

KiCad

19. Oktober 2016

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 einiger Optionen und Hilfsmittel . . . 5

2.4 Konfiguration der Pfade . . . 5

2.5 Voreinstellung des Texteditors . . . 6

2.6 Voreinstellung des PDF-Betrachter . . . 6

2.7 KiCad: Prinzipien der Benutzung . . . 7

3 Benutzung des KiCad Managers **8**

3.1 Kicad Hauptfenster . . . 8

3.2 Schnellstart-Panel der Hilfstools . . . 9

3.3 Die Projektübersicht . . . 9

3.4 Obere Toolbar . . . 10

4 Benutzung von Vorlagen **11**

4.1 Definitionen . . . 11

4.2 Vorlagen . . . 11

4.2.1 Metadaten . . . 11

4.2.2 Benötigte Dateien: . . . 11

4.2.3 Optionale Dateien: . . . 12

4.2.4 Beispiel: . . . 12

4.2.5 Benutzung . . . 15

4.2.6 Ablageorte von Vorlagen: . . . 17

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 (http://www.gnu.org/licenses/gpl.html), Version 3 oder später, oder der Creative Commons Attribution License (http://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

2015, Mai 21.

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
- **CvPcb** : Footprint Auswahl Hilfe (von Eeschema aus gestartet)
- **Pcbnew** : Designer für gedruckte Leiterplatten und Footprinteditor
- **GerbView** : Gerber Betrachter

3 weitere Hilfstools

- **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.
- **PlEditor**: 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 gedruckte Schaltungen erstellen zu lassen:

- Gerber Dateien für Foto-Plotter
- Bohrdateien
- Bauteilpositionsdateien
- Und viele mehr.

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 OS X (noch experimentell, funktioniert aber aktuell gut).

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 allen 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.
fp-lib-table	Liste der Footprint Bibliotheken (*Footprint Bibliotheks-Tabelle*): Liste der Footprint Bibliotheken (verschiedene Formate) welche vom Platinen oder Footprint Editor oder auch CvPcb geladen wird.

Gemeinsam genutzte Dateien:

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

*.net	Netzliste die vom Schaltplanditor 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	Speichert die Zuordnung zwischen Bauteilen aus dem Schaltplan und deren zugehörigen Footprints. Diese Dateien können von Pcbnew erstellt und von Eeschema importiert werden. Die Idee hinter dieser Art der Zuordnung liegt in der 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 *Austausch Footprint*) 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

Die Datei *kicad.pro* wird als Standard Konfiguration im Verzeichnis *kicad/template* ausgeliefert. Diese Datei dient als Template für jedes neues Projekt.

Wenn eine andere Standard Konfigurationsdatei, *fp-lib-table* benannt, 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.)

Die Vorlagendatei *kicad.pro* kann (wenn nötig) einfach modifiziert werden, hauptsächlich um die Liste der Bibliotheken festzulegen die von Eeschema geladen wird.

Es gibt einige weitere Parameter (Standard Textgröße, Standard Breite von Linien, hauptsächlich für Pcbnew) die hier gespeichert werden.

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

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

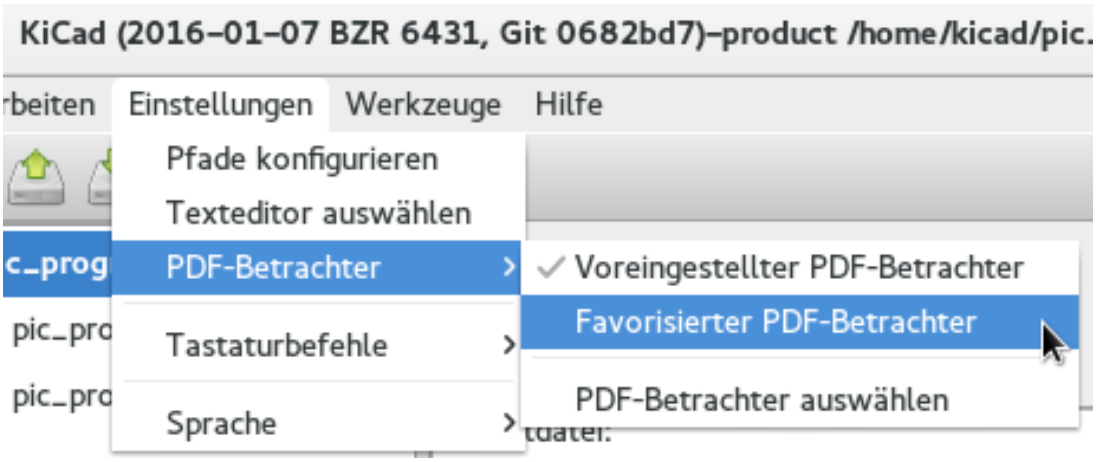
Starten Sie Eschema via KiCad Projekt Manager. Verändern und aktualisieren Sie die Eeschema Konfiguration, und vorzugsweise die Liste der Bibliotheken die Sie jedes mal benutzen wollen wenn Sie ein neues Projekt erstellen.

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* benannt. Die erste (im Heimatverzeichnis des Benutzers liegend) ist global für alle Projekte gültig. Die zweite Datei (wenn vorhanden und im Projektverzeichnis abgelegt) ist nur das Projekt gültig.

2.3 Vorbereitung einiger Optionen und Hilfsmittel

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

Die Einstellungen sind erreichbar über das Menü Einstellungen



3 Optionen die im Speziellen bedeutend sind:

- Pfade konfigurieren
- PDF-Betrachter
- Texteditor auswählen

2.4 Konfiguration der Pfade

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

Dies ist besonders hilfreich die absoluten Pfade nicht bekannt sind oder auch Gegenstand der notwendigen Veränderung sind. Dies trifft zum Beispiel auf die „offiziellen“ für KiCad erstellten Bibliotheken zu:

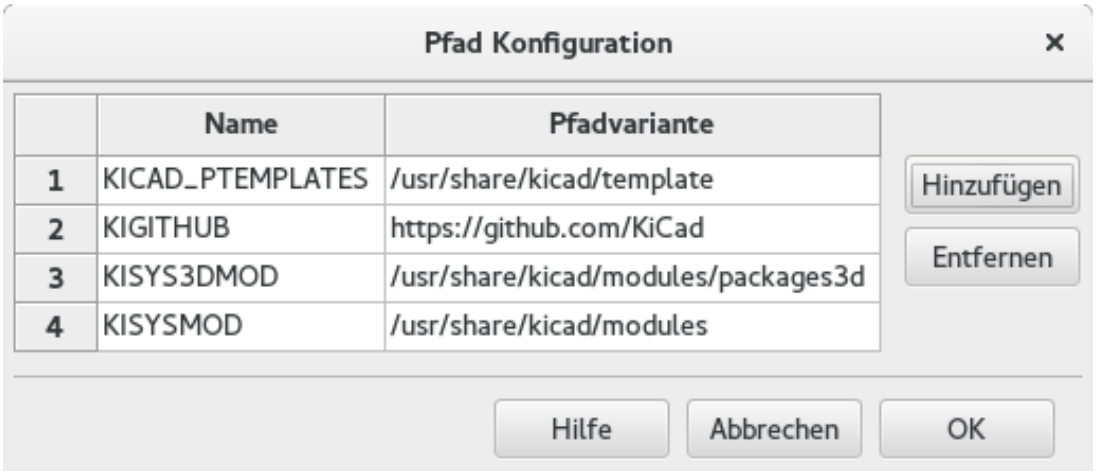
- der Pfad zu diesen Bibliotheken, wenn auf der Festplatte installiert
- den Pfad der Dateien mit den 3D-Formen der in den Footprint Definitionen benutzt wird.

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

Natürlich kann auch die übliche volle Pfad Definition benutzt werden, wenn dieser Pfad bekannt ist und sich niemals ändert.

Diese Optionen erlauben es einige Pfade durch Umgebungsvariablen zu setzen als auch eigene Umgebungsvariablen zu benutzen, und wenn nötig, eigene Pfadangaben zu benutzen.

KIGITHUB	Häufig benutzt in Beispielen der Footprint Bibliothek-Tabellen. Wenn Sie diese Variable benutzen muss sie definiert sein.
KISYS3DMOD	Standard Pfad für 3D-Formen, muss definiert sein da ein absoluter Pfad üblicher Weise 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, dass die Umgebungsvariable

- **KIPRJMOD**

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 möglichen Fehler bei der Pfad Benutzung zu verhindern.**

2.5 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. Das Menü:

Einstellungen/Texteditor auswählen

erlaubt es Ihnen einen beliebigen Texteditor auszuwählen.

2.6 Voreinstellung des PDF-Betrachter

Sie können den Standard PDF-Betrachter benutzen oder einen anderen PDF-Betrachter einstellen.

Gehen Sie zu *Einstellungen/PDF-Betrachter/Favorisierter PDF-Betrachter* um einen speziellen (Standard) PDF-Betrachter auszuwählen.

Ihr eigener (nicht Standard) Betrachter muss über das Menü *Einstellungen/PDF-Betrachter/PDF-Betrachter auswählen* eingestellt werden.

Unter Linux kann der Standard PDF-Betrachter manchmal Schwierigkeiten bereiten, die Auswahl und Benutzung von *Favorisierter PDF-Betrachter* nach der Auswahl des passenden PDF-Betrachter ist dann notwendig.

2.7 KiCad: Prinzipien der Benutzung

Die Verwaltung eines KiCad Projektes: 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 benutzen 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.xxx	Weitere verschiedene Dateien durch andere Tools erstellt.
example-cache.lib	Bibliotheksdatei die automatsich erstellt und vom Schaltplaneditor benutzt wird. (Diese Datei enthält Backups der Bauteile die im Schaltplan benutzt werden).

Kapitel 3

Benutzung des KiCad Managers

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

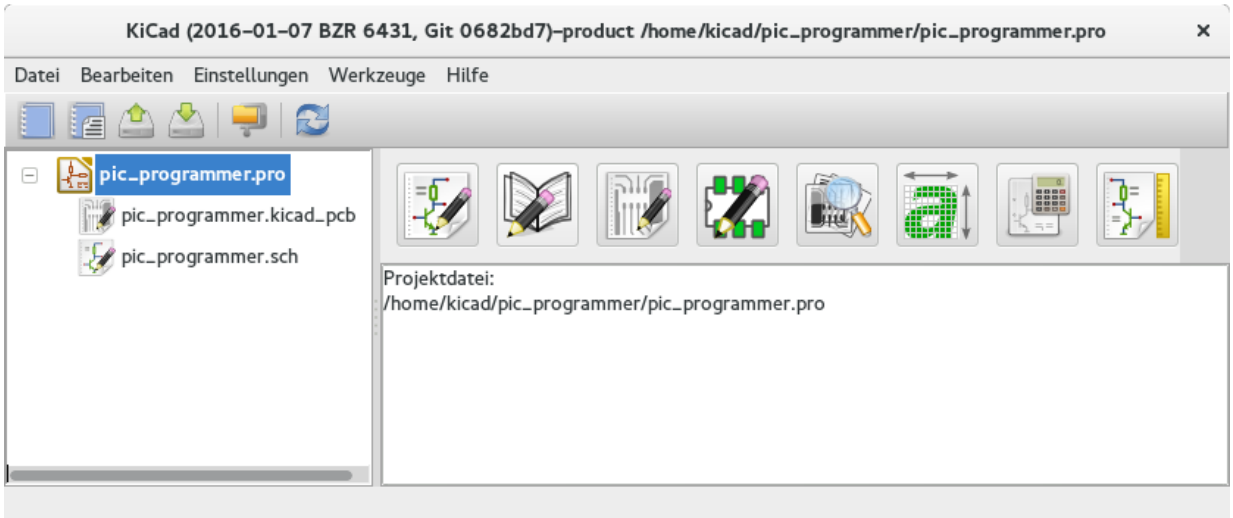
Das Starten der jeweiligen Tools aus dem KiCad Manager heraus hat einige Vorteile:

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

Sie können aber nur an einem Projekt zeitgleich arbeiten.

(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

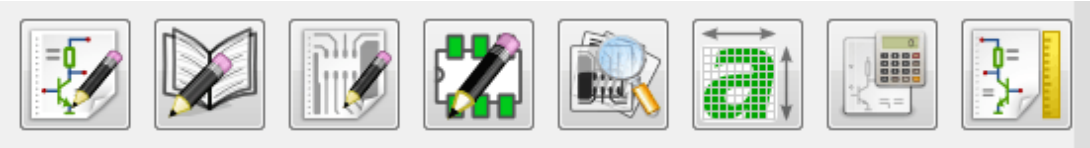


Das Hauptfenster im KiCad Projekt Manager zeigt die einzelnen Dateien des Projektes in einem Projektbaum dargestellt, ein Schnellstart-Panel mit den einzelnen Icons für die unterschiedlich Tools und ein Fenster für Meldungen. Das Menü als auch die Toolbar kann benutzt werden um Projekte zu erstellen,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

3.2 Schnellstart-Panel der Hilfstools

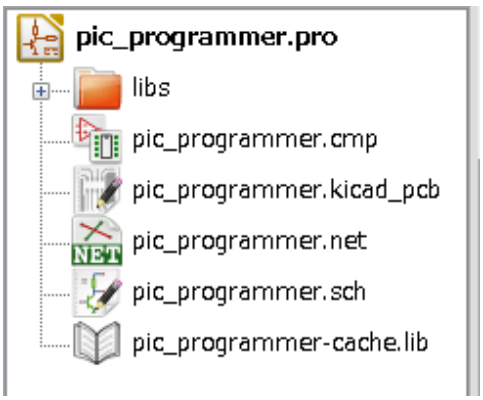
Der KiCad Projekt Manager erlaubt es Ihnen alle einzelnen Software Tools zu starten die zusammen mit KiCad verteilt werden.

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	**Pl Editor**	Seiten Layout Editor, erstellt oder passt Layouts an.

3.3 Die Projektübersicht



- Ein Doppel-Klick auf das Eeschema Icon startet den Schaltplaneditor welcher in diesem Fall die Datei *pic_programmer.sch* öffnen wird.
- Ein Doppel-Klick 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 (von Links nach Rechts).

	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

Benutzung von Vorlagen

4.1 Definitionen

Eine Vorlage ist ein Ordner mit Dateien, welche einen Ordner mit Metadaten inkludieren.

Der Vorlagen Systemname (SYSNAME) ist das Verzeichnis in dem die Datei-Vorlagen gespeichert sind. Das Verzeichnis mit den Metadaten enthält vordefinierte Dateien die Informationen über die Vorlage enthalten.

Alle Dateien und Ordner aus einer Vorlage werden in den neuen Projektordner kopiert wenn ein neues Projekt unter Verwendung dieser Vorlage erstellt wird. Davon ausgenommen ist das Verzeichnis METADIR.

Alle Verzeichnisse und Dateien die in SYSNAME liegen werden SYSNAME durch den neuen Namen des Projektes ersetzt bekommen, mit Ausnahme der Dateierweiterung.

4.2 Vorlagen

Vorlagen erleichtern das Setup eines Projektes welche gemeinsam benutzte Attribute wie vordefinierte Platinengrößen, Verbinder Positionen, Schaltplan Elemente, Design Regeln und anderes besitzen.

4.2.1 Metadaten

Das MEATDIR einer Vorlage muss die benötigten Dateien enthalten, und kann andere optionale Dateien enthalten.

4.2.2 Benötigte Dateien:

meta/info.html

Enthält HTML formatierte Informationen über die Vorlage, welche vom Benutzer benutzt werden kann um festzustellen, ob die Vorlage das ist was der Anwender benutzen möchte. Der <title> Tag bestimmt den aktuellen Namen der Vorlage welcher dem Benutzer angezeigt wird bei der Auswahl der Vorlagen.

Durch die Benutzung von HTML ist es möglich, dass Bilder einfach eingebunden werden können ohne das neue Schema erweitert zu müssen.

Es können nur Basis Elemente der HTML Sprache verwendet werden um das Dokument zu formatieren.

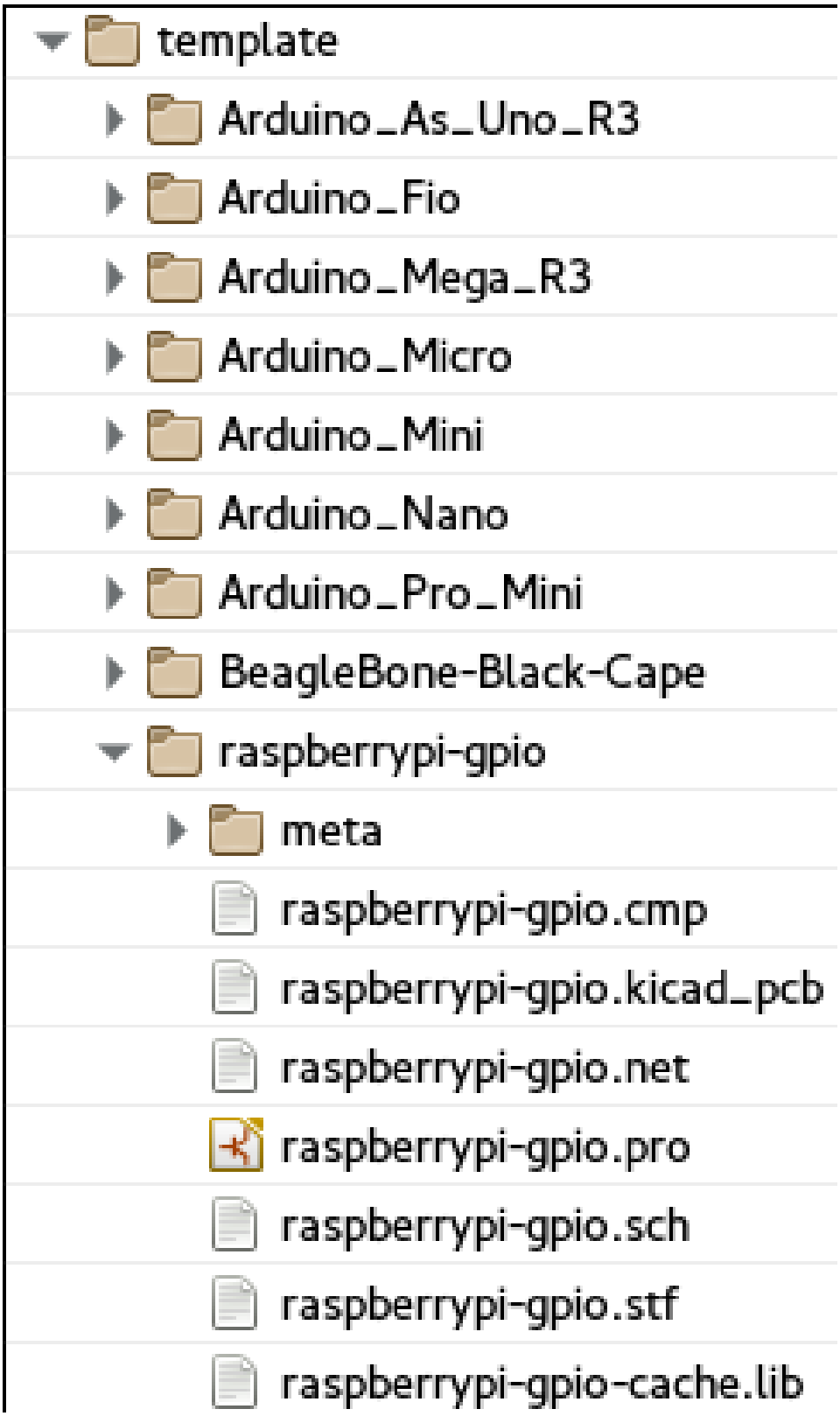
4.2.3 Optionale Dateien:

meta/icon.png

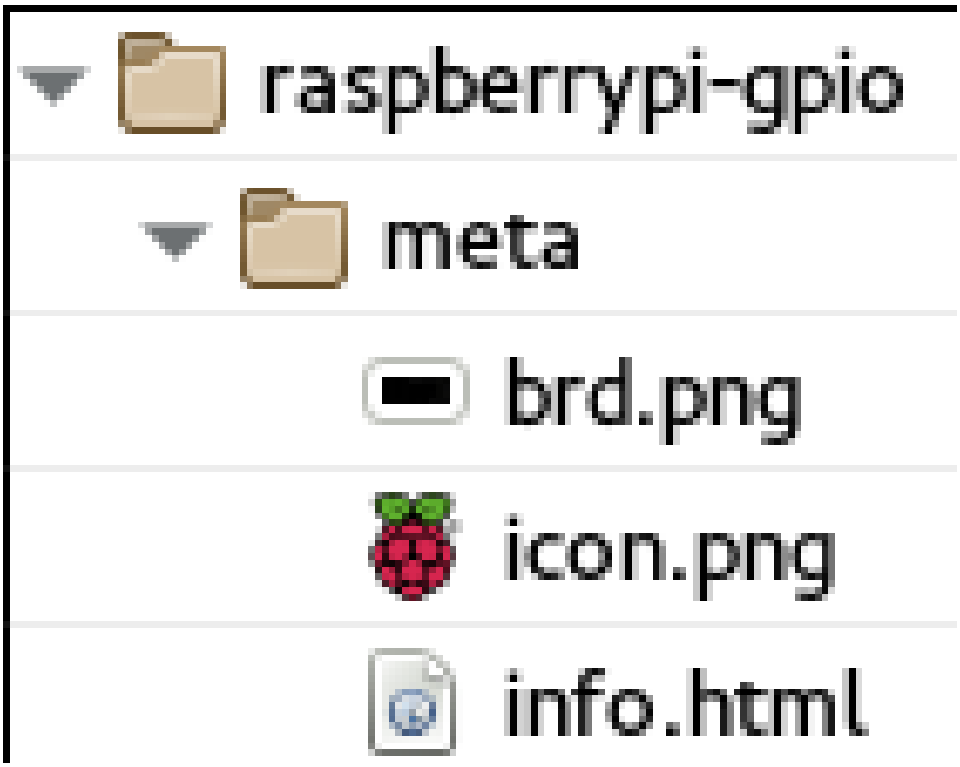
Eine PNG Grafik mit 64x64 Pixel welches als anklickbares Icon im Auswahldialog der Vorlagen verwendet wird.

4.2.4 Beispiel:

Hier ist die Vorlage für das Raspberry-GPIO Board abgelegt:



Und die Informationen mit den Metadaten:



brd.png ist einen optionale Datei.

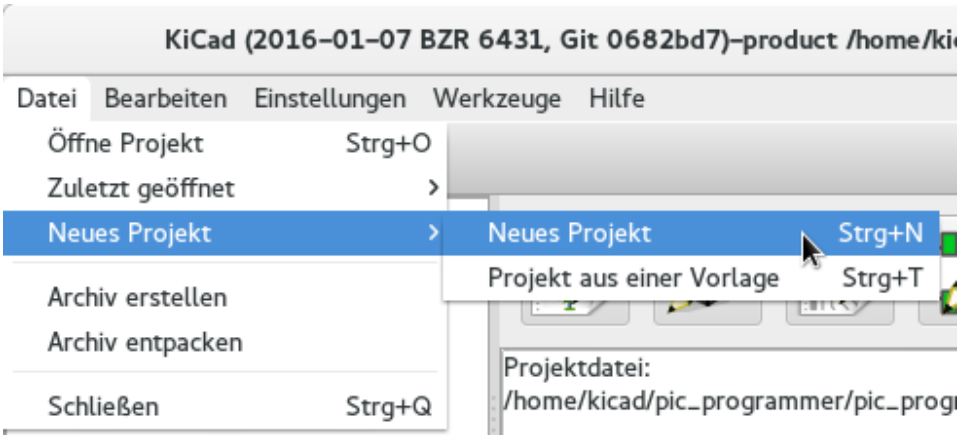
Und hier das Beispiel für eine Datei *info.html*.

```
<!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&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>
```

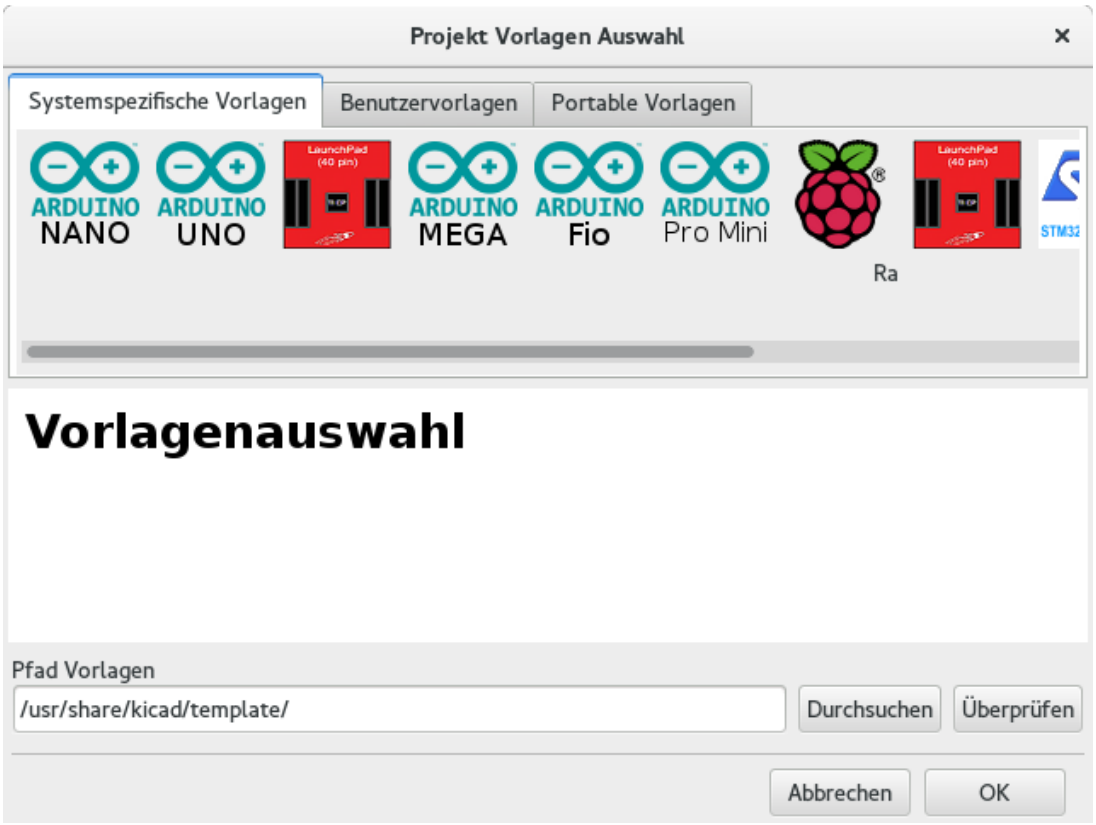
```
</HTML>
```

4.2.5 Benutzung

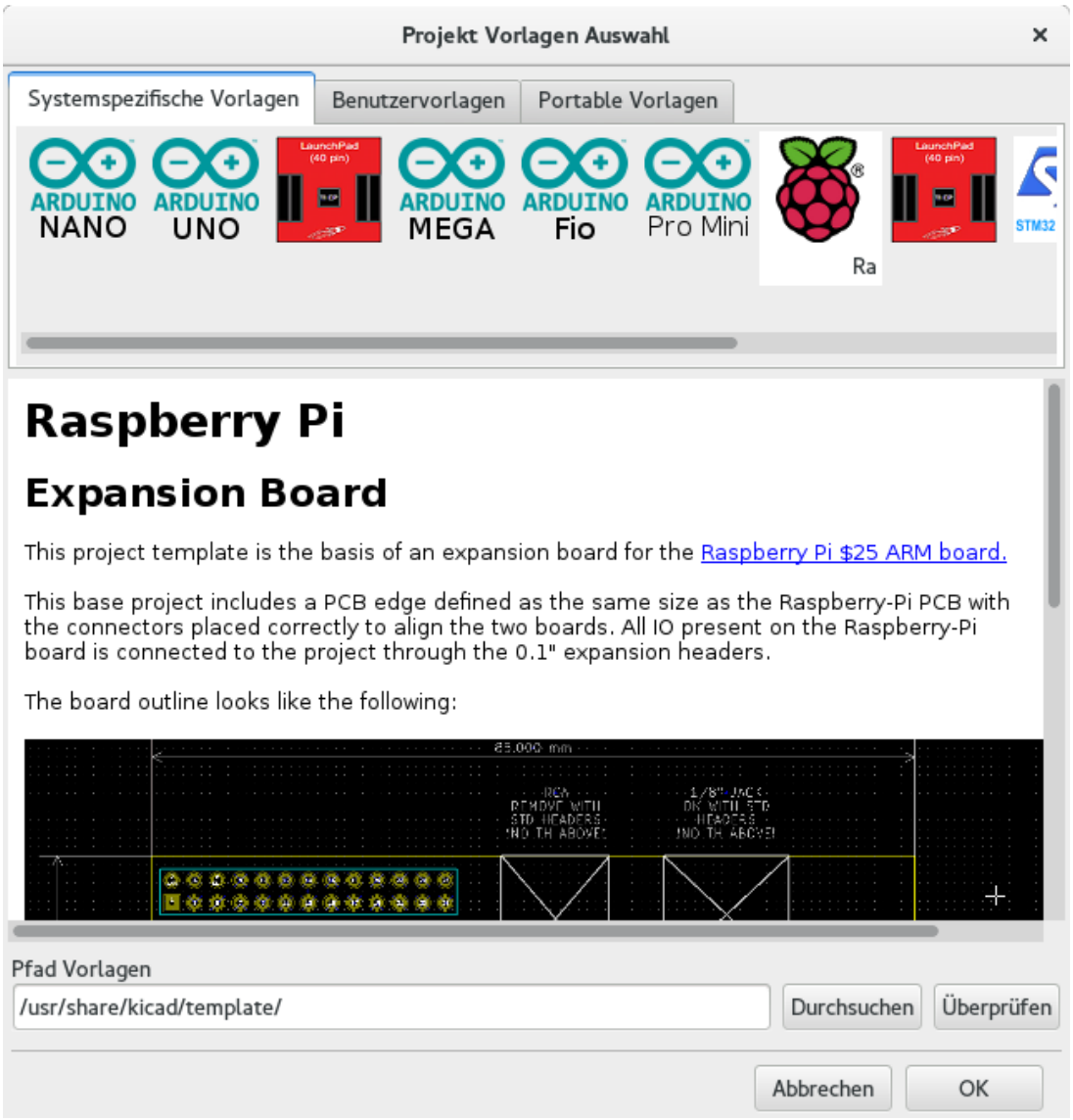
Das KiCad Datei Menü *Neues Projekt* enthält zwei Optionen:



- **Neues Projekt** Erstellt ein neues leeres Projekt durch kopieren von template/kicad.pro in den aktuellen Ordner.
- **Projekt aus einer Vorlage** Öffnet den Auswahldialog zur Auswahl einer Vorlage, nachdem der Projektordner ausgewählt und ggfls. angelegt worden ist. Der Auswahldialog für die Vorlagen hat eine Liste mit Icons und ein Anzeigefenster. Ein Klick auf eines der Icons der Vorlagen lädt die Datei info.html und zeigt deren Inhalt im Anzeigefenster an. Ein Betätigen des OK Buttons startet die Erstellung des neuen Projektes. Die Vorlagen werden in den Projektordner kopiert (ohne den Inhalt von METADIR wie zuvor erläutert) und jede Datei deren Name unter die Regeln der automatischen Ersetzung fällt wird unter Verwendung des Projektnamens umbenannt und abgespeichert.



Nach der Auswahl einer Vorlage:



4.2.6 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:\Documents and Settings\username\My Documents\kicad\templates
  - Unter Apple OS X: ~/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.