

Комплекс программ для разработки электрических схем и печатных плат
с открытым кодом и свободной лицензией

KiCAD

Сборка комплекса программ

Руководство программиста

0XXXXX.000YY-01 33 01

Листов 16

Содержание

1 НАЗНАЧЕНИЕ ПРОГРАММНОГО КОМПЛЕКСА KICAD	3
2 СБОРКА KICAD В ОПЕРАЦИОННОЙ СИСТЕМЕ MS WINDOWS.....	4
2.1 Подготовка к сборке KICAD для WINDOWS.....	4
2.1.1 Установка технологической среды MinGW	4
2.1.2 Установка системы MSYS.....	4
2.1.3 Установка GUI-библиотеки WxWidgets.....	5
2.1.4 Установка библиотеки сжатия данных Zlib.....	5
2.1.5 Установка программы-конфигуратора CMAKE.....	6
2.1.6 Установка потоковой библиотеки Boost	6
2.2 Трансляция и сборка комплекса программ KICAD.....	6
2.3 Главное окно менеджера KICAD для WINDOWS XP.....	9
3. СБОРКА KICAD В ОПЕРАЦИОННОЙ СИСТЕМЕ ALT LINUX.....	10
3.1 Подготовка к сборке KICAD в LINUX.....	10
3.1.1 Установка программы-конфигуратора CMAKE.....	10
3.1.2 Установка библиотеки сжатия данных Zlib.....	11
3.1.3 Установка GUI-библиотеки WxWidgets.....	11
3.1.4 Установка потоковой библиотеки Boost.....	12
3.2 Трансляция и сборка комплекса программ KICAD.....	12
3.3 Главное окно менеджера KICAD для ALT LINUX.....	15
4 РЕДАКТОР ПЕРЕВОДОВ ДЛЯ РУСИФИКАЦИИ ИНТЕРФЕЙСА KICAD.....	15

1 Назначение программного комплекса KiCAD

KiCAD – сквозная система автоматизированного проектирования многослойных печатных плат от разработки схемы электрической принципиальной и моделирования до выхода на технологическое оборудование (Gerber-файлы фотошаблонов, Excellon-файлы отверстий и др.).

KiCAD – комплекс прикладных программ с открытыми кодами и общедоступной лицензией. Комплекс развивается KiCAD-сообществом программистов и пользователей под руководством Жан-Пьера Шарра (Франция). Поддержка разработки осуществляется через svn-репозиторий хост-сервера sourceforge сети Интернет. Международным языком сопровождения разработки служит английский язык.

Адаптация KiCAD для России выполняется программистами из KiCAD Russian Team. Для кодировки кириллицы используется Юникод.

Комплекс программ (КП) KiCAD располагает русскоязычным интерфейсом для пользователя, возможностью работы в отечественной лицензионной операционной системе ALT Linux и эксплуатационной документацией для пользователя на русском языке. Эксплуатация КП KiCAD возможна и на ПЭВМ с Windows.

При разработке, сборке и поддержке САПР KiCAD используются открытые программные продукты (трансляторы для кода и меню-интерфейса, графические интерфейсные библиотеки, офисные пакеты, инсталляторы, средства доступа к системам хранения исходного кода и отслеживания изменений).

В качестве библиотек электронных компонентов (в папках **/library** и **/modules**) возможна подстановка наработанных ранее библиотек, конвертированных из формата системы P-CAD.

Для подготовки файлов эксплуатационной документации в форматах Adobe PDF и ODT использован текстовый редактор OpenOffice Writer со свободной лицензией.

Для подстановки русского меню-интерфейса использован открытый редактор переводов Poedit.

Для установки КП KiCAD на ПЭВМ пользователей разработан инсталляционный скрипт для транслятора NSIS со свободной лицензией.

2 Сборка KICAD в операционной системе MS Windows

2.1 Подготовка к сборке KICAD для Windows

2.1.1 Установка технологической среды MinGW

MinGW (от [англ.](#) Minimalist [GNU](#) for [Windows](#)) — общедоступный кросс-компилятор графических и консольных программ на C/C++ и других языках в Windows (порт Linux GCC), позволяющий строить “родные” для Windows программы, используя Linux-подобные инструменты.

MinGW устанавливается простым копированием содержимого архивов в созданную вами директорию C:\MinGW, или любую другую. Распаковываем в папку C:\MinGW следующие архивы:

- **gcc-core-3.4.5-20060117-3.tar.gz** – ядро транслятора GCC 3.4.5
- **gcc-g++-3.4.5-20060117-3.tar.gz** – транслятор C++
- **binutils-2.19.1-mingw32-bin.tar.gz** – набор открытых утилит (линковщик, ассемблер, архиватор и т.д.)
- **mingwrt-3.15.2-mingw32-dev.tar.gz** и **mingwrt-3.15.2-mingw32-dll.tar.gz** – run-time библиотеки
- **w32api-3.13-mingw32-dev.tar.gz** - библиотеки и заголовочные файлы интерфейса WinAPI

Собственно компилятор установлен. Теперь дописываем к переменной окружения PATH путь до бинарных файлов компилятора C:\MinGW\bin. Для этого открываем **Панель управления – Система – Дополнительно – Переменные среды**. Чтобы проверить правильность внесенных изменений, вызываем Windows-консоль **cmd** и пишем команду gcc. Если ответ – «gcc: no input files», значит все нормально. Сайт среды MinGW в Интернет - <http://mingw.org/>.

Как вариант, можно использовать среду MinGW, идущую вместе со свободным инструментарием Qt Creator и библиотекой Qt от компании Nokia.

2.1.2 Установка системы MSYS

MSYS (от [англ.](#) Minimal System) – минимальная POSIX-система, используемая в ОС Win32 для выполнения конфигураций при сборке пакетов программ. Проще говоря, MSYS – это “Unix в Window’s”. Для установки MSYS нужно скачать из Интернет следующие архивы:

- **MSYS-1.0.10.exe** – MSYS

<http://downloads.sourceforge.net/mingw/MSYS-1.0.10.exe>

- **msysDTK-1.0.1.exe** - инструментарий разработки

<http://downloads.sourceforge.net/mingw/msysDTK-1.0.1.exe>

- **msysCORE-1.0.11-2007.01.19-1.tar.bz2** – обновление

<http://downloads.sourceforge.net/mingw/msysCORE-1.0.11-2007.01.19-1.tar.bz2>

Инсталлятор попросит установиться в C:\msys\1.0 - разрешаем. После копирования файлов выводится консольное окошко, в котором задается вопрос “установлен ли MinGW и где он располагается?”. В ответ на запрос пути к MinGW нужно написать **c:/mingw**. Далее запускаем инсталлятор msysDTK-1.0.1.exe, который вопросов не задает, просто копирует все что нужно. Обновляем MSYS до версии 1.0.11, распаковав msysCORE-1.0.11-2007.01.19-1.tar.bz2 в C:\MSYS\1.0, с заменой всех файлов.

В процессе установки шрифт консоли получается очень мелким и почти не читаемым. Правим его! В C:\MSYS\1.0\msys.bat находим строчку:

Заменяем в ней размер шрифта Courier-12 на Courier-16, например.

2.1.3 Установка GUI-библиотеки WxWidgets

WxWidgets – развитая кросс-платформенная C++ библиотека для написания графических интерфейсов приложений для пользователя (GUI – Graphic User Interface).

Для установки WxWidgets необходимо решить, где будет находиться директория построения wxWidgets. Она должна находиться в директории, для которой у вас будет доступ через среду MSYS (например, home/<user>/wx2810). Распаковываем содержимое архива с исходным кодом wxWidgets-2.8.10 в папку wx2810, и создаем там рабочую папку build_release. Переходим в эту директорию с помощью команды MSYS:

```
$ cd c:/msys/1.0/home/<user>/wx2810/build_release
```

Затем, создаем управляющий файл makefile для трансляции и сборки библиотеки с помощью следующих последовательностей опций:

```
$ ../configure --enable-unicode --disable-debug --enable-shared --disable-monolithic --with-opengl --with-odbc (динамические разделяемые библиотеки с юникодом)
```

либо:

```
$ ../configure --enable-unicode --disable-debug --disable-shared --enable-monolithic --with-opengl --with-odbc (статическая монолитная библиотека)
```

Запускаем построение библиотеки.

```
$ make
```

```
$ make install
```

Сайт библиотеки Wx в Интернет -

<http://prdownloads.sourceforge.net/wxwindows/> .

2.1.4 Установка библиотеки сжатия данных Zlib

Распаковать архив zlib123.zip в папку *home/zlib*. Перейти в эту директорию с помощью команды MSYS:

```
$ cd c:/msys/1.0/home/zlib
```

\$./configure - создание makefile
 \$ make - построение zlib
 \$ make install
 Сайт библиотеки в Интернет - <http://www.zlib.net/zlib123.zip>

2.1.5 Установка программы-конфигуратора CMAKE

CMake – кросс-платформенный генератор файлов управления сборкой - **makefiles**. Установите cmake-2.6.4-win32-x86.exe и в процессе установки выберите режим **to add cmake to your path** (добавить cmake в переменные окружения).

2.1.6 Установка потоковой библиотеки Boost

Boost C++ Libraries - расширенные сервисные библиотеки для работы с потоками данных. Каждый поток имеет свою машинную область, включающую программный счетчик и регистры. Установка Boost C++ Libraries возможна 2 способами:

- Скачать http://sourceforge.net/project/showfiles.php?group_id=7586.
 Распаковать архив boost_1_34_1.zip в C:\Program Files\boost\boost_1_34_1
- Скачать http://www.boost-consulting.com/boost_1_34_1_setup.exe и установить

2.2 Трансляция и сборка комплекса программ KiCAD

- переписать свежую версию с исходным кодом системы KiCAD из репозитория KiCAD Team (берем стабильную или свежую рабочую svn-версию). Для этого скопировать тексты программ в папку kicad-svn и завести в ней рабочую подпапку build/release
 \$ cd c:/msys/1.0/home/<user>/kicad-svn/build/release
- использовать консольный вариант программы cmake (без GUI). Предварительно необходимо удалить файл CmakeCash.txt и папку CmakeFiles. Команда на подготовку к трансляции и формирование файла управления сборкой **makefile** следующая:

```
$ cmake -G "MSYS Makefiles" -DCMAKE_BUILD_TYPE=Release
-DwxWidgets_ROOT_DIR=c:\MSYS\1.0\home\<user>\wx2810\build_release
-DKICAD_CYRILLIC=ON -DKICAD_GOST=ON -DKICAD_PYTHON=ON
-DCMAKE_INSTALL_PREFIX=c:\MSYS\1.0\home\<user>\kicad-ins ../..
```

Важные параметры условной трансляции для Cmake:

- **DCMAKE_BUILD_TYPE=<buildtype>** - <buildtype> может быть "Debug" (отладочная версия) или "Release" (рабочая версия);
- **DwxWidgets_ROOT_DIR=<wxInstallDir>** - директория с библиотекой wxWidget;
- **DKICAD_CYRILLIC=ON** – строит KiCad с поддержкой кириллицы;
- **DKICAD_GOST=ON** – строит KiCad с поддержкой наших ГОСТов;
- **DKICAD_PYTHON=ON** – строит KiCad с поддержкой PYTHON;
- **DCMAKE_INSTALL_PREFIX=<prefix>** - папка выходных файлов.

```

MINGW32:/home/kicad_new/kicad-svn/build/release
stepanov@STEPANOV /home/kicad_new/kicad-svn/build/release
$ cmake -G "MSYS Makefiles" -DCMAKE_BUILD_TYPE=Release -DwxWidgets_ROOT_DIR=c:\
MSYS\1.0\home\stepanov\wx2810\build_release -DKICAD_CYRILLIC=ON -DKICAD_GOST=ON
-DKICAD_PYTHON=ON -DCMAKE_INSTALL_PREFIX=c:\MSYS\1.0\home\stepanov\kicad_new\k
icad-ins ../../
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: C:/Qt/qtcreator-1.2.0/mingw/bin/gcc.exe
-- Check for working C compiler: C:/Qt/qtcreator-1.2.0/mingw/bin/gcc.exe -- work
s
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: C:/Qt/qtcreator-1.2.0/mingw/bin/g++.exe
-- Check for working CXX compiler: C:/Qt/qtcreator-1.2.0/mingw/bin/g++.exe -- wo
rks
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for installed OpenGL -- found
-- Check for installed wxWidgets -- found
-- Looking for malloc.h
-- Looking for malloc.h - found
-- Looking for iso646.h
-- Looking for iso646.h - found
-- Looking for strings.h

```

Команда на сборку следующая:

\$ make

```

MINGW32:/home/kicad_new/kicad-svn/build/release
mmer.cpp.obj
[ 11%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/hel
lp.cpp.obj
[ 11%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/Hi
dden_Pin.cpp.obj
[ 11%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/Hi
erarchy_cursor.cpp.obj
[ 11%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/Hi
erarchy_Nav.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ho
tkeys.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_3d.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_cvpcb_small.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_cvpcb.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_eeschema.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_gerbview_small.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_gerbview.cpp.obj
[ 12%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/ic
on_kicad.cpp.obj

```

\$ make install

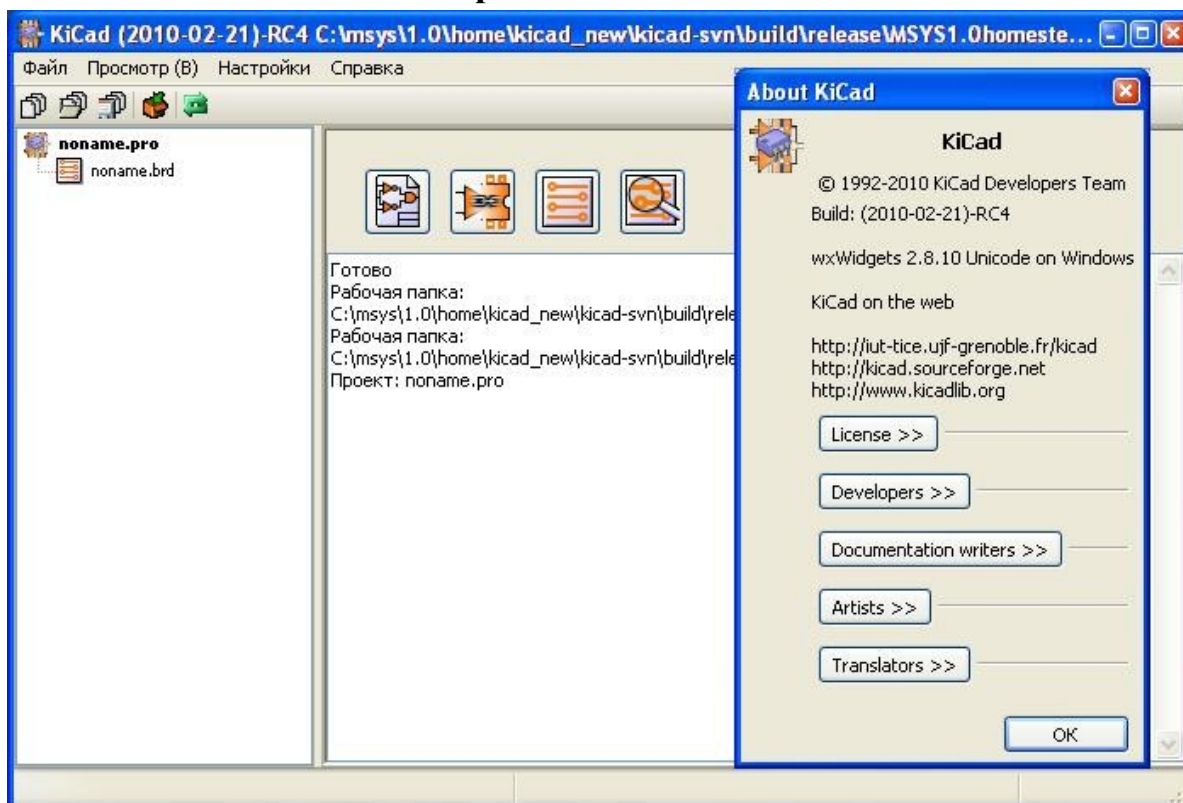
```

MINGW32:/home/kicad_new/kicad-svn/build/release
stepanov@STEPANOV /home/kicad_new/kicad-svn/build/release
$ make install
[ 0%] Built target 3d-viewer
[ 63%] Built target bitmaps
[ 67%] Built target common
[ 69%] Built target pcbcommon
[ 70%] Built target kbool
[ 71%] Built target polygon
[ 73%] Built target cvpcb
[ 83%] Built target eeschema
[ 83%] Built target netlist_form_pads-pcb
[ 86%] Built target gerbview
[ 87%] Built target kicad
[ 88%] Built target minizip
[100%] Built target pcbnew
Install the project...
-- Install configuration: "Release"
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/doc/INSTALL.txt
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/bin/freeroute.jnlp
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/bin/cvpcb.exe
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/bin/eeschema.exe
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/bin/plugins/netlist_form_
pads-pcb.exe
-- Installing: c:MSYS1.0homestepanovkicad_newkicad-ins/bin/gerbview.exe

```

Сборка комплекса программ KiCAD выполняется за время от 20 минут до часа в зависимости от мощности компьютера.

2.3 Главное окно менеджера KiCAD для Windows XP

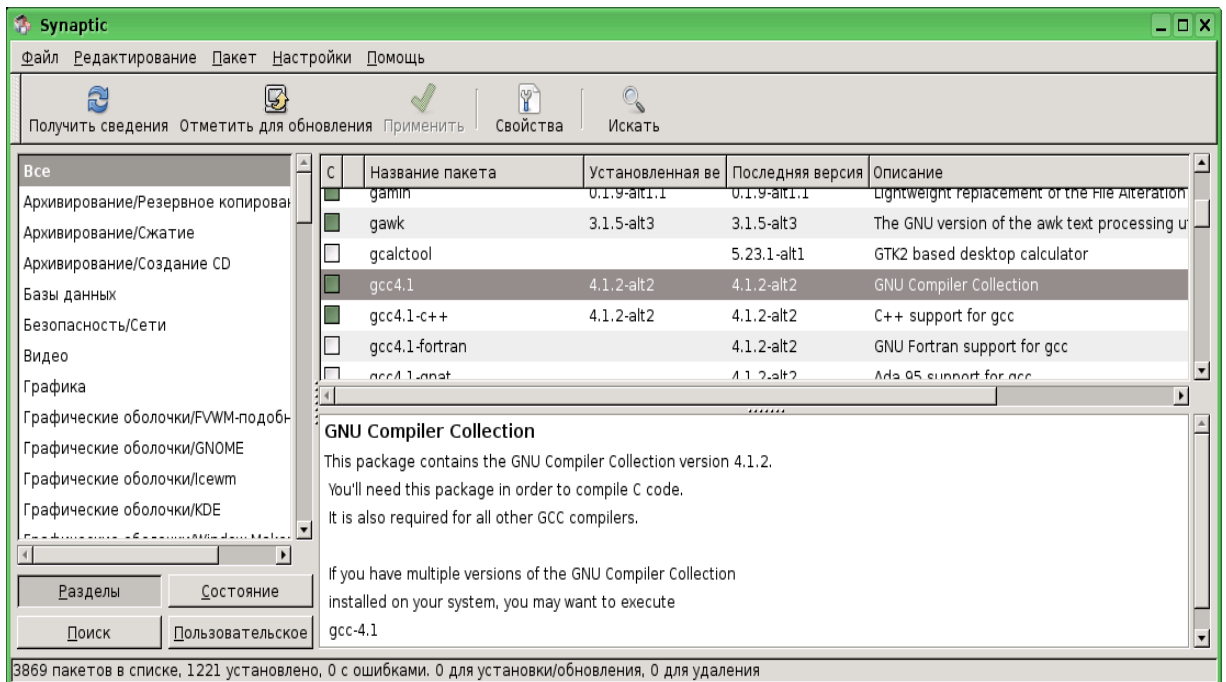


3. Сборка KICAD в операционной системе ALT Linux

3.1 Подготовка к сборке KICAD в Linux

3.1.1 Установка программы-конфигуратора CMAKE

Перед установкой CMake необходимо убедиться, что установлены свободные GPL-версии пакетов трансляторов с языков C/C++ (gcc4.1, gcc4.1-c++ и служебная программа make. Если они не установлены, то необходимо установить. Для этого в ALT Linux используется программа управления пакетами Synaptic:



Пакеты, отмеченные зеленым цветом — это установленные пакеты. Для того чтобы установить пакет, необходимо выделить его и, нажав правой кнопкой мыши, в появившемся окне выбрать пункт **Отметить для установки** и затем в главном окне программы нажать кнопку **Применить**.

ВАЖНО: ПАКЕТЫ НЕОБХОДИМО СТАВИТЬ ПРИ НАЛИЧИИ ДИСКА С ОС ALT LINUX В CD-R приводе!

Также существует альтернативный метод построения пакетов. На CD-диске необходимо найти нужный пакет, скопировать его в предварительно созданную папку. Затем через терминал зайти в эту папку и набрать команду *apt-get install <имя пакета>*.

Далее приступаем непосредственно к установке программы CMake (версия 2.8.0). Для установки необходим дистрибутив с открытым программным кодом **cmake-2.8.0-alt2.src.rpm**. Распакуем содержимое в созданную предварительно папку Cmake. Там будет еще один архив, который

тоже необходимо распаковать. Через терминал переходим в папку с распакованным содержимым и выполняем следующие команды:

```
$ mkdir Cmake/cmake-2.8.0
```

```
$ apt-get install
```

Затем, создаем управляющий файл `makefile` для трансляции и сборки библиотеки.

```
$ ./bootstrap
```

Если все прошло успешно, то запускаем построение библиотеки.

```
$ make
```

```
$ su root
```

```
Password: ****
```

```
$make install
```

Все! CMake установлен!

3.1.2 Установка библиотеки сжатия данных Zlib

Под Linux'ом `zlib` обычно установлен. Если это не так, то с помощью менеджера пакетов устанавливаем пакеты `zlib` и `zlib` для разработчика (`zlib-devel`).

3.1.3 Установка GUI-библиотеки WxWidgets

`WxWidgets` – развитая кросс-платформенная C++ библиотека для написания графических интерфейсов приложений для пользователя (GUI – Graphic User Interface).

Перед установкой `wxWidgets` должны быть установлены следующие вспомогательные пакеты: Linux-конфигуратор ***pkg-config*** (`pkg-config-0.23-alt3.i586.rpm`) и графические библиотеки нижнего уровня GTK+ версии 2 для разработчика ***libgtk+2*** (пакеты `libgtk+2-common-devel-2.12.12-alt0.M41.1.i586.rpm` и `libgtk+2-devel-2.12.12-alt0.M41.1.i586.rpm`). Их можно установить с помощью менеджера пакетов `Synaptic`.

Далее распаковываем содержимое архива с открытым программным кодом **`wxGTK-2.8.10.tar.gz`**, переходим в папку с содержимым и начинаем установку:

```
$ mkdir build_gtk
```

```
$ cd build_gtk
```

Создаем управляющий файл `makefile` для построения и сборки библиотеки.

```
$ ../configure --with-gtk --with-opengl --enable-unicode --enable-release
```

Построение библиотеки:

```
$ make
```

```
$ su root
```

```
Password:****
```

```
$ make install
```

3.1.4 Установка потоковой библиотеки Boost

Boost C++ Libraries - расширенные сервисные библиотеки потоков данных. Каждый поток имеет свою машинную область, включающую программный счетчик и регистры. С помощью менеджера пакетов устанавливаем boost-thread.

3.2 Трансляция и сборка комплекса программ KiCAD

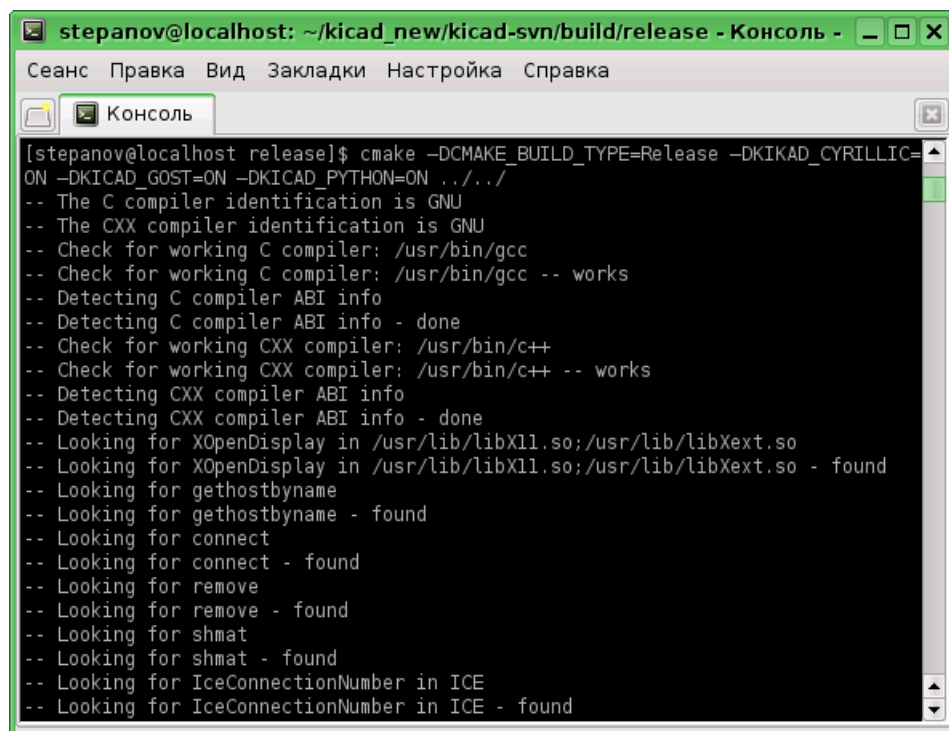
Распаковываем содержимое архива **kicad-svn-rXXXX-src**, где XXXX – текущая версия комплекса программ KiCAD. Переходим в этот каталог:

```
$ cd kicad-svn
$ mkdir build/release
$ cd build/release
```

Предварительно необходимо удалить файл CmakeCash.txt и папку CmakeFiles.

Команда на подготовку к трансляции и формирование файла сборки **makefile** следующая:

```
$ cmake -DCMAKE_BUILD_TYPE=Release -DKICAD_CYRILLIC=ON -DKICAD_GOST=ON -DKICAD_PYTHON=ON ../../
```



```
stepanov@localhost: ~/kicad_new/kicad-svn/build/release - Консоль -
Сеанс  Правка  Вид  Закладки  Настройка  Справка
Консоль
[stepanov@localhost release]$ cmake -DCMAKE_BUILD_TYPE=Release -DKICAD_CYRILLIC=ON -DKICAD_GOST=ON -DKICAD_PYTHON=ON ../../
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Looking for XOpenDisplay in /usr/lib/libX11.so;/usr/lib/libXext.so
-- Looking for XOpenDisplay in /usr/lib/libX11.so;/usr/lib/libXext.so - found
-- Looking for gethostbyname
-- Looking for gethostbyname - found
-- Looking for connect
-- Looking for connect - found
-- Looking for remove
-- Looking for remove - found
-- Looking for shmat
-- Looking for shmat - found
-- Looking for IceConnectionNumber in ICE
-- Looking for IceConnectionNumber in ICE - found
```

Параметры CMake:

- **-DCMAKE_BUILD_TYPE=<buildtype>** - <buildtype> может быть "Debug" или "Release";
- **-DKICAD_CYRILLIC=ON** – строит Kicad с поддержкой кириллицы;
- **-DKICAD_GOST=ON** – строит Kicad с поддержкой ГОСТов;
- **-DKICAD_PYTHON=ON** – строит Kicad с поддержкой PYTHON.

Команда на сборку следующая:

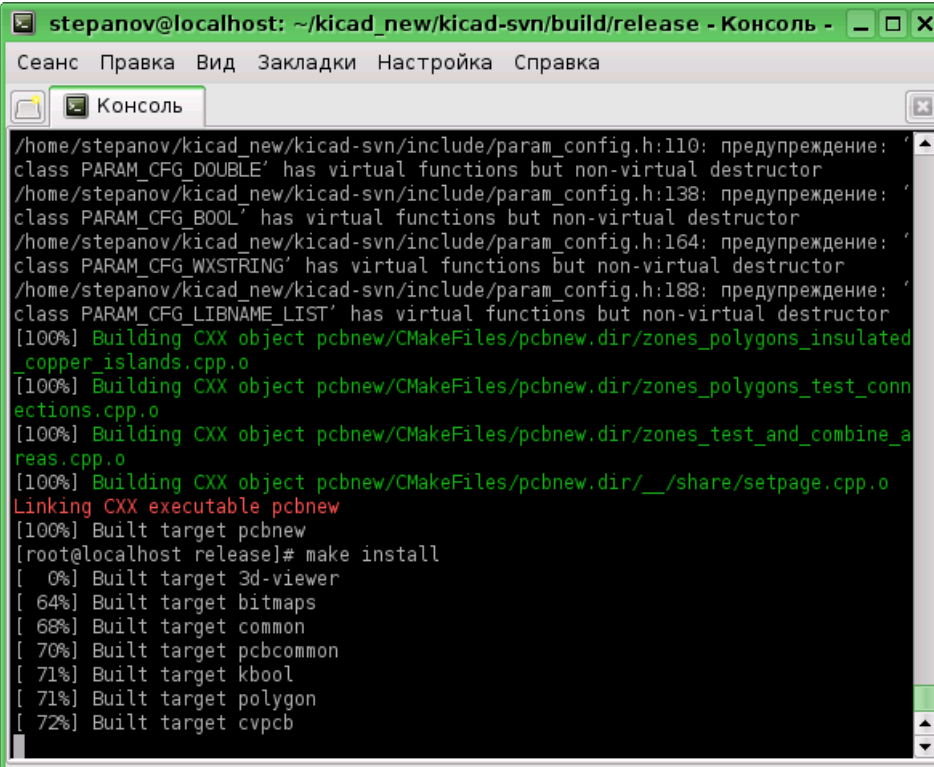
\$ make

```

stepanov@localhost: ~/kicad_new/kicad-svn/build/release - Консоль
Сеанс  Правка  Вид  Закладки  Настройка  Справка
Консоль
[ 32%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/Zo
om_Selected.cpp.o
[ 32%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/Zo
om_Select.cpp.o
[ 32%] Building CXX object bitmaps/CMakeFiles/bitmaps.dir/auto_renamed_to_cpp/zo
om.cpp.o
Linking CXX static library libbitmaps.a
[ 64%] Built target bitmaps
Scanning dependencies of target common
[ 64%] Building CXX object common/CMakeFiles/common.dir/about_kicad.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/base_screen.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/base_struct.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/basicframe.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/bezier_curves.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/block_commande.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/build_version.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/class_colors_design_sett
ings.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/class_marker_base.cpp.o
[ 64%] Building CXX object common/CMakeFiles/common.dir/class_plotter.cpp.o
[ 65%] Building CXX object common/CMakeFiles/common.dir/class_undoredo_container
.cpp.o
[ 65%] Building CXX object common/CMakeFiles/common.dir/common.cpp.o

```

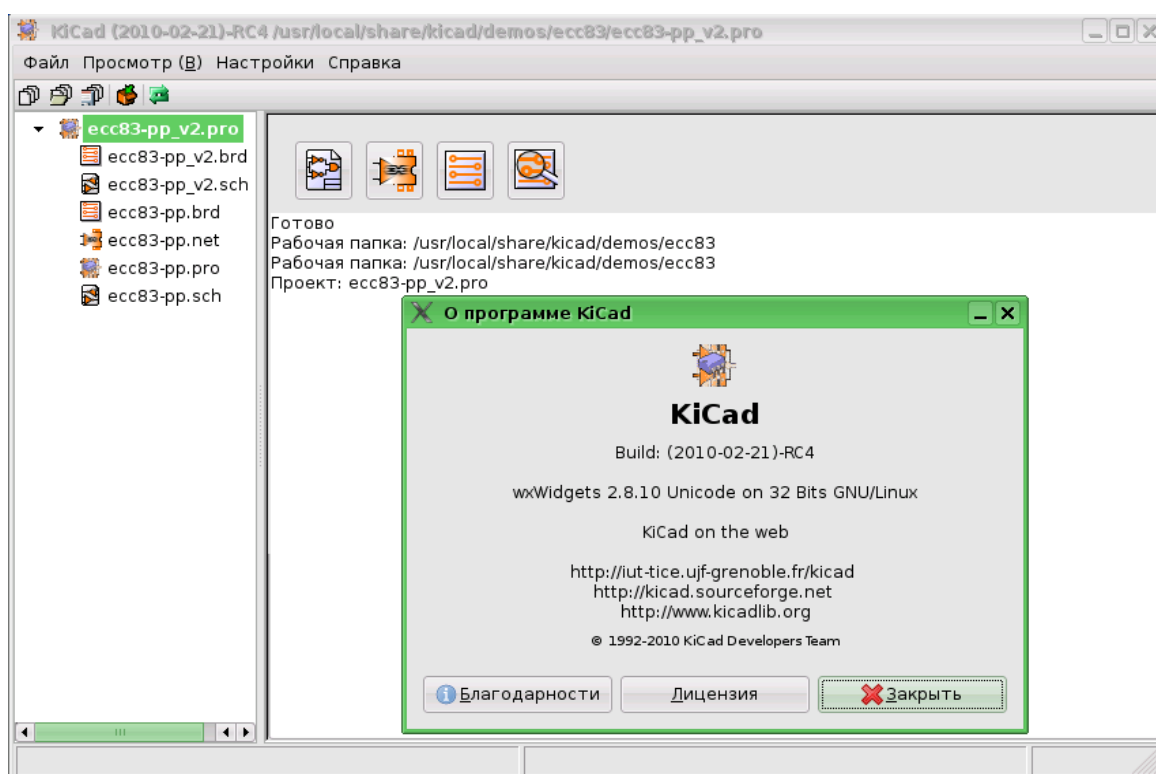
\$ make install



```
stepanov@localhost: ~/kicad_new/kicad-svn/build/release - Консоль
Сеанс  Правка  Вид  Закладки  Настройка  Справка
Консоль
/home/stepanov/kicad_new/kicad-svn/include/param_config.h:110: предупреждение:
class PARAM_CFG_DOUBLE' has virtual functions but non-virtual destructor
/home/stepanov/kicad_new/kicad-svn/include/param_config.h:138: предупреждение:
class PARAM_CFG_BOOL' has virtual functions but non-virtual destructor
/home/stepanov/kicad_new/kicad-svn/include/param_config.h:164: предупреждение:
class PARAM_CFG_WXSTRING' has virtual functions but non-virtual destructor
/home/stepanov/kicad_new/kicad-svn/include/param_config.h:188: предупреждение:
class PARAM_CFG_LIBNAME_LIST' has virtual functions but non-virtual destructor
[100%] Building CXX object pcbnew/CMakeFiles/pcbnew.dir/zones_polygons_insulated
copper_islands.cpp.o
[100%] Building CXX object pcbnew/CMakeFiles/pcbnew.dir/zones_polygons_test_conn
ections.cpp.o
[100%] Building CXX object pcbnew/CMakeFiles/pcbnew.dir/zones_test_and_combine_a
reas.cpp.o
[100%] Building CXX object pcbnew/CMakeFiles/pcbnew.dir/__/share/setpage.cpp.o
Linking CXX executable pcbnew
[100%] Built target pcbnew
[root@localhost release]# make install
[  0%] Built target 3d-viewer
[ 64%] Built target bitmaps
[ 68%] Built target common
[ 70%] Built target pcbcommon
[ 71%] Built target kbool
[ 71%] Built target polygon
[ 72%] Built target cvpcb
```

Сборка комплекса программ KiCAD для операционной системы ALT Linux выполняется за время от 20 минут до часа в зависимости от мощности компьютера.

3.3 Главное окно менеджера KiCAD для ALT Linux



4 Редактор переводов для русификации интерфейса KiCAD

При разработке открытого программного обеспечения сообщения оригинального интерфейса (в случае KiCAD - на английском языке), как правило, оформляются таким образом, что с помощью специальных программ-редакторов переводов, возможно дальнейшее их извлечение из исходного кода программы и подстановка для вместо них предложений любого другого языка. Ниже показано окно кросс-платформенного редактора Poedit, использованного для трансляции интерфейса KiCAD. Файл `/share/internat/ru/kicad.po` с помощью Poedit транслируется в бинарный файл `/share/internat/ru/kicad.mo`, используемый для подстановки русского интерфейса в систему KiCAD.

В интерфейсе KiCAD на сегодня около трех тысяч сообщений. При каждом изменении оригинального интерфейса процедуры извлечения, перевода и трансляции новых сообщений повторяются.

