SandBoxShield con kicad

BricoLabs

20 de xunio de 2015

Qué imos facer? Qué se describe en este documento?

Imos deseñar un shield para Arduino. Usando KiCad

Daremos unha visión xeral da funcionalidade da suite KiCad pero non vamos a redactar un tutorial detallado nin de deseño de circuitos nin do mesmo KiCad. Daremos as pistas xustiñas para empezar a traballar con KiCad, se queredes un tutorial detallado, en youtube tedes un super recomendable, elaborado por TutoElectro

Tamén comentaremos de pasada como fumos desenvolvendo iste proxecto.

Qué é KiCad?

KiCad é unha suite de deseño electrónico automatizado [1]. KiCad permite o deseño tanto de esquemas de circuitos como das placas de circuito impreso a nivel profesional. Hai versións de KiCad dispoñibles para Windows, Linux, Apple OS X. A suite está dispoñible para Windows, Linux e Apple OS X. É un programa gratuito e **libre** distribuido baixo licencia **GNU GPL v2**.

Mellor ainda, a suite KiCad é a elexida polo CERN para o desenvolvemento e deseño de electrónica. É de esperar que con este respaldo a suite mellore ainda mais.

Requisitos do shield

Queremos facer un shield sinxelo para a enseñanza.

[1] EDA suite en inglés

- 1 x LDR
- 3-4 x Botóns
- 1 x RGB
- 1 x LDR
- 1 x Pines servo
- 1 x sensor temp
- 2 x potenciómetros
- 1 x LED bermello
- 1 x LED amarelo
- 1 x LED verde
- 1 x Zoador

Instalación de KiCad (en Ubuntu)

Para instalar o KiCad en Ubuntu basta con facer o típico ciclo de instalación:

```
sudo apt-get install kicad
```

Se queremos estar á última temos o ppa de Monsieur Reynaud dispoñible:

```
sudo apt-add-repository ppa:js-reynaud/ppa-KiCad
sudo apt-get update
sudo apt-get install kicad
```

Nos escollimos esta opción.

Se non usades un linux baseado en Debian, teredes que consultar na rede como facer a instalación para o voso sistema operativo. De todolos xeitos a instalación é moi doada, donde podemos atopar algún problemiña é na instalación das bibliotecas de compoñentes que vos contaremos cos mais detalle mais adiante.

Configuración de directorios para este proxecto

Además de desenvolver o proxecto con KiCad queremos ter o proxecto dispoñible en github.

Agora que temos KiCad instalado imos preparar un directorio de traballo ao que chamamos **sandboxShield**.

O directorio **sandboxShield** será o "repositorio" ou depósito do noso proxecto para git. Contén os seguintes subdirectorios:

doc

: Contén a documentación do proxecto (o que estás a leer agora mesmo) redactada en Pandoc

kicad

: Contén o proxecto KiCad

Unha vez que temos preparado o directorio do proxecto activamos git para iniciar o control de versións.

Describir a configuración de git??

Biblioteca de compoñentes incluindo un shield para Arduino

As bibliotecas de KiCad están organizadas en duas partes:

- Un ficheiro que contén os símbolos dos compoñentes para usarse no editor de esquemas electrónicos **Eescheme**
- As pegadas dos compoñentes electrónicos, é dicir, a forma que ten que ter a pista da placa de circuito impreso (*PCB*) para poder soldar o compoñente.

O KiCad non trae por defecto unha biblioteca de compoñentes que inclúa shields de Arduino. Pero non hai problema hai bibliotecas que podemos descarregar da rede.

Unha biblioteca moi completa é a de Freetronics que podemos atopar tamén en github en:

https://github.com/freetronics/freetronics_KiCad_library.git

As bibliotecas de KiCad poden estar almacenadas en diferentes directorios do noso ordenador. Poderíamos engadir as bibliotecas que usemos en algún subdirectorio de */usr/share/kicad* ou de */usr/local/share*. Esta podería ser unha boa estratexia nun servidor compartido por varios usuarios. Tamén poderíamos descarregar todas as bibliotecas a un directorio común do noso *home*. Pero como estamos facendo un control de versións do noso proxecto con git a propia páxina da biblioteca suxírenos o xeito mais adoitado de facer a instalación: coma un submódulo git do noso proxecto.

Describir as vantaxes de usar un git submodule

Engadir a biblioteca como un submódulo de git

Dende o directorio principal de noso proxecto descarregamos a biblioteca de Freetronics coma un submodulo do noso proxecto:

```
git submodule add https://github.com/freetronics/freetronics_kicad_library.git kicad/ftlib
```

Despois de engadir a biblioteca como un submódulo se consultamos o estado git do noso proxecto aparecerán dous novos ficheiros:

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitmodules
    new file:   kicad/ftlibrary
```

Git engadíu automáticamente os dous novos ficheiros, o directorio que contén a nosa biblioteca eo ficheiro **.gitmodule** que levará o control de todolos submódulos que usemos.

En realidade os ficheiros que compoñen a biblioteca non pertencen ao noso depósito de software, git só leva conta da versión da biblioteca de Freetronics que estamos a usar.

Se queres saber mais de esta potente funcionalidade de git podes leer: http://www.git-scm.com/book/en/v2/Git-Tools-Submodules

Configurar *Eescheme* para usar a nova biblioteca

No menú Preferences -> Component Library sinalamos na ventá inferior o directorio do noso proxecto. Na ventá superior engadimos o ficheiro da biblioteca.

No github da biblioteca nos aconsellan poñer a biblioteca de primeira na nosa lista por que definen todo tipo de compoñentes. Nos de momento seguimos o consello de Freetronics, e a puxemos de primeira.

Configurar acceso aos datos de pegadas (*footprints*) en pcbnew

Configuramos un ficheiro para o noso proxecto declarando os *footprint* extra que imos a utilizar.

```
(fp_lib_table
  (lib
   (name FT)
   (type KiCad)
   (uri ${KIPRJMOD}/ftlibrary/freetronics_footprints.pretty)
   (options "")
   (descr "Freetronics Kicad Library")
   )
)
```

Engadimos o novo ficheiro ao noso repositorio

```
git add fp-lib-table
```

Abrimos *Pcbnew* e no menú *Preferences->Footprint Libraries Manager* comprobamos que na pestaña *Project Specific Libraries* figura o noso ficheiro.

Outra biblioteca moi currada

git://smisioto.eu/KiCad_libs.git

Ainda mais bibliotecas

http://www.kicadlib.org/

Instalación das bibliotecas

http://www.arunet.co.uk/tkboyd/ele2pcbka.htm

Tutorial

A pantalla xeral

Pantalla xeral de KiCad opcións, citar a lista de hotkeys

Abrindo un proxecto

Abrimos un novo proxecto: File::New Project (Ctrl+N) **sandbox_shield**

---

falar das propiedades do documento

Crear e Editar o esquema do circuito

O primeiro que imos facer é o esquema do circuito. Para isto temos que usar a ferramenta *Eeschema* que podemos atopar en tres lugares diferentes [2] na barra de iconos de ferramentas, no menú de KiCad no título da fiestra, ou có atallo **Ctrl+E**.

Abrimos eescheme e creamos un novo ficheiro de esquema.

Checklist

- Crear o esquema do circuito (usando Eescheme)
- Chequeo de erros (opción *Perform Electrical Rules Check*)
- Xerar o ficheiro NET (opción *Generate netlist*)

Tips

- Falar dos flags
  - Power flags
  - Not used flag

[2] Isto de ter varios xeitos de facer unha cousa é habitual en KiCad como iremos vendo

Meta

Este documento está escrito en Markdown-Pandoc. Pandoc e un sistema moi sinxelo de documentación que permite xerar multiples formatos de saída.

As fontes do documento están no directorio **doc/src**. Os formatos de saída son este fichero **README.md** en formato Markdown-github e os documentos que podes atopar no directorio **doc/out** incluindo un pdf.

Os documentos xeneranse automáticamente a partir do ficheiro fonte sen mai que executar:

```
$ cd doc
$ ./makeDoc
```

É importante cambiar ao directorio doc antes de executar o **makeDoc**.