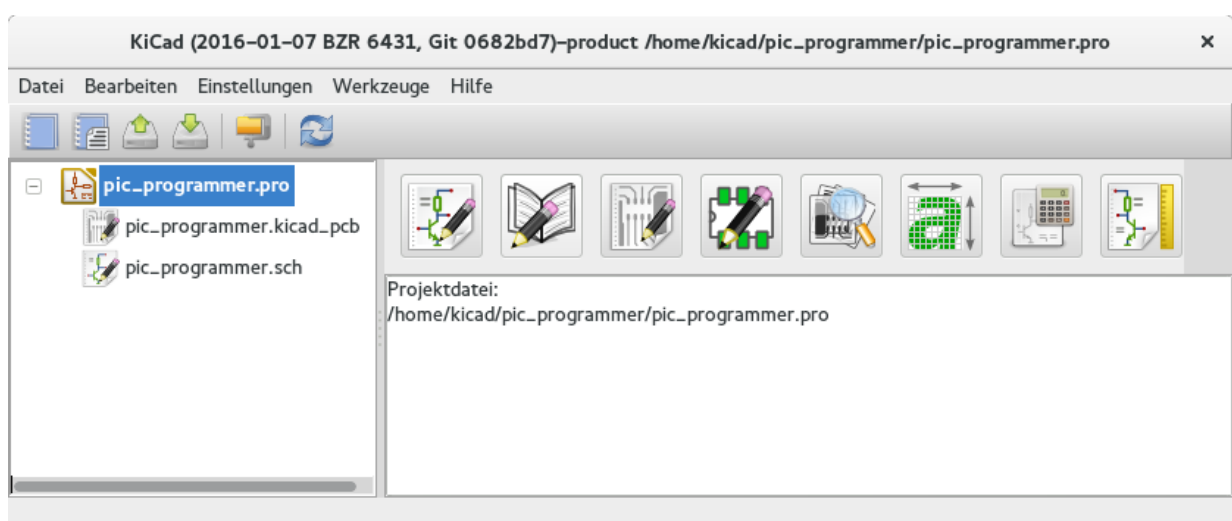
Der KiCad Projektmanager (unter Linux *kicad* und in Windows *kicad.exe* benannt) ist ein Tool, welches die anderen Tools (Editoren, Betrachter für Gerberdaten und Hilfstools) starten kann, wenn ein Entwicklungsprozess läuft.

Das Starten der jeweiligen Tools aus dem KiCad Projektmanager heraus hat einige Vorteile:

- Quervergleiche zwischen Schaltplaneditor und Platineneditor
- Quervergleiche zwischen Schaltplaneditor und der Footprint Auswahl (CvPcb)

Sie können nur die aktuellen Projektdateien bearbeiten. Wenn diese Tools als *eigenständige Applikation* aufgerufen werden, können Sie jede Datei aus unterschiedlichen Projekten öffnen, Quervergleiche zwischen diesen Tools können dann aber merkwürdige Ergebnisse entstehen lassen.

3.1 Kicad Hauptfenster

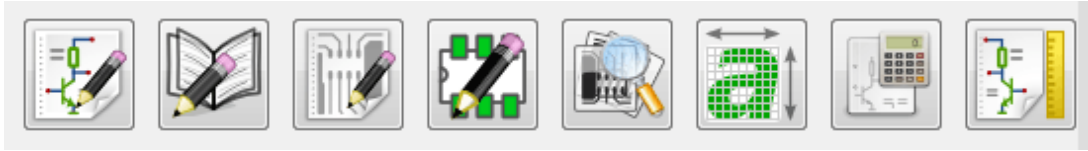


Das Hauptfenster im KiCad Projekt Manager zeigt die einzelnen Dateien des Projektes in einem Projektbaum dargestellt, ein Schnellstart-Panel mit den einzelnen Icons für die unterschiedlich Tools und ein Fenster für Meldungen. Das Menü als auch die Toolbar kann benutzt werden, um Projekte zu erstellen, zu öffnen oder zu speichern.

3.2 Schnellstart-Panel der Hilfstools

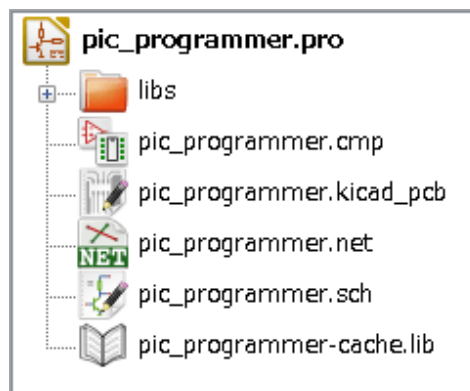
Der KiCad Projekt Manager erlaubt es Ihnen, alle einzelnen Software Tools zu starten, die zusammen mit KiCad verteilt werden.

Das Schnellstart-Panel beinhaltet acht Buttons, welche folgende Kommandos ausführen (1 bis 8, von Links nach Rechts):



1	Eeschema	Der Schaltplaneditor.
2	Bauteileditor	Der Editor für Bauteile und der Manager für die Bauteil Bibliotheken.
3	Pcbnew	Der Platineneditor.
4	PCB Footprinteditor	Der Footprinteditor und Footprint Bibliotheksmanager.
5	GerbView	Ein Betrachter für GERBER Daten. Kann ebenfalls Bohrdateien anzeigen.
6	Bitmap2Component	Ein Tool, um Footprints oder Bauteile von einem S&W Bitmap zu erstellen, oder auch Logos.
7	Pcb Kalkulator	Ein Kalkulationstool, um Leiterbahnbreiten oder viele andere Dinge berechnen zu können.
8	PI Editor	Seiten Layout Editor, erstellt oder passt Layouts an.

3.3 Die Projektübersicht



Ein Doppelklick auf das Eeschema Icon startet den Schaltplaneditor, welcher in diesem Fall die Datei **pic_programmer.sch** öffnen wird.







Ein Doppelklick auf das Pcbnew Icon startet den Platineneditor, welcher hier die Datei **pic_programmer.kicad_pcb** öffnen wird.

Ein Rechts-Klick auf irgendeine Datei in der Projektübersicht ermöglicht diverse generische Dateiveränderungen.

3.4 Obere Toolbar



Die obere Toolbar im Projektmanager erlaubt einige allgemeine Dateioperationen:

	Erstellt eine Projektdatei. Wenn eine Vorlage kicad.pro in kicad/template vorhanden ist, wird diese in das Arbeitsverzeichnis kopiert.
	Erstellt ein Projekt aus einer Vorlage.
	Öffnet ein bestehendes Projekt.
	Aktualisiert und speichert den aktuellen Projektbaum.
	Erstellt ein Zip-Archiv vom kompletten Projekt. Diese beinhaltet Schaltplandateien, Bibliotheken, Leiterplattendaten und mehr.
	Erstellt den Projektbaum und dessen Anzeige neu, ist manchmal nötig nach Veränderungen im Baum.

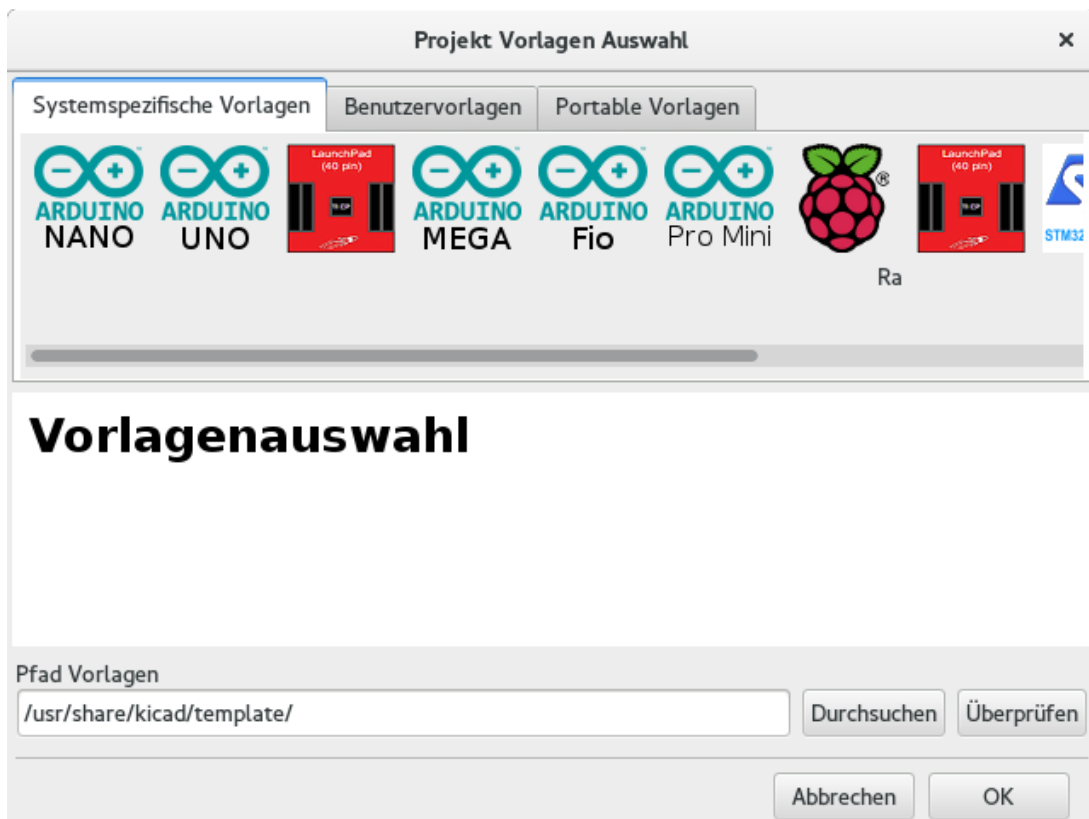
Kapitel 4

Projekt Vorlagen:

Vorlagen erleichtern das Setup eines neuen Projektes, welche gemeinsam benutzte Attribute wie vordefinierte Platinengrößen, Verbindert Positionen, Schaltplan Elemente, Design Regeln und anderes besitzen. Komplette Schaltpläne und/oder Leiterplatten können ebenfalls als Vorlagen für neue Projekte eingebunden werden.

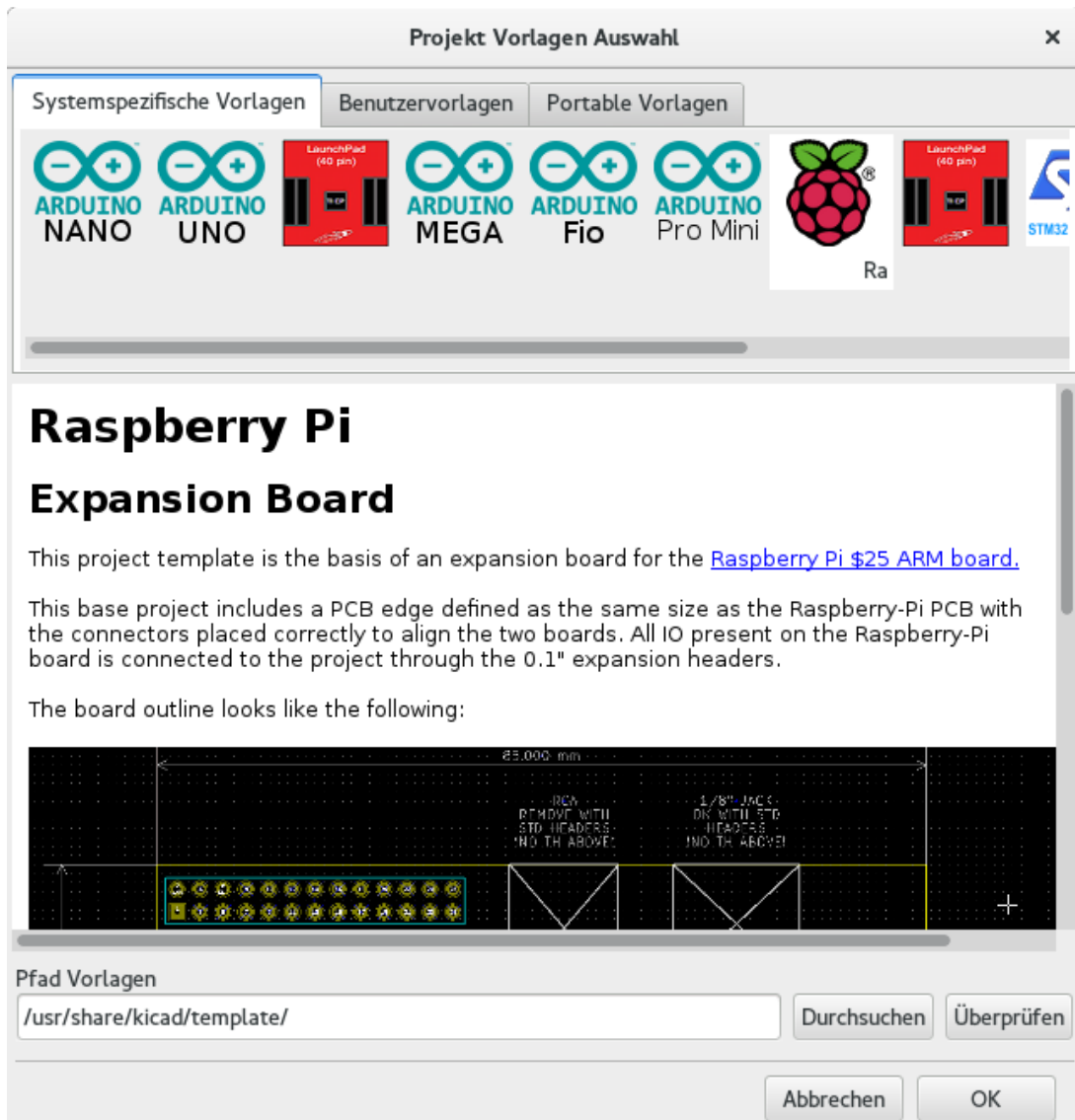
4.1 Benutzung von Vorlagen

Das Menü *Datei* → *Neues Projekt* → *Projekt aus einer Vorlage* wird den Dialog zur Auswahl der Projektvorlage öffnen:



Ein Einfachklick auf ein Icon einer Vorlage wird die Information der Vorlage laden, ein weiter Klick danach auf den OK Button erstellt ein neues Projekt auf Basis der Vorlage. Die Dateivorlagen werden in den Ablageort für das neue Projekt kopiert und auf Grundlage des zuvor neu gewählten Projektname umbenannt.

Nach der Auswahl einer Vorlage:



4.2 Ablageorte von Vorlagen:

Die Liste der möglichen Vorlagen kann man sich aus folgenden Quellen zusammenstellen:

- Systemweite Vorlagen: <kicad bin dir>/../share/template/ oder auch /usr/share/kicad/template/
- Benutzer definierte Vorlagen:
 - Unter Linux: ~/kicad/templates/
 - Unter Windows: C:\Dokumente und Einstellungen\Benutzer\Meine Dokumente\kicad\templates

- Unter Apple Mac OS: ~/Documents/kicad/templates/
- Wenn die Umgebungsvariable KICAD_PTEMPLATES definiert ist, dann gibt es einen weiteren Ablageort: Portable Vorlagen, die Liste der Vorlagen, welche im Pfad von KICAD_PTEMPLATES gefunden werden.

4.3 Erstellen von Vorlagen

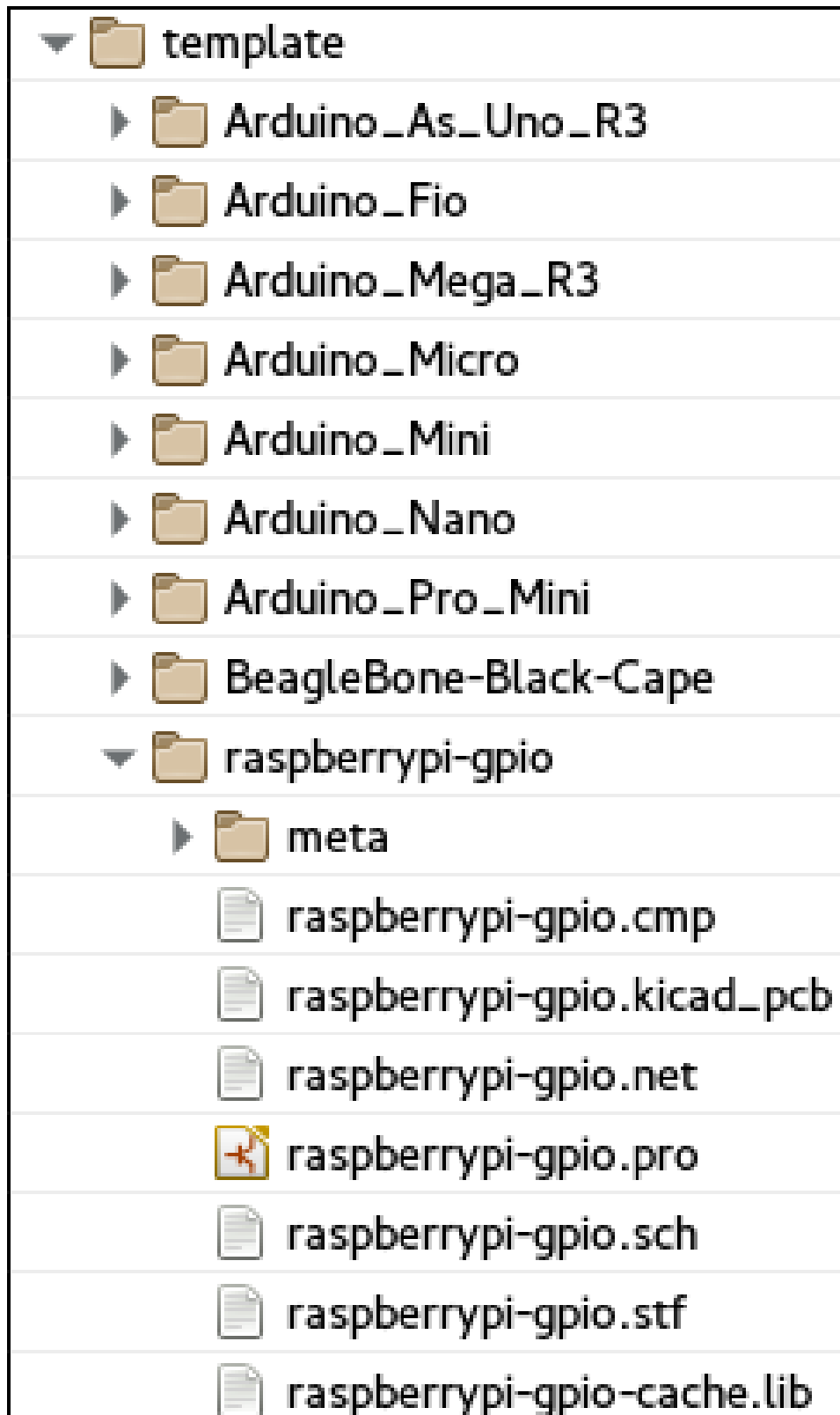
Der Vorlagenname ist das Verzeichnis, in dem die Datei-Vorlagen gespeichert sind. Das Verzeichnis mit den Metadaten, **meta** genannt, enthält vordefinierte Dateien, die Informationen über die Vorlage enthalten.

Alle Dateien und Ordner in einer Vorlage werden in den neuen Projektordner kopiert, sofern ein neues Projekt unter Verwendung dieser Vorlage erstellt wird. Davon ausgenommen ist das Verzeichnis **meta**.

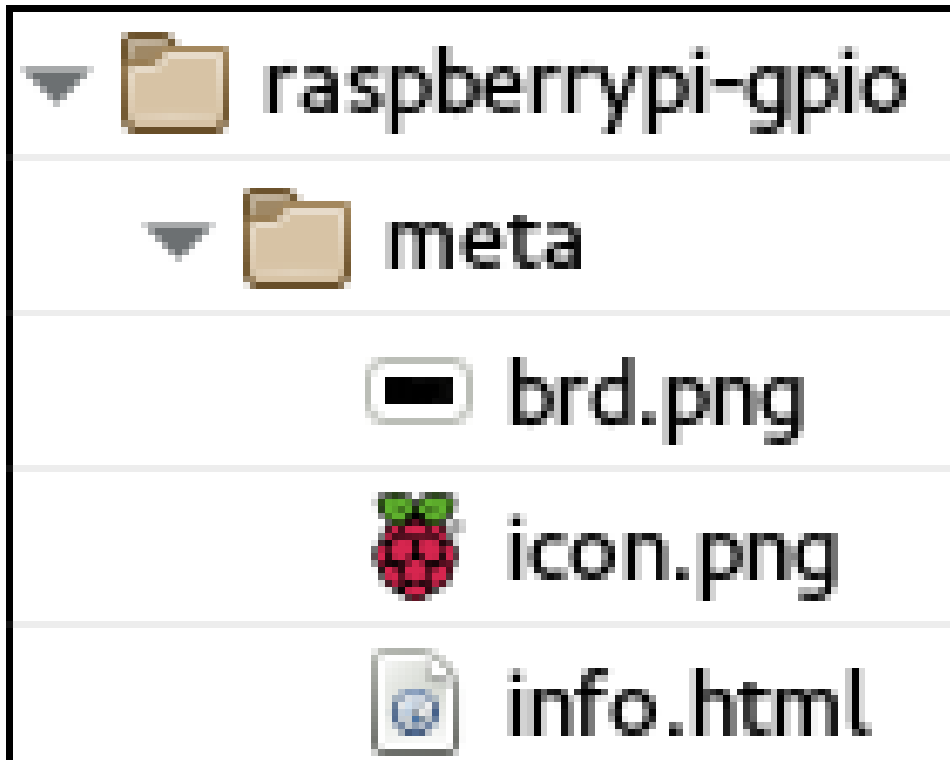
Alle Verzeichnisse und Dateien, die mit einem Vorlagenamen starten, werden diesen durch den Namen des Projektes ersetzt bekommen, mit Ausnahme der Dateierweiterung.

Die Metadaten bestehen aus minimal einer benötigten Datei, weitere optionale Dateien sind aber möglich. Alle Dateien müssen durch den Benutzer unter Verwendung eines Texteditor erstellt werden oder durch Abändern anderer vorhandener KiCad Projektdateien. Die Dateien müssen passend in der Verzeichnisstruktur abgelegt werden.

Hier sind Projektdateien für das Raspberry-GPIO Board als Vorlage abgelegt:



Und die Dateien mit den Metadaten:



4.3.1 Benötigte Dateien:

meta/info.html	HTML formatierte Information, welche die Vorlage beschreibt.
----------------	--

Der <title> Tag bestimmt den aktuellen Namen der Vorlage, die dem Benutzer in der Auswahl angezeigt wird. Beachten Sie, dass der Name der Projektvorlage abgeschnitten wird, sollte dieser zu lang sein. Durch die Eigenschaften des Schriftfonts können typischer Weise bis zu sieben oder acht Buchstaben angezeigt werden.

Durch die Benutzung von HTML ist es möglich, dass Bilder einfach eingebunden werden können, ohne das neue Schemata erstellt werden müssen. Es können nur Basis HTML Tags in diesem Dokument benutzt werden.

Und hier das Beispiel für eine Datei **info.html**:

```
<!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" expansion headers. [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 Optionale Dateien:

meta/icon.png	Eine PNG Grafik mit 64x64 Pixel, welche als anklickbares Icon im Auswahldialog der Vorlagen verwendet wird.
---------------	---

Jede weitere Grafik, die in **meta/info.html** benutzt wird, wie zum Beispiel das Bild des Leiterplattendesign von oben, wird ebenfalls in diesem Ordner abgelegt.