



PCB
Memorias de Laboratorio

Memorias del desarrollo de una PCB para
la asignatura Laboratorio de Desarrollo
de Hardware: Joystick Arduino

Rafael Torres Malpartida

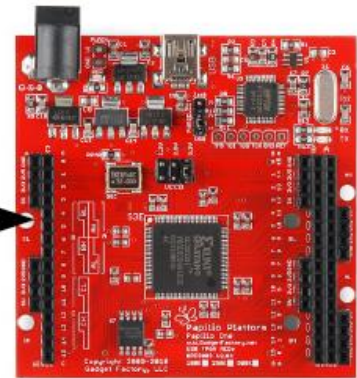
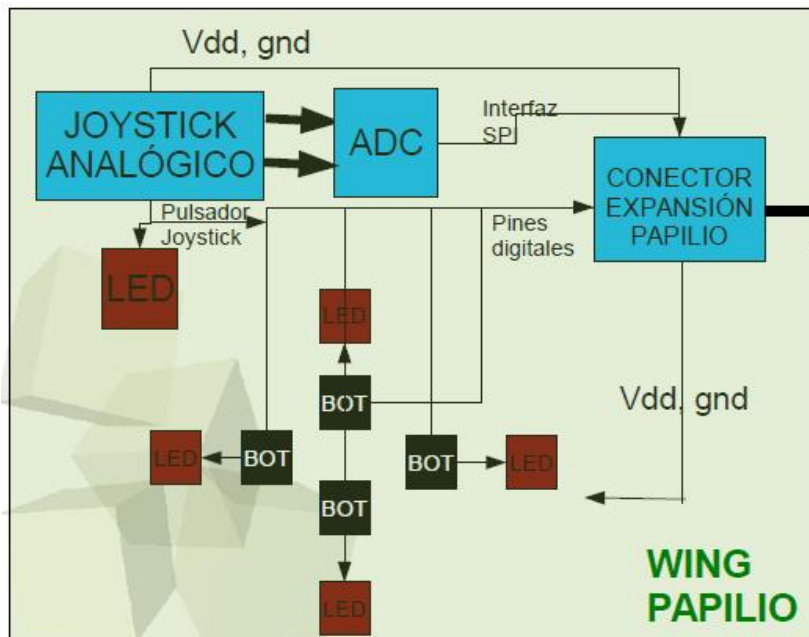
LDH

Contenido

Introducción	2
Partes del sistema	3
Recomendaciones	3
Instalación del entorno	4
Estructura de Carpetas tras la instalación.....	6
Creación de un proyecto en Eagle.....	6
Creación de un esquemático.....	7
Analog Thumb Joystick.....	7
Momentary Switch	7
Led genérico	8
Resistencias SMD M1206	8
Condensadores.....	8
FerriteBead.....	9
ADC1285102.....	9
Conectores para expansión	10
Shield de arduino	10
Notas*	11
Colocación de componentes en esquemático	12
Diseño para arduino únicamente.....	15
Enrutado a mano en board	16
Plano de masa	18
FINAL	22
Generar Gerbers.....	23
Fresado y corte.....	24
Ensamblaje de componentes	25
Testado de la placa	26

Introducción

La PCB que se pretende diseñar va a ser una placa de expansión dual para Arduino y ZPUino que funcione de manera similar a un mando de consola: con joystick y pulsadores.

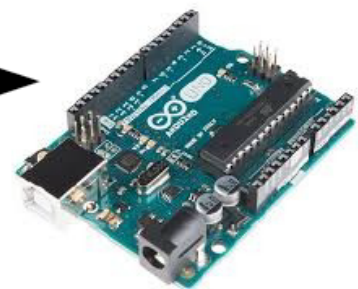
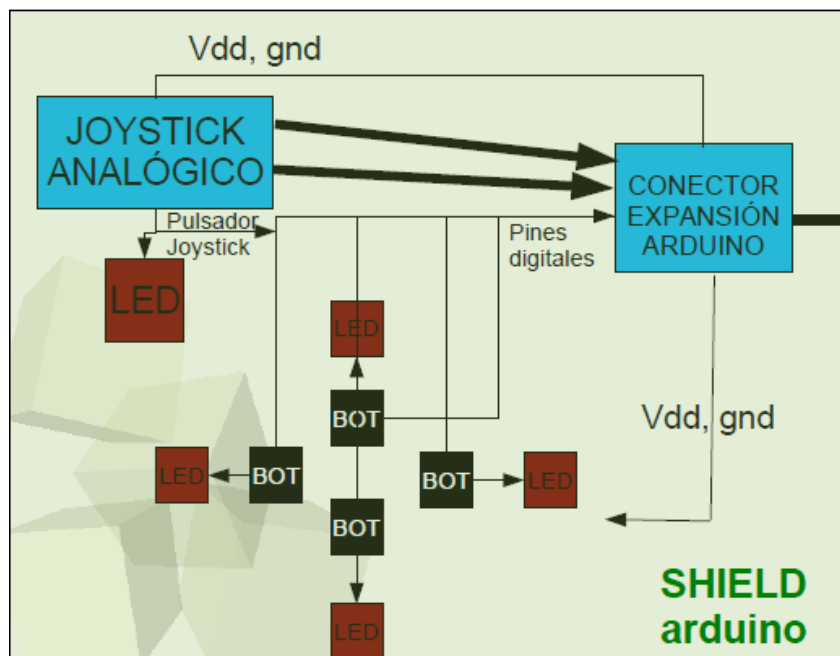


7

Debido a que no dispongo de todo el tiempo que me gustaría. He optado por realizar los 2 diseños:

1. Esquemático para Arduino/ZPUino
2. Esquemático para Arduino

El diseño que se realizará y fabricará finalmente será el más sencillo: Arduino.



8

Partes del sistema

- Joystick analógico: Genera señal vertical y horizontal analógica entre VDD (5v) y GND (0v). Dispone de pulsador central. Le conectaremos LED. El pulsador debe ir conectado a un pin digital de la placa.
- Pulsadores: UP, DOWN, LEFT, RIGHT. Tendrán resistencia de PULL-DOWN. Los conectaremos a LEDs. Irán conectados a pines digitales de la placa
- Conectores para los terminales de la Placa tipo Shield arduino. Se emplearán solo los conectores que realmente vayamos a utilizar. Lo decidiré durante el diseño a conveniencia.
- Leds
- Convertidor Analógico/Digital: ADC. Recibirá las señales analógicas del Joystick para digitalizarlas y enviarlas a la placa a través de interfaz SPI
- Conectores para los terminales de la Placa (tipo wing): WING C y WING B

Recomendaciones

El diseño de la board o layout final debe tener en cuentas las siguientes recomendaciones:

- Área máxima: 10cm x 6cm
- Es un diseño a una sola cara (solder side, bottom side)
- Tamaño del cable o wire a usar: 0,4064 mm (0,016inch).
- Crear plano de tierra

Una vez diseñados las dos placas habrá que analizar cuál de los diseños board se va a fabricar con la máquina ProtoMat S62. Una vez seleccionado habrá que generar los Gerber.

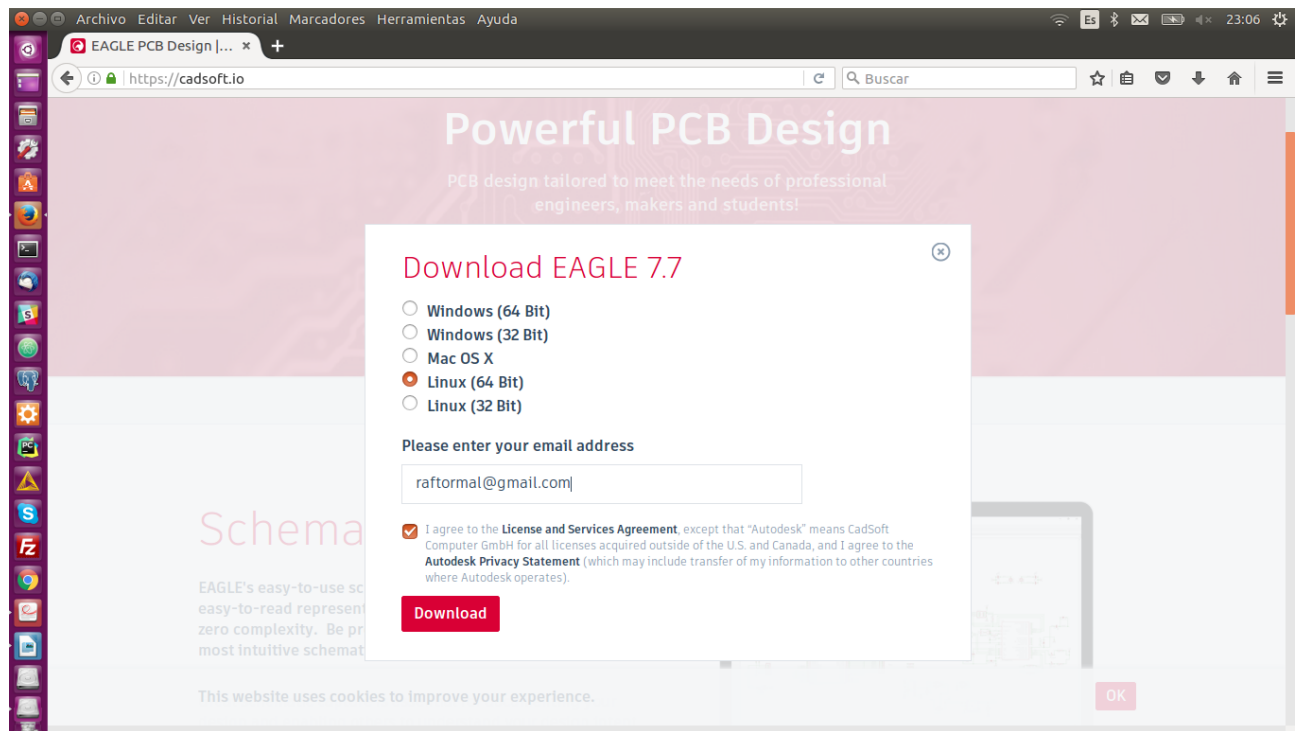
Instalación del entorno

Eagle puede descargarse desde:

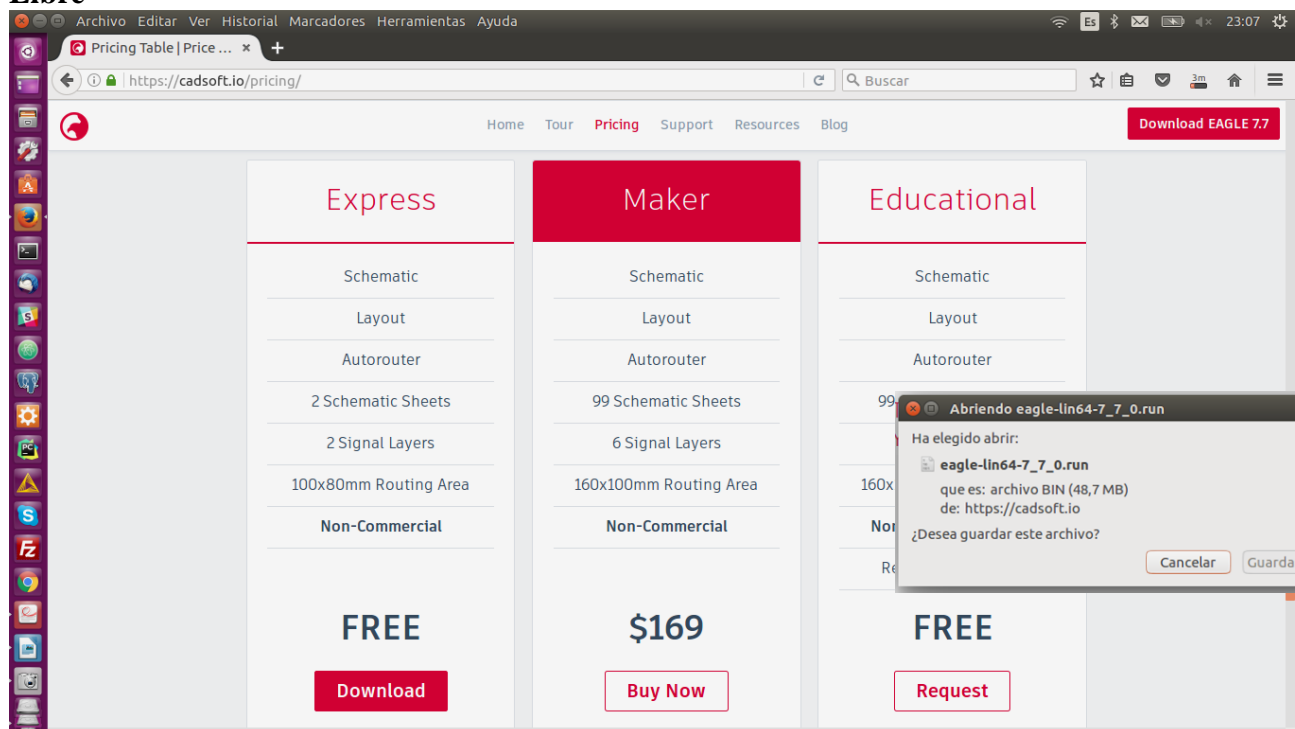
<https://cadsoft.io/>

Es posible instalarlo en diferentes plataformas, Windows, Linux, Mac.

Se puede ejecutar con una licencia restringida sin coste: **Eagle express**



Libre

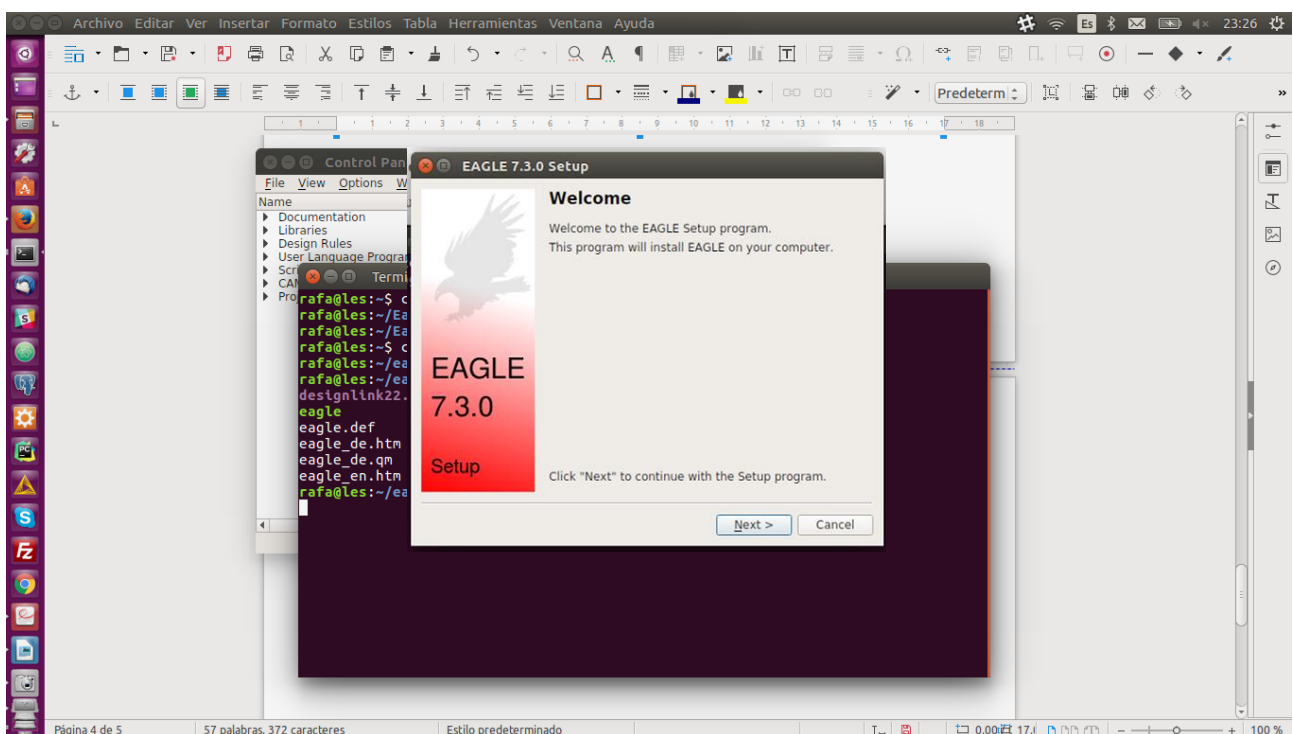


```
Terminal Archivo Editar Ver Buscar Terminal Ayuda
rafa@les:/media/rafa/dat/Downloads$ ./eagle-lin64-7_7_0.run
bash: ./eagle-lin64-7_7_0.run: Permiso denegado
rafa@les:/media/rafa/dat/Downloads$
```

```
chmod a+x eagle-lin64-7_7_0.run
./eagle-lin64-7_7_0.run
```

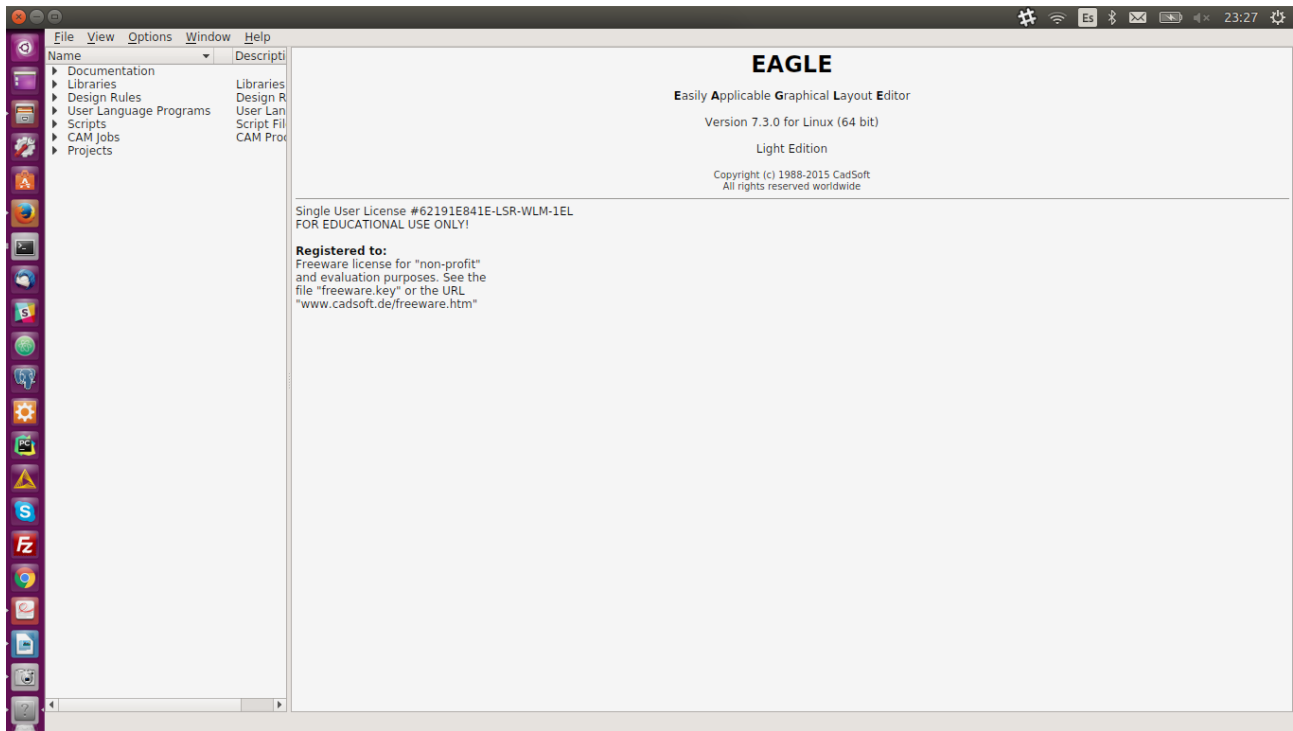
A pesar de esto, el problema persiste. Con la versión: eagle-lin64-7.3.0.run no tuve problemas

*Más tarde en el laboratorio pudimos hacer funcionar la versión 7.7 y continué el trabajo en esta nueva versión.



TRICK: Si queremos lanzar Eagle en otro proceso y dejar la consola libre podemos usar el parámetro & al final de la llamada:

```
./eagle &
```

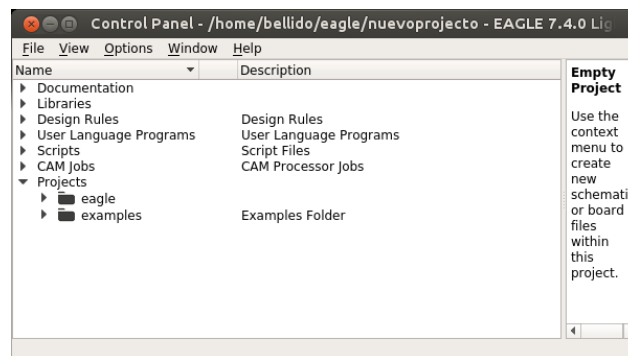


Estructura de Carpetas tras la instalación

- El ejecutable se encuentra en:
 - <eagle-X.X.X>/bin/eagle
 - <eagle-X.X.X>/lbr/ Carpeta de librería de componentes incluida con la herramienta
 - <eagle-X.X.X>/projects/ Carpeta para proyectos con eagle
 - <eagle-X.X.X>/dru/ Carpeta con ficheros de reglas de diseño
 - <eagle-X.X.X>/doc/ Carpeta con documentación. Incluye manuales y tutoriales en PDF
 - <eagle-X.X.X>/scr/ Carpeta para scripts de configuración de Eagle
 - <eagle-X.X.X>/ulp/ Carpeta para programas en lenguaje propio de eagle

Creación de un proyecto en Eagle

En panel de control seleccionar y expandir el path de projects y seleccionar carpeta donde crearemos el proyecto.



Seleccionada la carpeta, botón de la derecha del ratón y New project. Cambiar nombre y tendremos creado el proyecto, inicialmente vacío.

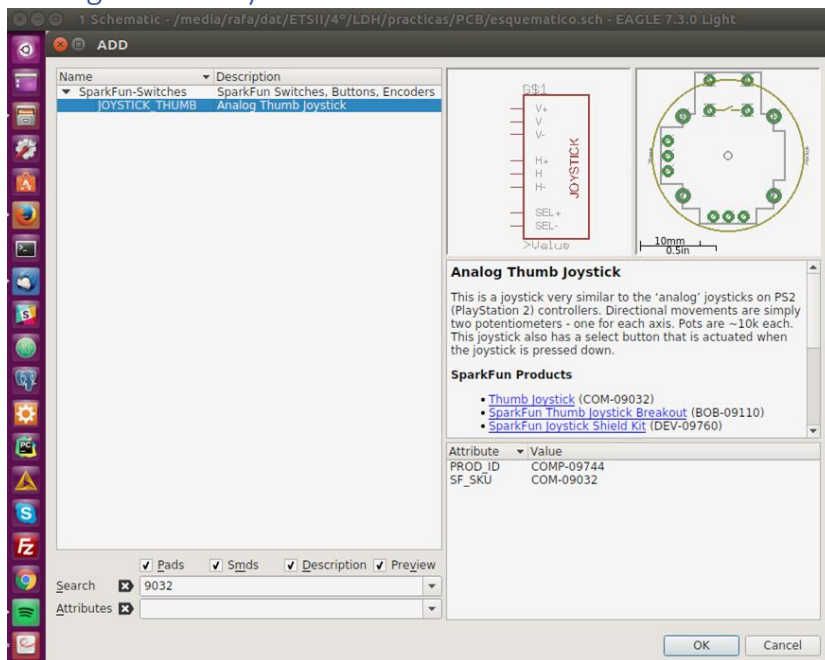
Para modificar la carpeta de proyectos: En la pestaña Options ==>Directories y modificar/añadir nueva carpeta.

Sobre el proyecto nuevo ya podremos crear un nuevo esquemático, layout o incluso una librería. En esta ocasión, nos bastará con obtener las huellas de librerías externas, muy importante su uso.

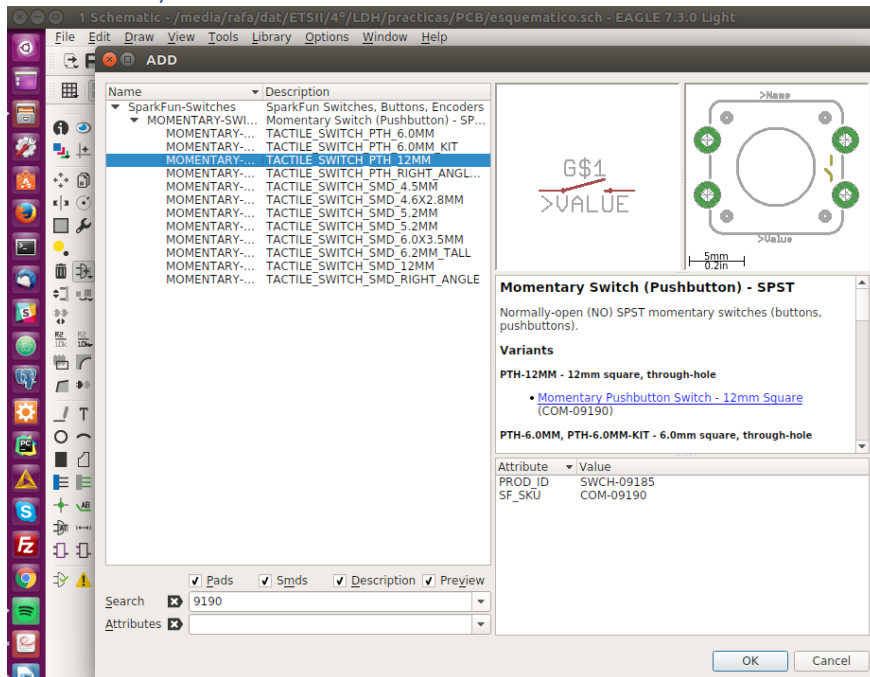
Creación de un esquemático

Librerías siempre debemos tenerla en el directorio raíz de librerías especificado en opciones. En este punto, debemos buscar los componentes uno a uno y en caso de ser necesario descargar otras. Se trata de un trabajo laborioso para el que hay que tener mucho cuidado en no confundir las huellas. Los vamos colocando uno a uno en el esquemático. En principio no voy a preocuparme del orden y la cantidad ya que la herramienta permite luego toda clase de opciones de edición. Podemos ir comprobando en el board que la huella es correcta si no estamos seguros de haber elegido el componente adecuado.

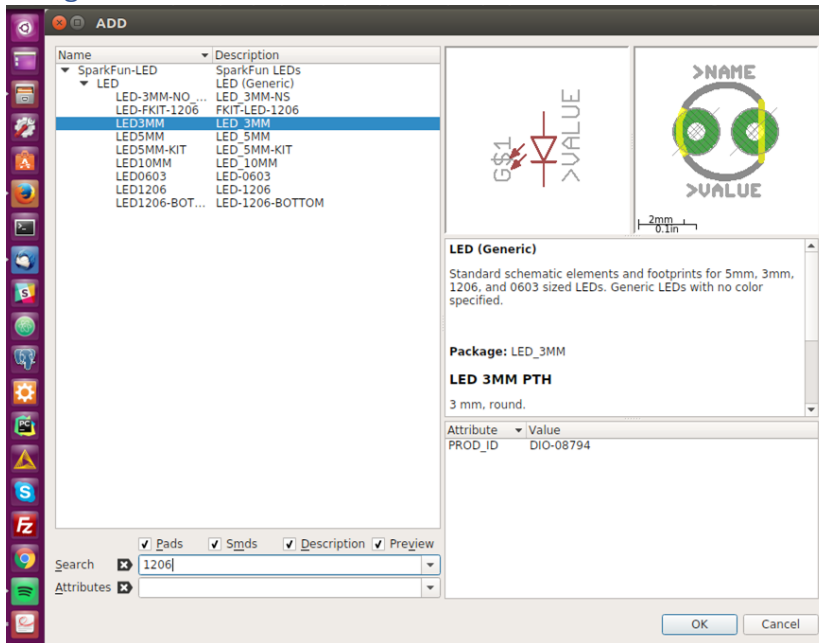
Analog Thumb Joystick



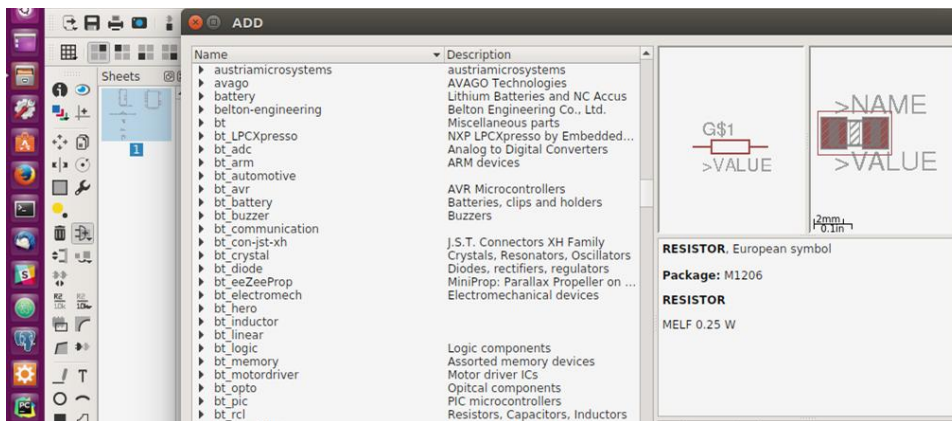
Momentary Switch



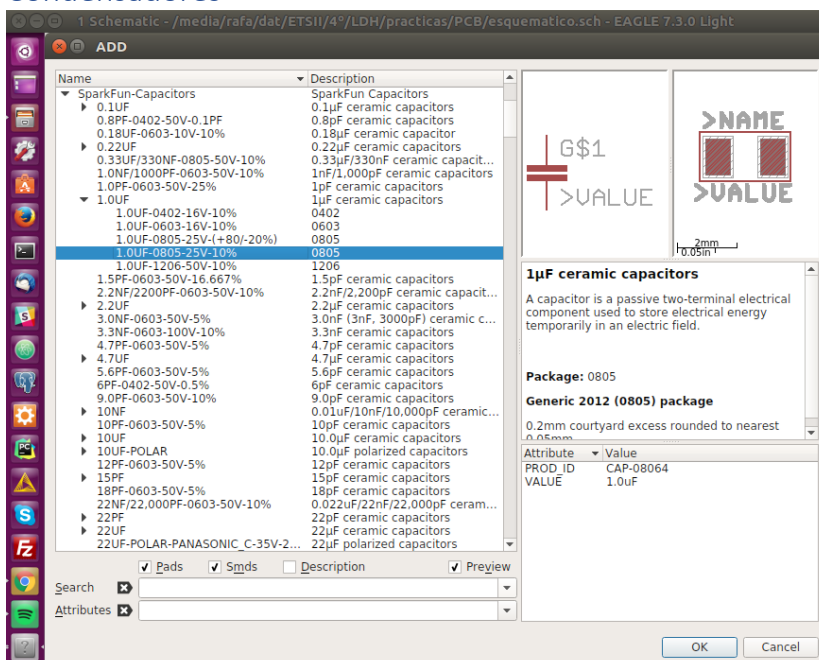
Led genérico



Resistencias SMD M1206

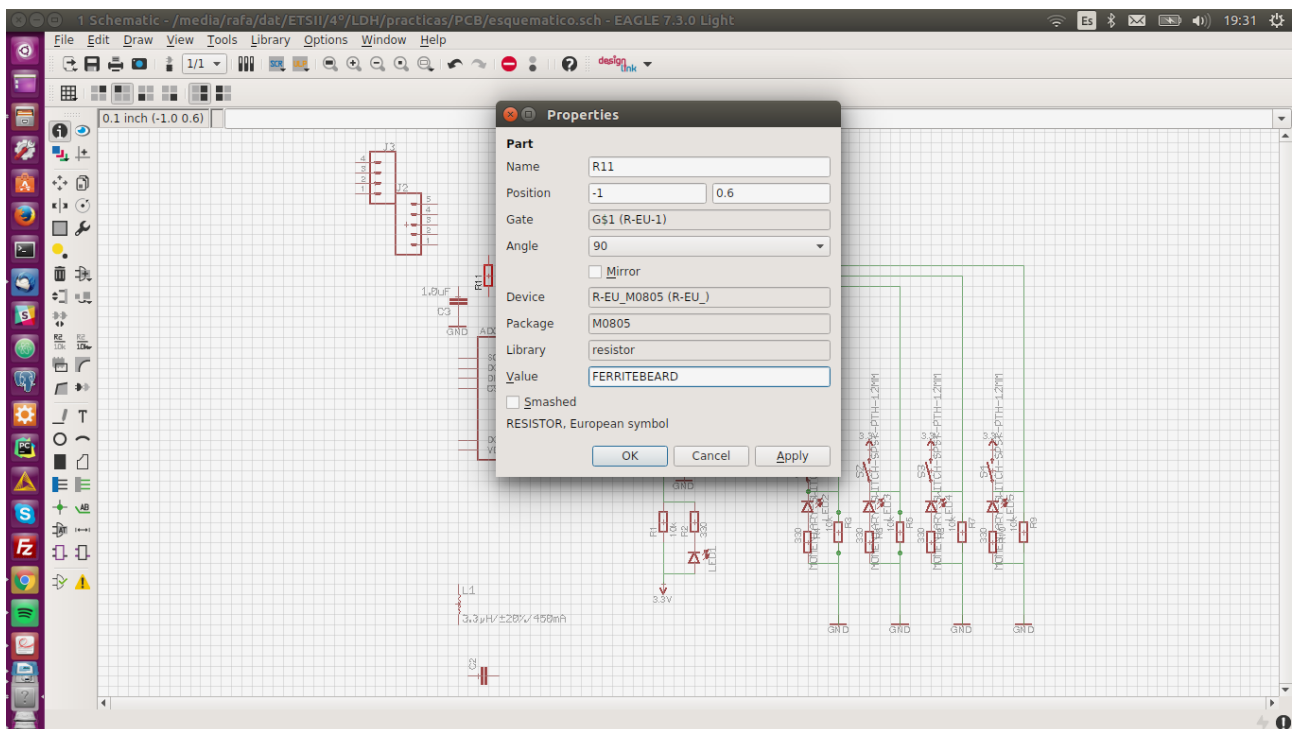


Condensadores

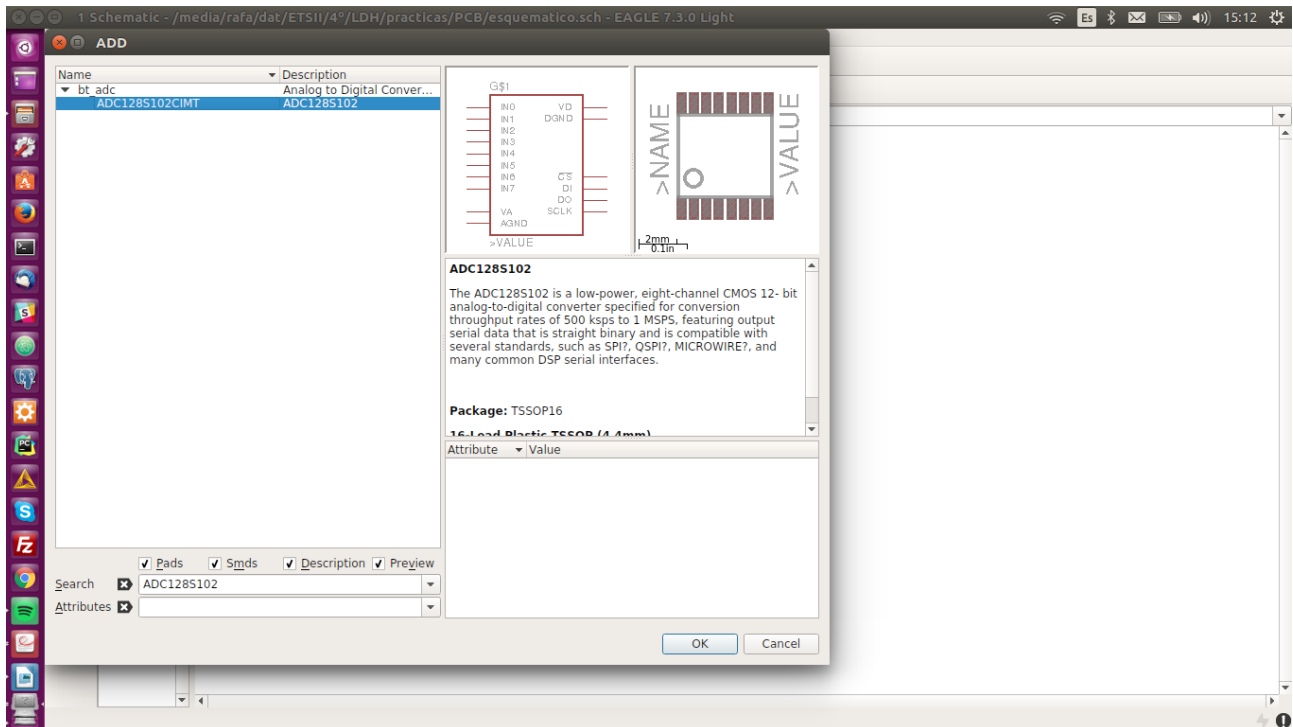


FerriteBead

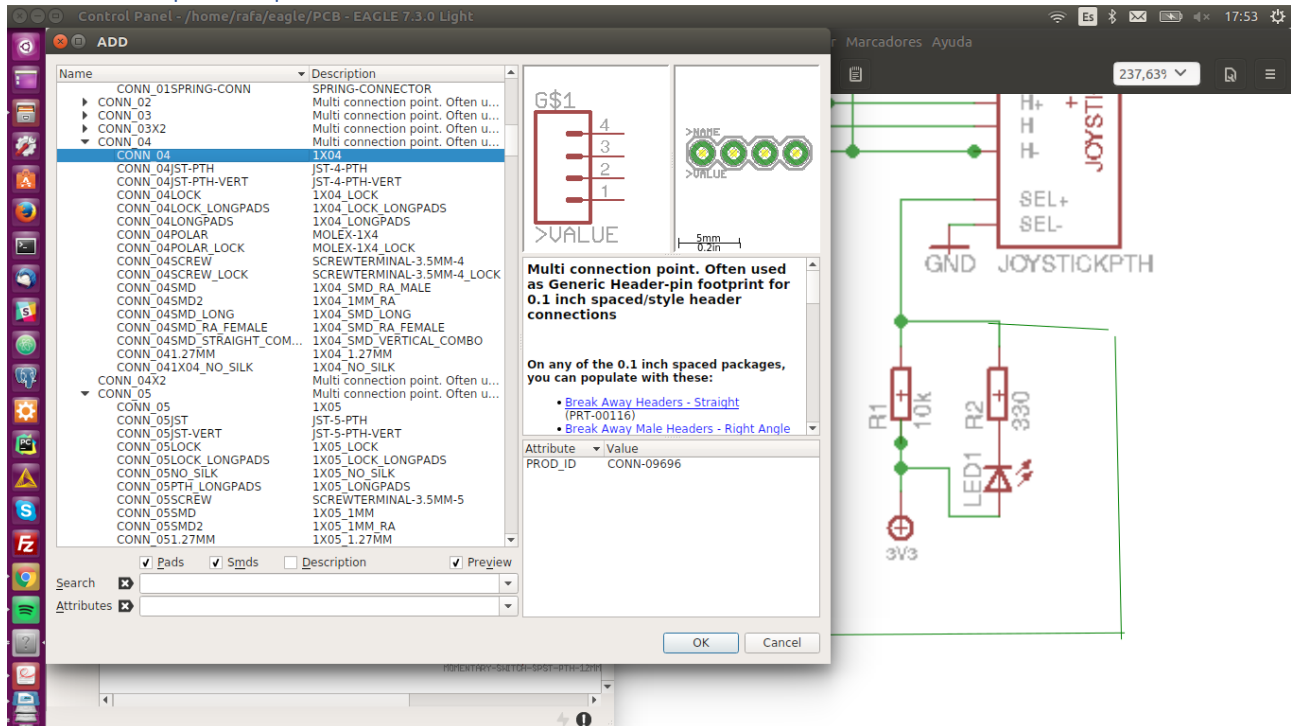
Usando una Resistencia SMD 0805 que tiene misma huella, modificando nombre y value



ADC128S102

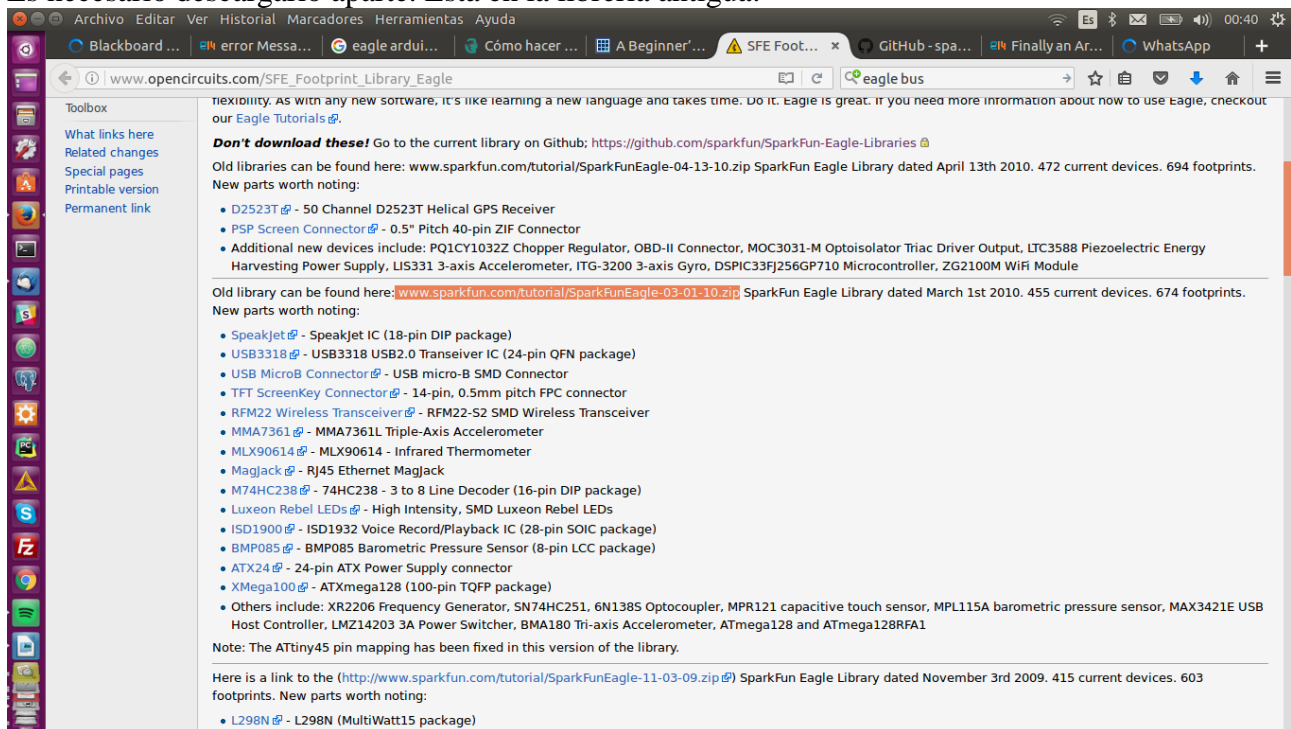


Conectores para expansión

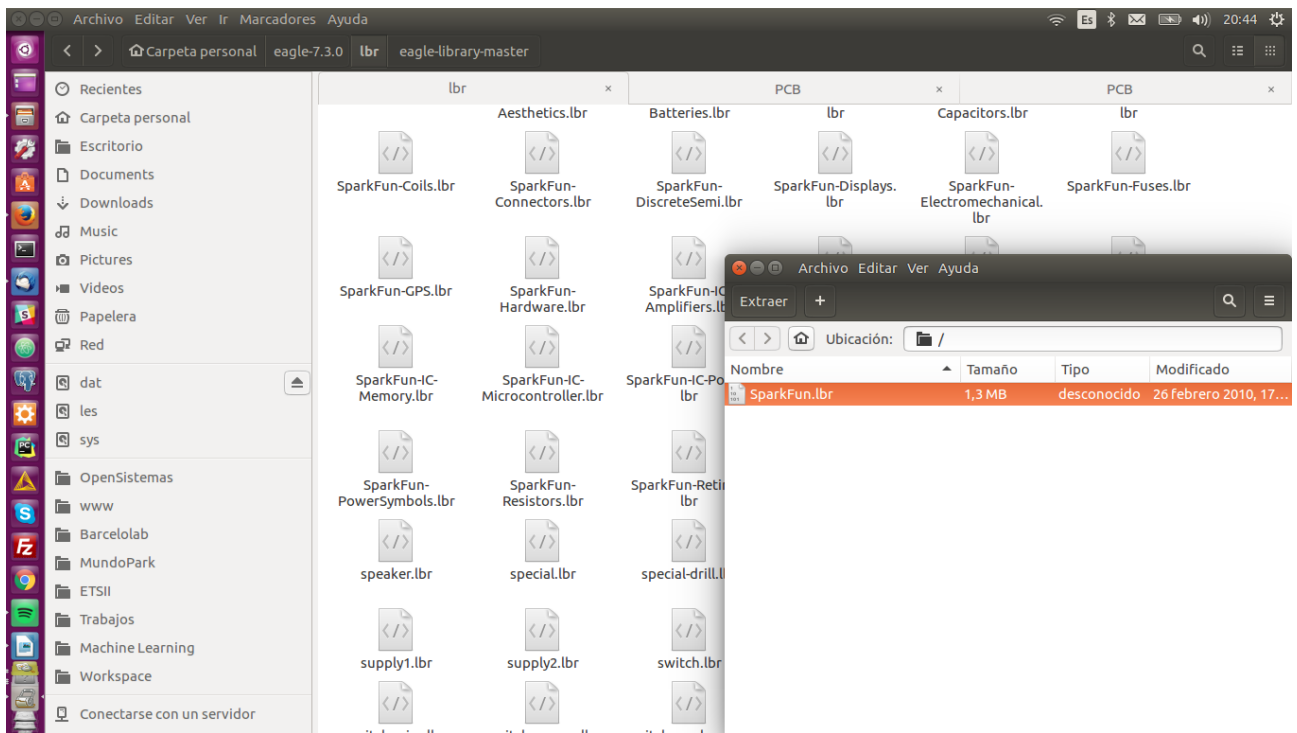


Shield de arduino

Es necesario descargarlo aparte. Está en la librería antigua:



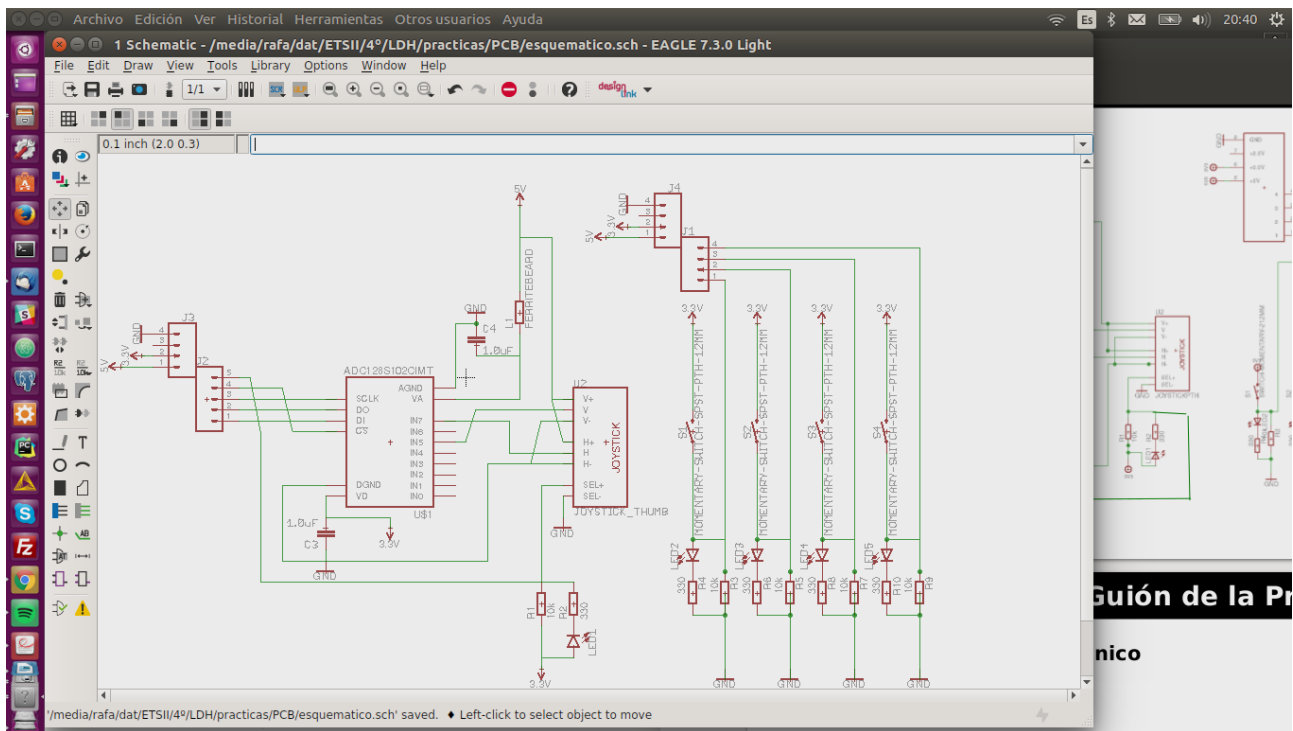
Mismo procedimiento cuando anteriormente añadimos la librería completa de SparkFun obtenida de GitHub



Buscar las librerías lleva bastante tiempo si no sabes utilizar bien el buscador. Es importante, en el panel de control, botón derecho en tu proyecto y **use all** de esta forma estaremos seguro de que todas las librerías estén presentes. Aun así, muchas veces el buscador sigue sin ayudarte demasiado y en ocasiones me ha sido de más ayuda buscarlo gracias a las descripciones de la columna de la derecha de cada librería en **Panel de control**.

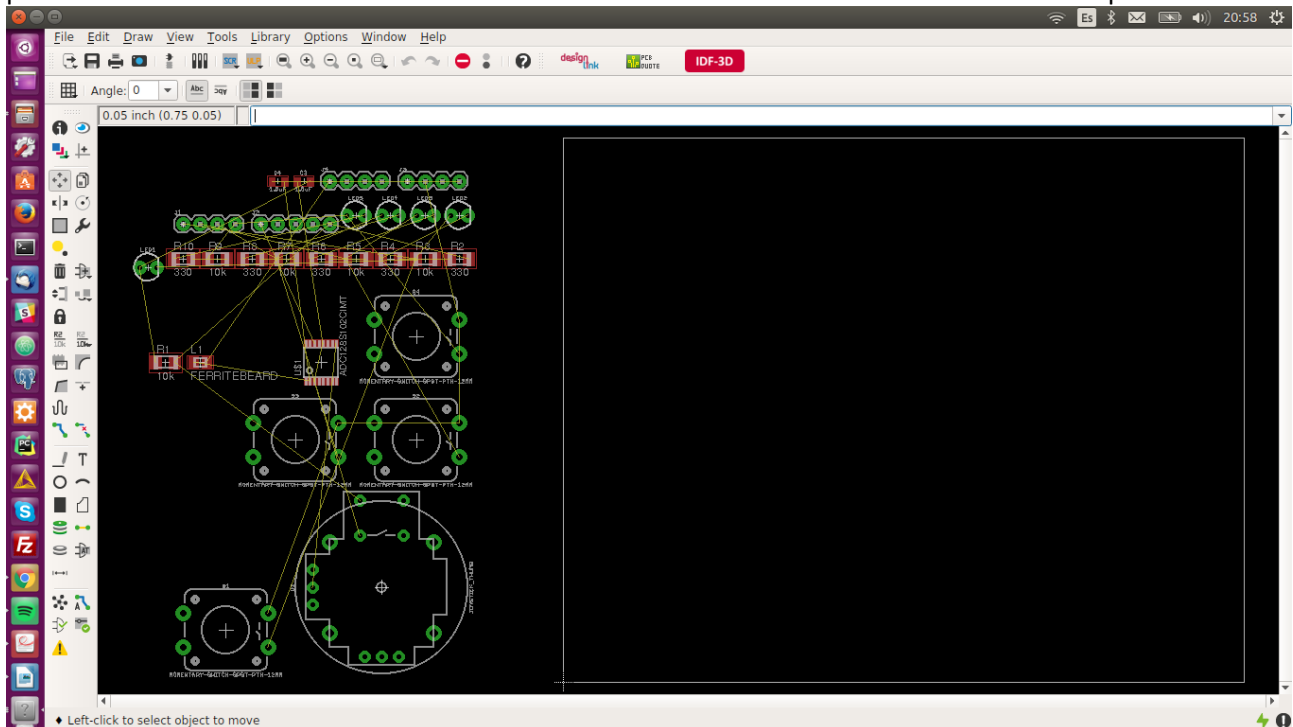
Colocación de componentes en esquemático

Este momento, usando las herramientas de copiar, cortar, pegar colocamos tantos componentes como necesitemos y damos a cada uno un nombre y valor (si necesario).

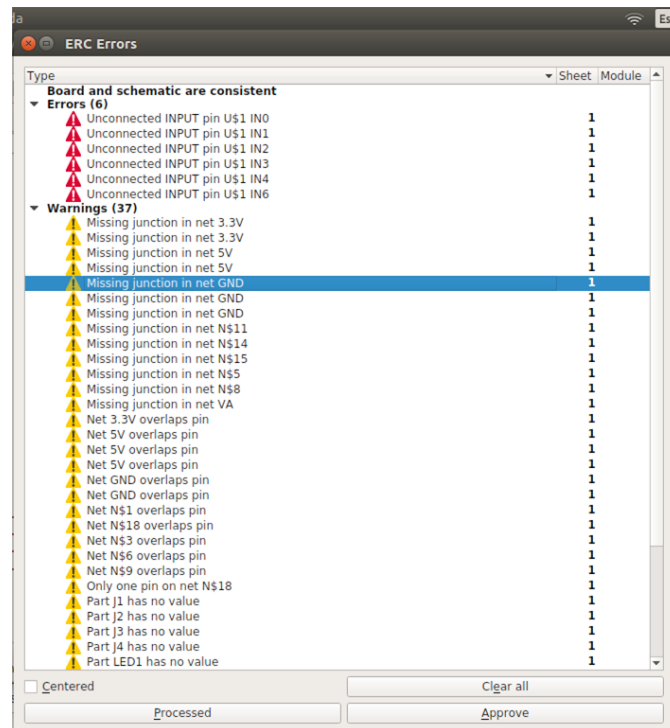


Posteriormente hacemos las conexiones con la herramienta wire.

Si ahora vemos el board file, podemos comprobar como ya tenemos nuestros componentes preparados para ser distribuidos cuando creamos convenientes. Pero antes debemos hacer muchas comprobaciones...



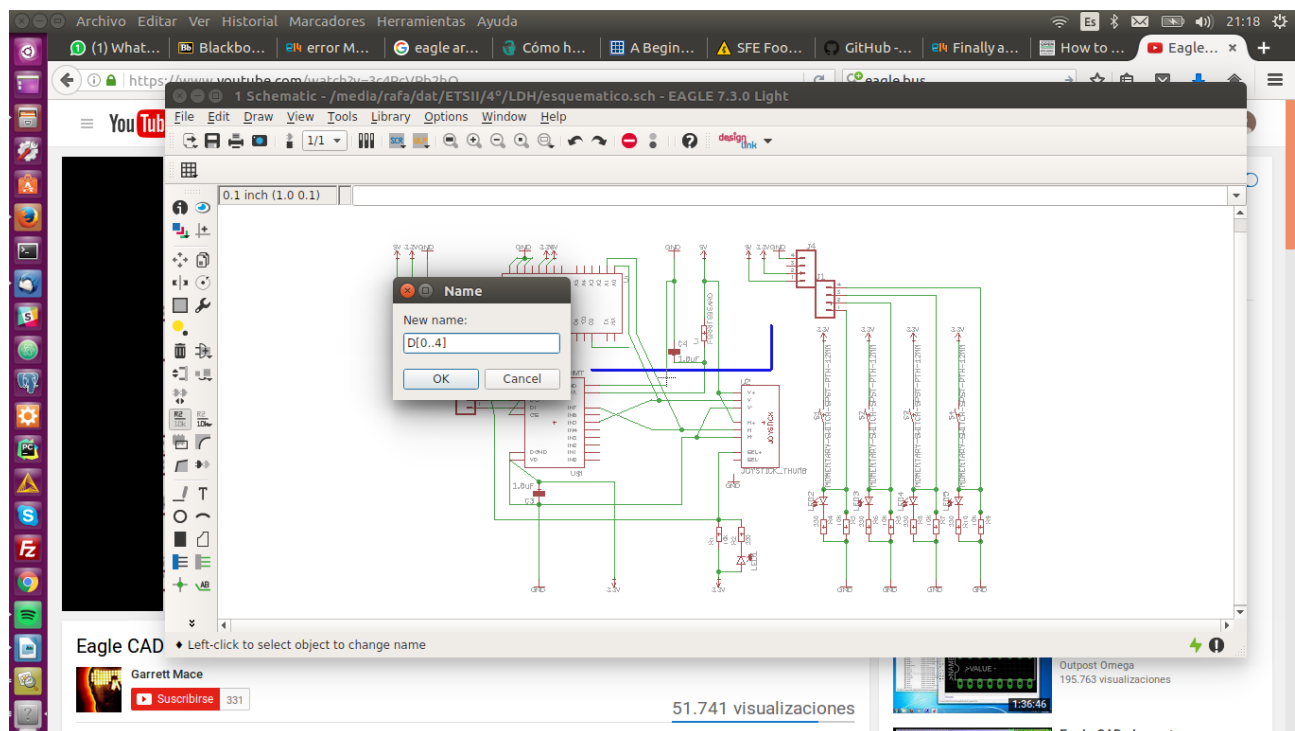
Terminamos primero el esquemático, comprobamos errores frecuentemente



Solucionamos los warnings missing junctions fácilmente con la herramienta **junction** que nos ayuda a hacer conexiones de más de 2 cables.

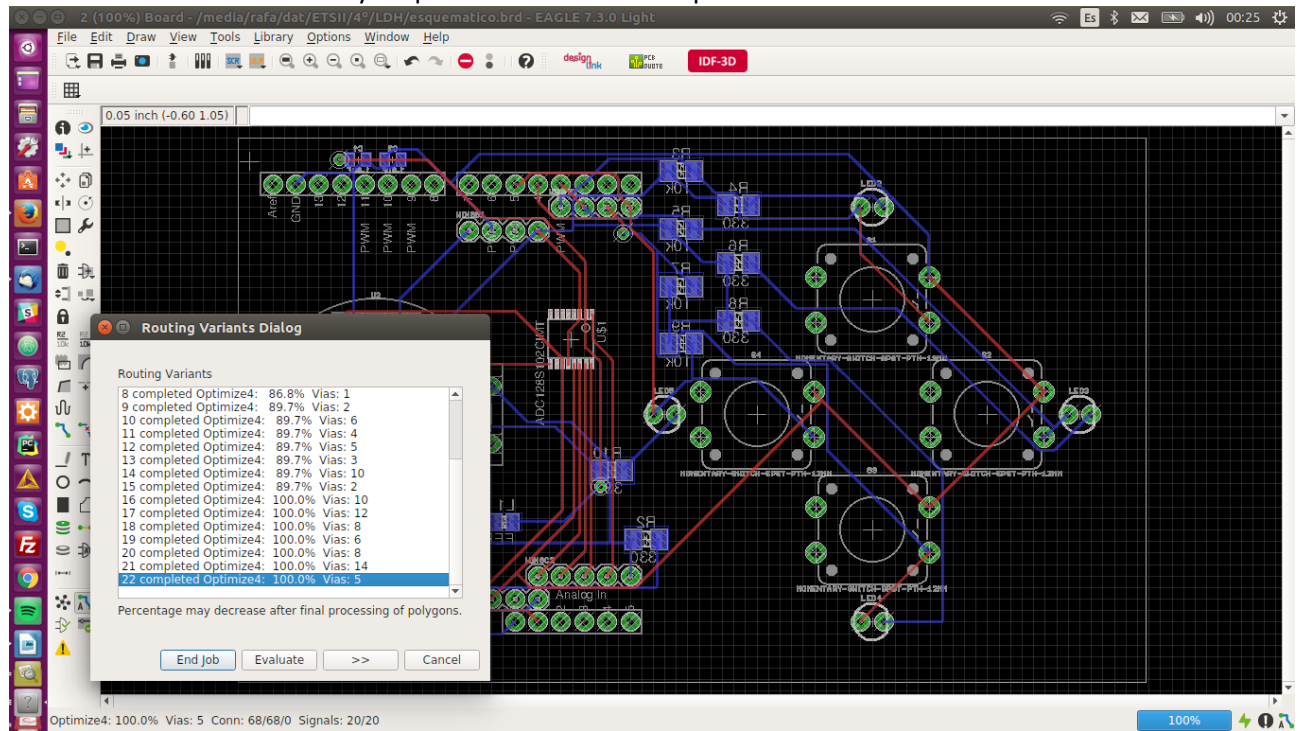
Podemos ver como existen muchas conexiones que se han hecho mal, de ahí la importancia del chequeo. Cada vez hay más wires... vamos a hacer uso de la herramienta de BUS. Es muy sencillo, lo importante es que al crear el bus coloquemos el nombre correcto para el bus deseado:

BUS NAME: D[2..5] ...



Haciendo ahora uso de la herramienta NET podemos elegir que línea usar, nos mostrará un desplegable con D2, D3, D4, D5 al clicar en el bus para conectarlo.

Ahora volvemos al board file y empezamos a colocar componentes



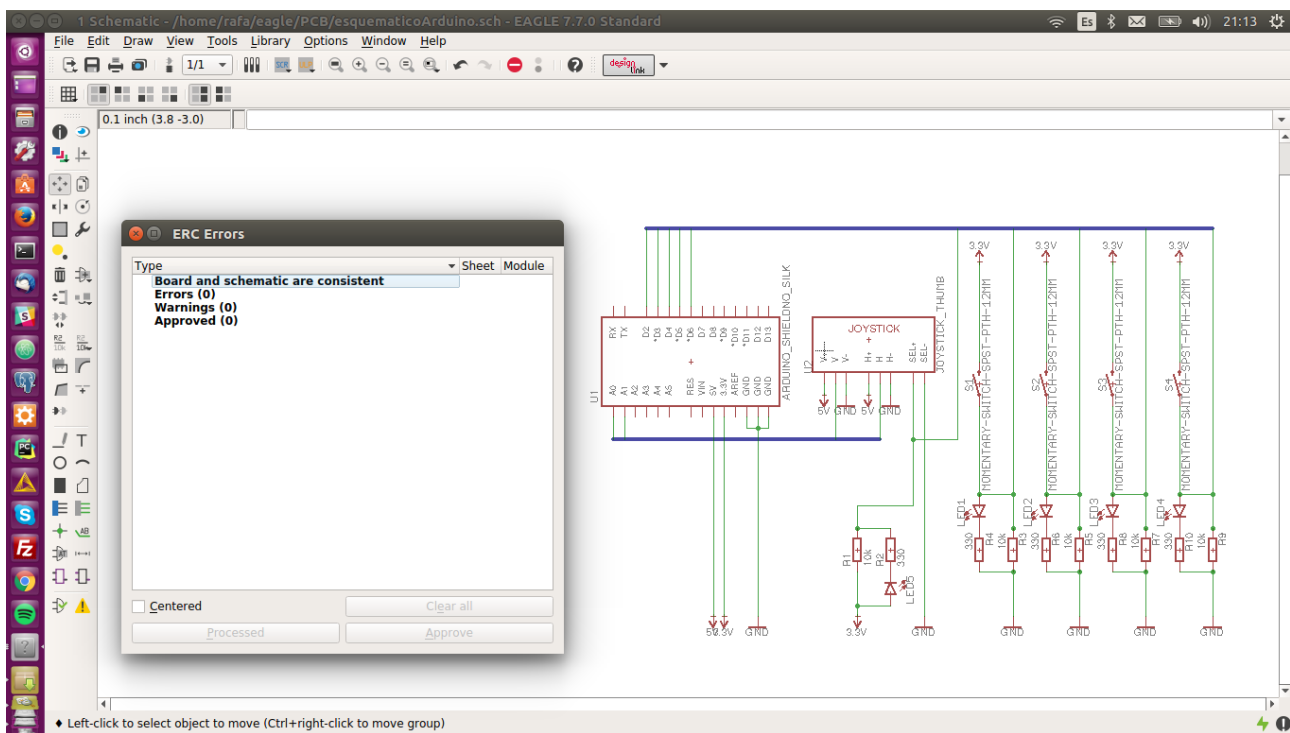
El autorouter ha funcionado bastante bien ordenando los componentes

Problemas:

Falta de tiempo

2 capas → solo 1!

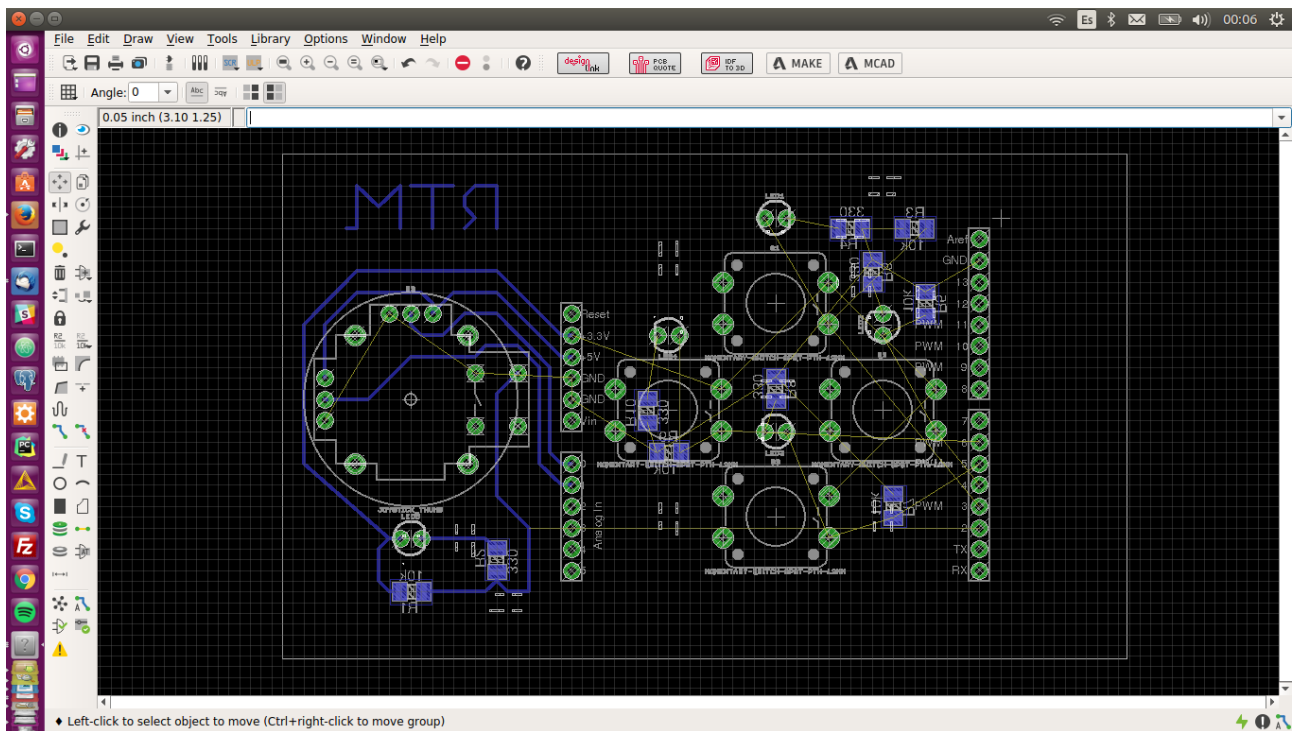
Diseño para arduino únicamente



2 buses: 2 y 5 líneas para datos analógicos y digitales respectivamente

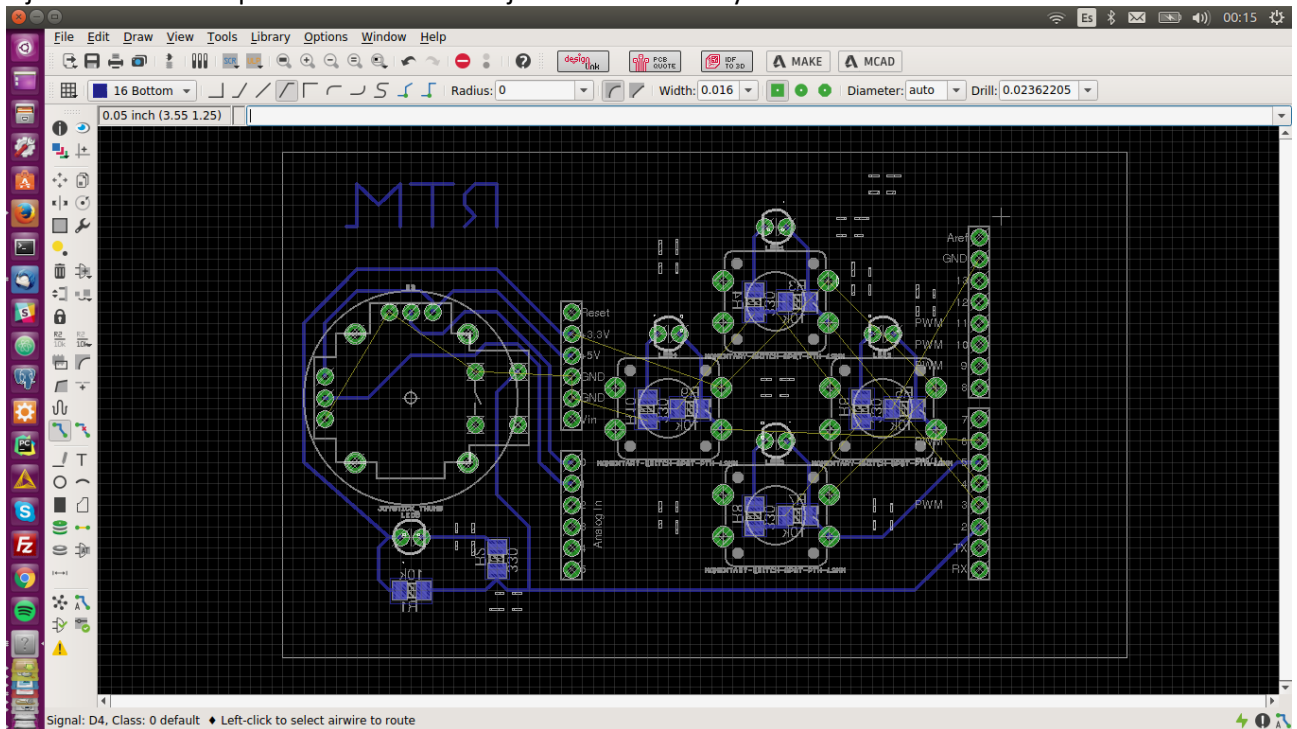
Para terminar de limpiar los Warnigs de LED voy a poner un espacio en su value y así no tendré más missing values.

Enrutado a mano en board

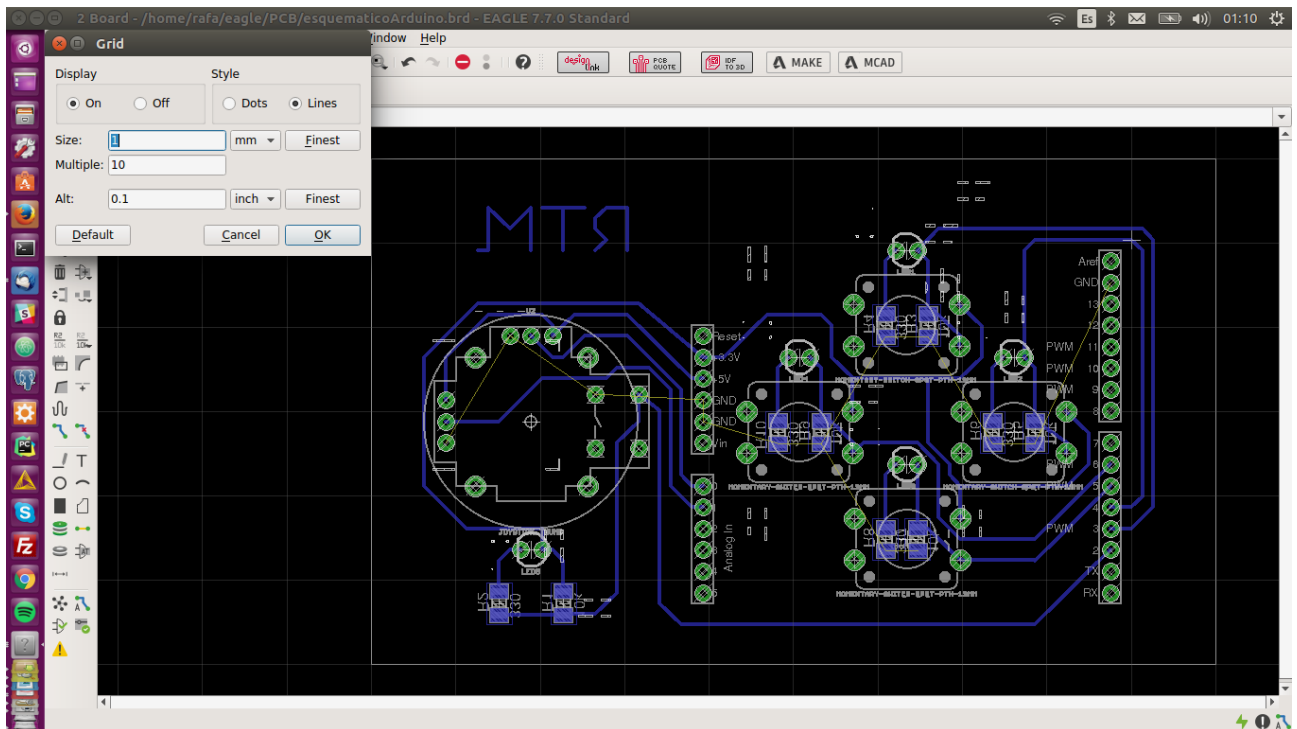
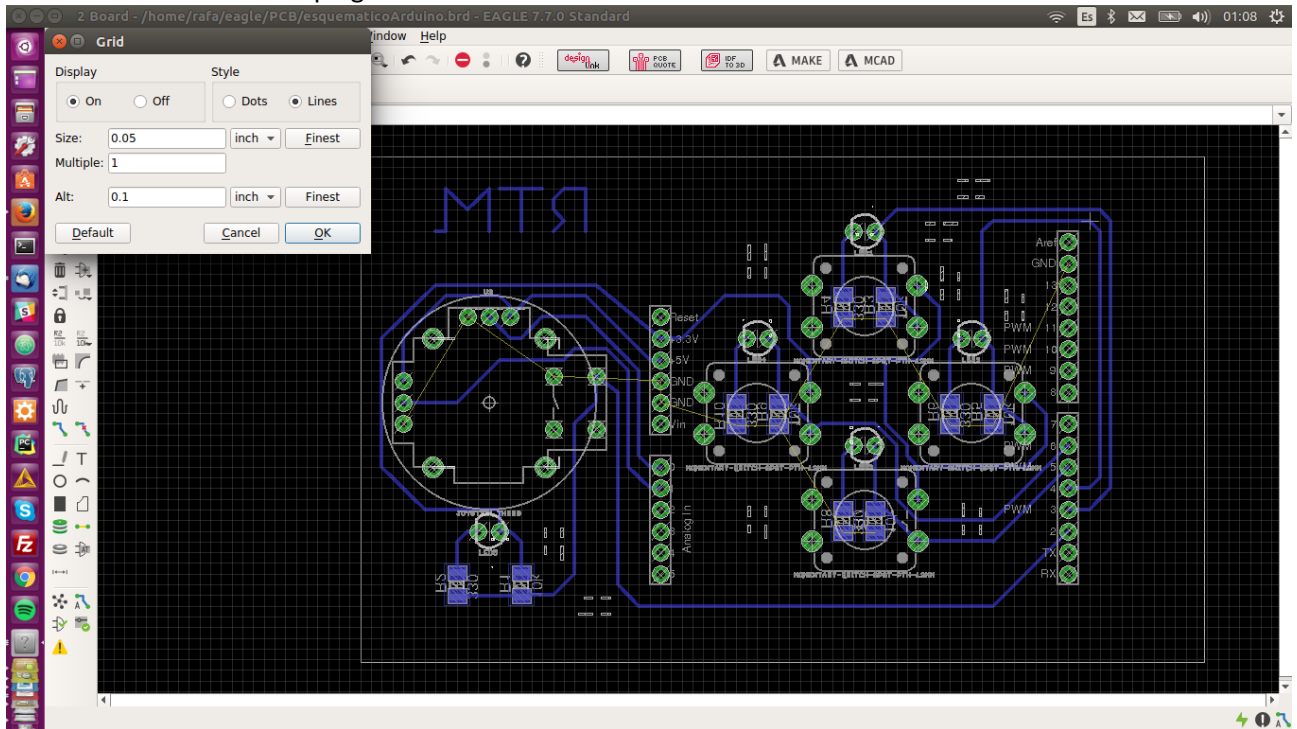


Debemos ir trazando las pistas, dejando las separaciones mencionadas, comienza una especie de tetris (al menos en mi caso) más bien juego de la serpiente. Para que cada conexión llegue a donde debe sin cruzarse, sin crearme demasiados problemas para etapas posteriores como la soldadura, no pueden estar encima o muy cerca de huellas... etc.

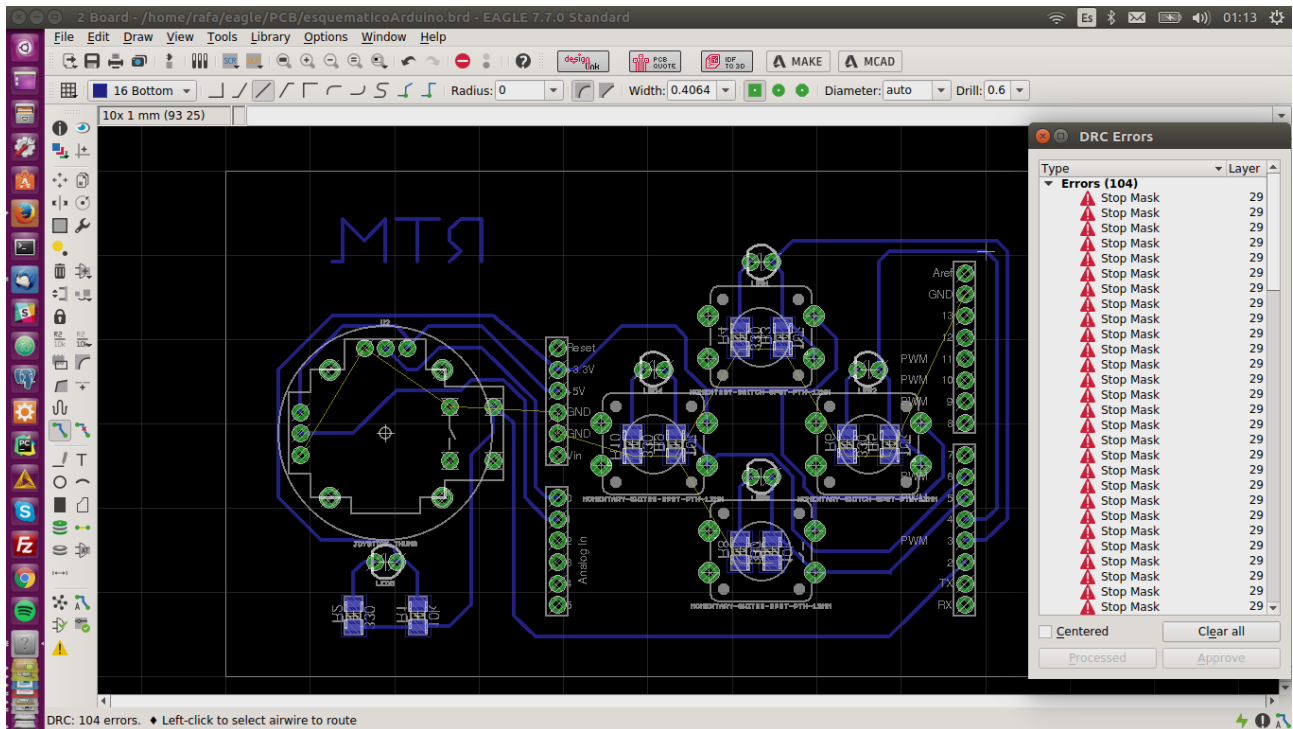
Fijándome en el esquema he colocado mejor las resistencias y los leds



En ciertas ocasiones es muy útil modificar el grid para poder trabajar correctamente, cuidado con no confundir milímetros con pulgadas.



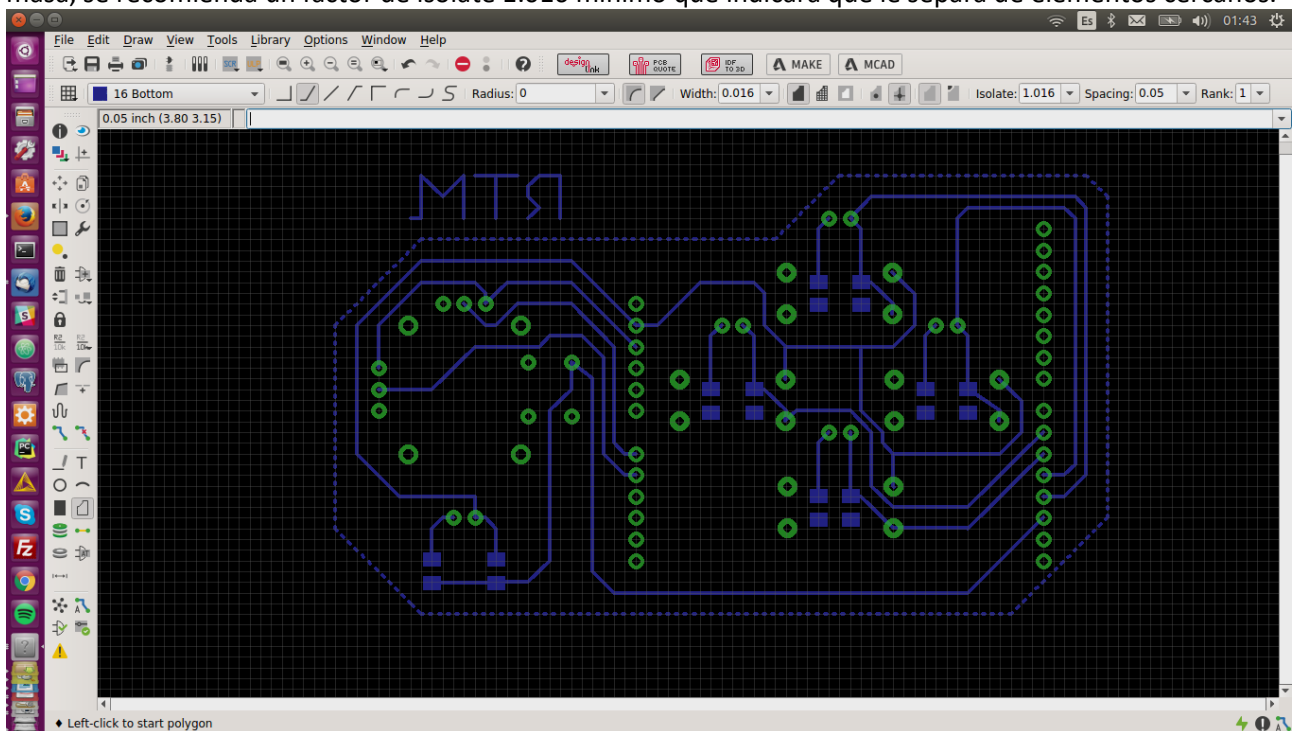
Design rule check muestra este tipo de errores que podemos ignorar



No electrical rule errors

Plano de masa

Usando la herramienta polígono creamos una forma que recubra nuestra placa para que actúe de plano de masa, se recomienda un factor de isolate 1.016 mínimo que indicará qué le separa de elementos cercanos.



Como recordatorio:

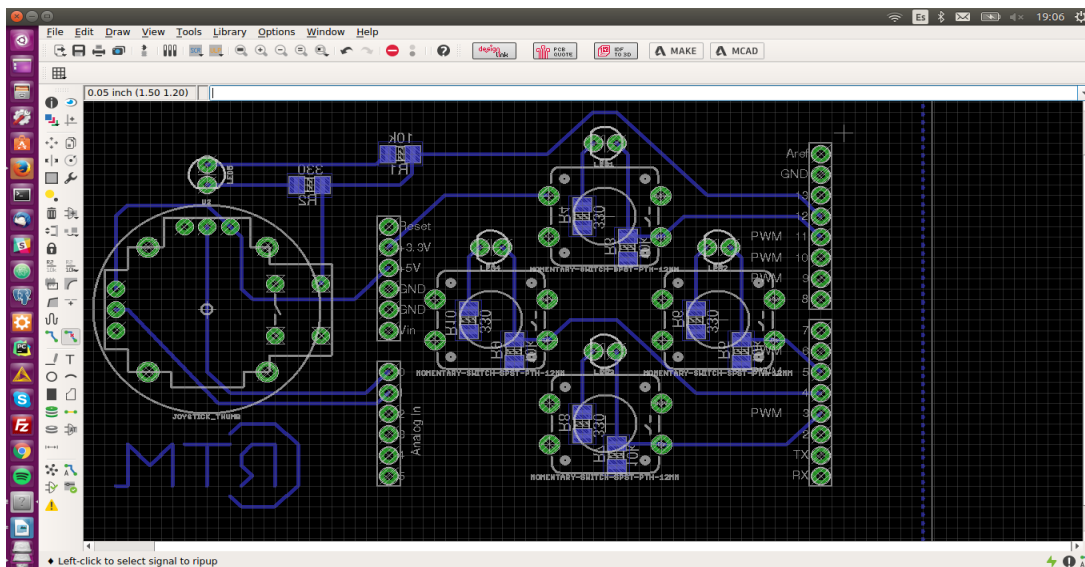
Cuidado con confundir pulgadas con milímetros. En mi caso, un isolate de 1,016 significaba más de 2,5 centímetros puesto equivoqué la medida. El resultado fue un plano de masa que parecía no existir. Y así era, tanta separación no permitía que se mostrase.

Otro detalle importante a no olvidar:

En mi caso, usé una huella para el **switch** que, aunque coincide con la que se recomienda en un principio en pads, no tiene una conexión entre los pines que se comparten 2 a dos teóricamente.

Conociendo este detalle, he ignorado los rats de eagle y conectado teniendo en cuenta este detalle.

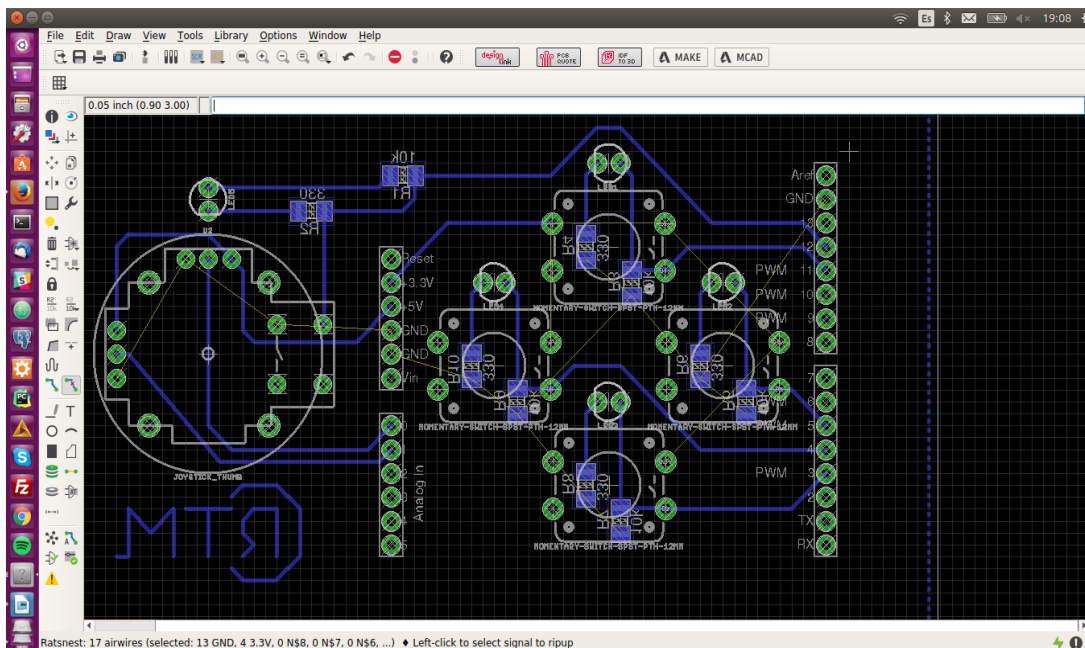
Para conectar los pads por el sitio contrario y no liarme, he usado **Airwires hidden**. Esta utilidad permite ocultar conexiones que no quieres hacer para que no molesten. Util para ocultar las conexiones GND si pretendemos crear un plano de masa posterior o para mi caso con los switches.



¿cómo volver a mostrarlos?

Consola:

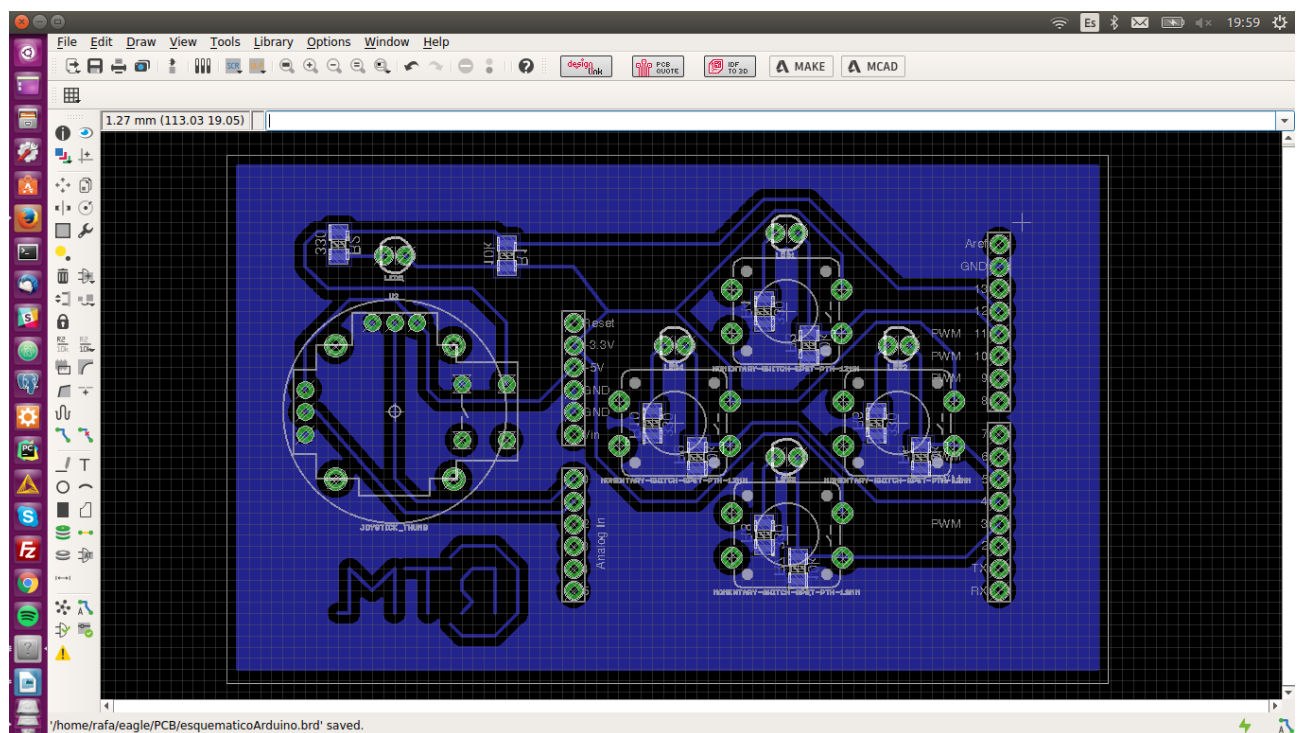
RATSNEST *



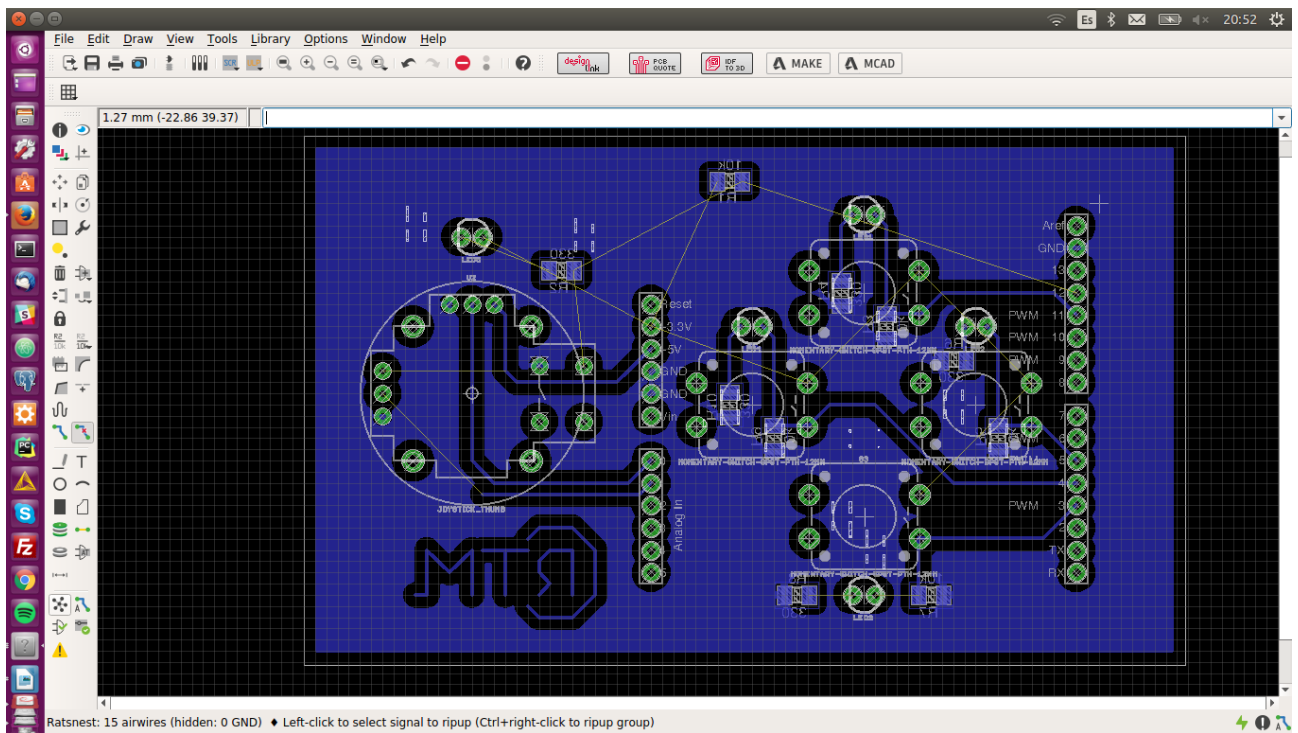
The screenshot displays a KiCad PCB layout for a custom board. The board dimensions are 0.1 inch (5.2-1.8) pitch. The layout includes the following components and connections:

- U1: ARDUINO_SHIELD_NO_SILK** (Arduino Uno) connected to the top layer.
- U2: JOYSTICK** connected to the bottom layer.
- LED1, LED2, LED3, LED4** (LEDs) connected to the bottom layer.
- R1, R2, R3, R4, R5, R6, R7, R8, R9, R10** (Resistors) connected to the bottom layer.
- S1, S2, S3, S4** (Switches) connected to the bottom layer.

The layout shows the placement of components, their pin connections, and the resulting copper pour. The board is oriented vertically, with the top layer on the left and the bottom layer on the right. The components are labeled with their respective values and footprints.



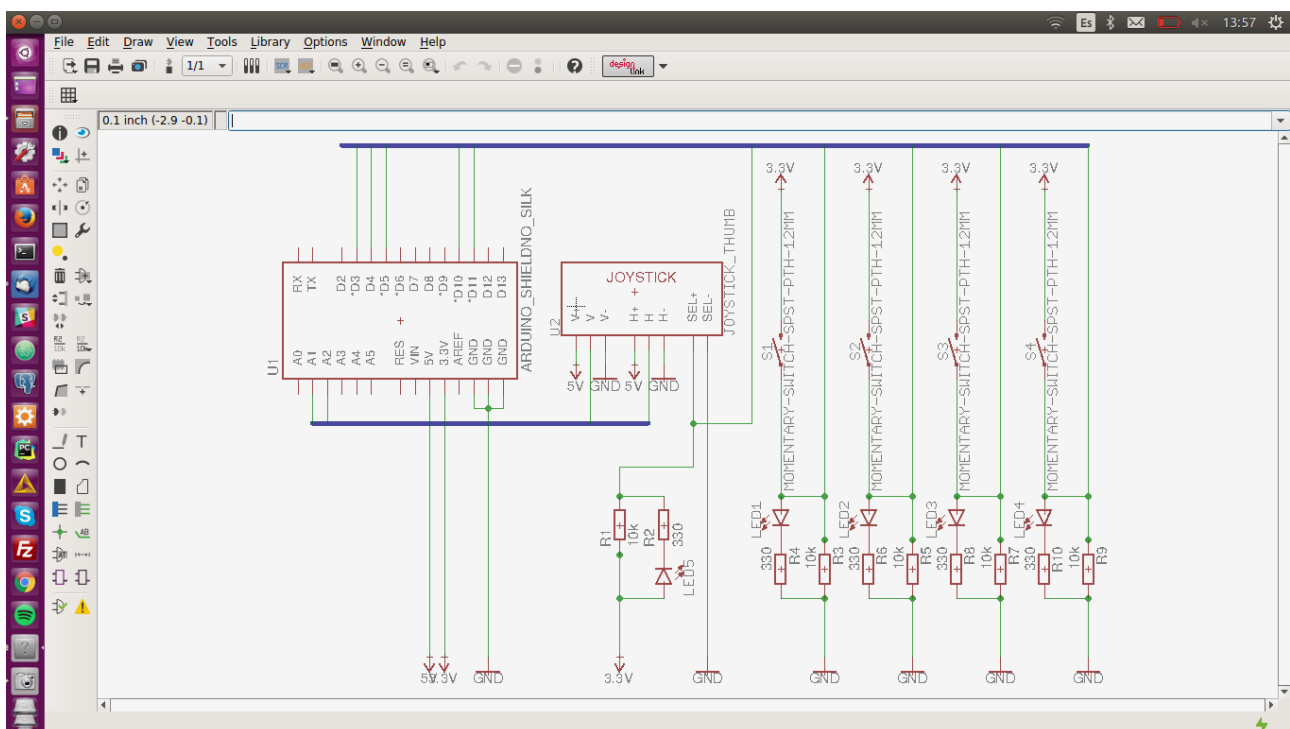
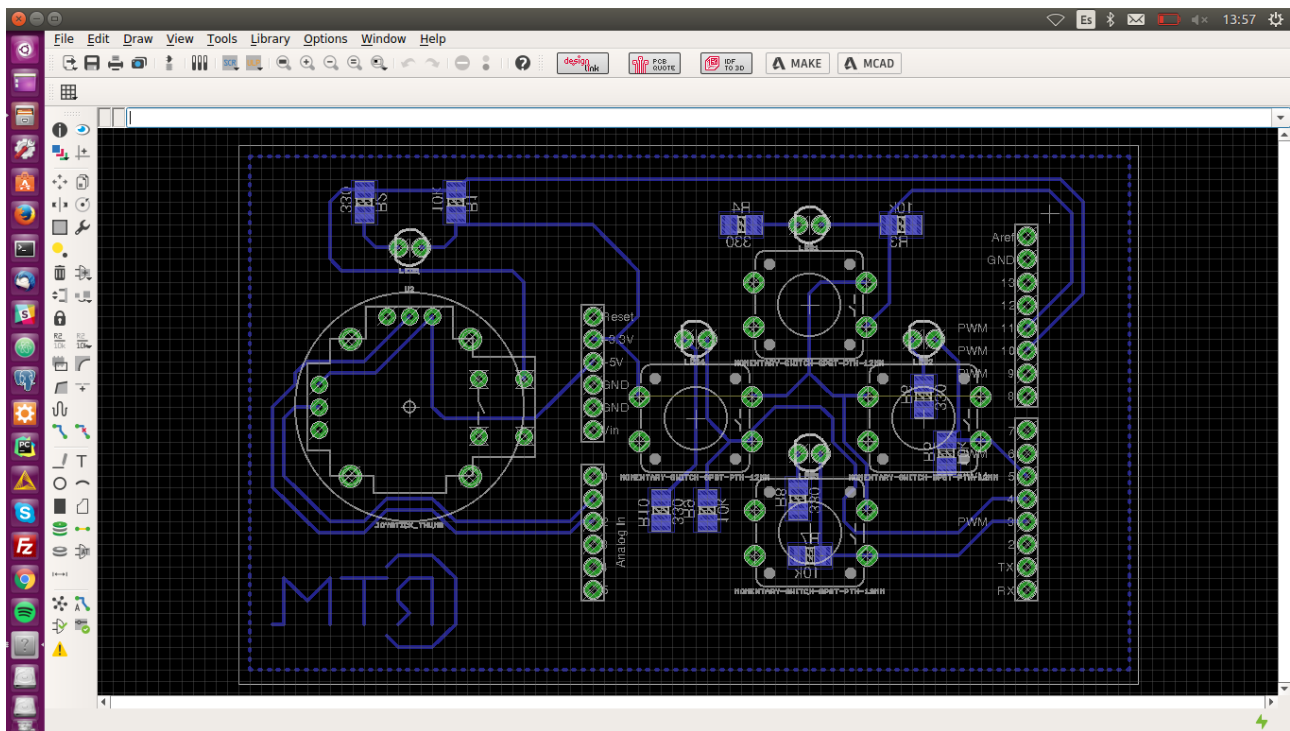
En este momento entiendo que para mí sería más cómodo trabajar con el plano de masa creado desde el principio puesto que es algo que verdaderamente restringe la colocación de los componentes.



1,027 de isolate para facilitarme algo más la soldadura que viene después.
Con la ayuda de la herramienta **ratness** vamos refrescando cualquier cambio.

FINAL

Este es el diseño que más me ha gustado, con los leds colocados encima de cada uno de los pulsadores que lo activan.



Generar Gerbers

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

How to globally sho... MEAM.Design : S62 : G... +

medesign.seas.upenn.edu/index.php/Guides/GeneratingGerberFilesForS62

MEAM.Design : S62 : Generating Gerber Files For S62

GENERAL
Hall of Fame
Laboratories
Contact Info

COURSES
MEAM 101
MEAM 201
MEAM 410/510
MEAM 520
IPD 501
SAAST

GUIDES
Materials
Laser Cutting
3D Printing
Machining
ProtoTRAK
PUMA 260
PHANToM
BeagleBoard
MAEVARM
Phidget
Tap Chart
S62

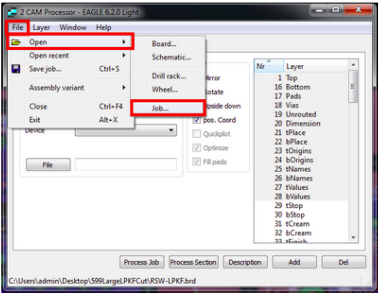
SOFTWARE
SolidWorks

MEAM.Design - S62 - Generating Gerber Files For S62

<< Preparing an Eagle Board File for the S62 | S62 | Generating tool paths from Gerber Files >>

We use the CAM processor in Eagle to generate the necessary Gerber and Excellon files for the S62. Download the two necessary .cam files here: [LPKFCAMFiles.zip](#)

Accessing CAM Processor and Jobs:
To access the CAM processor go to [File->CAM Processor] from the Eagle Board Layout window.
The CAM parameters will be empty at this time. You can load either of the .cam files by selecting [File->Open->Job] from the CAM Processor window and then selecting the .cam files using resulting [Open] window.



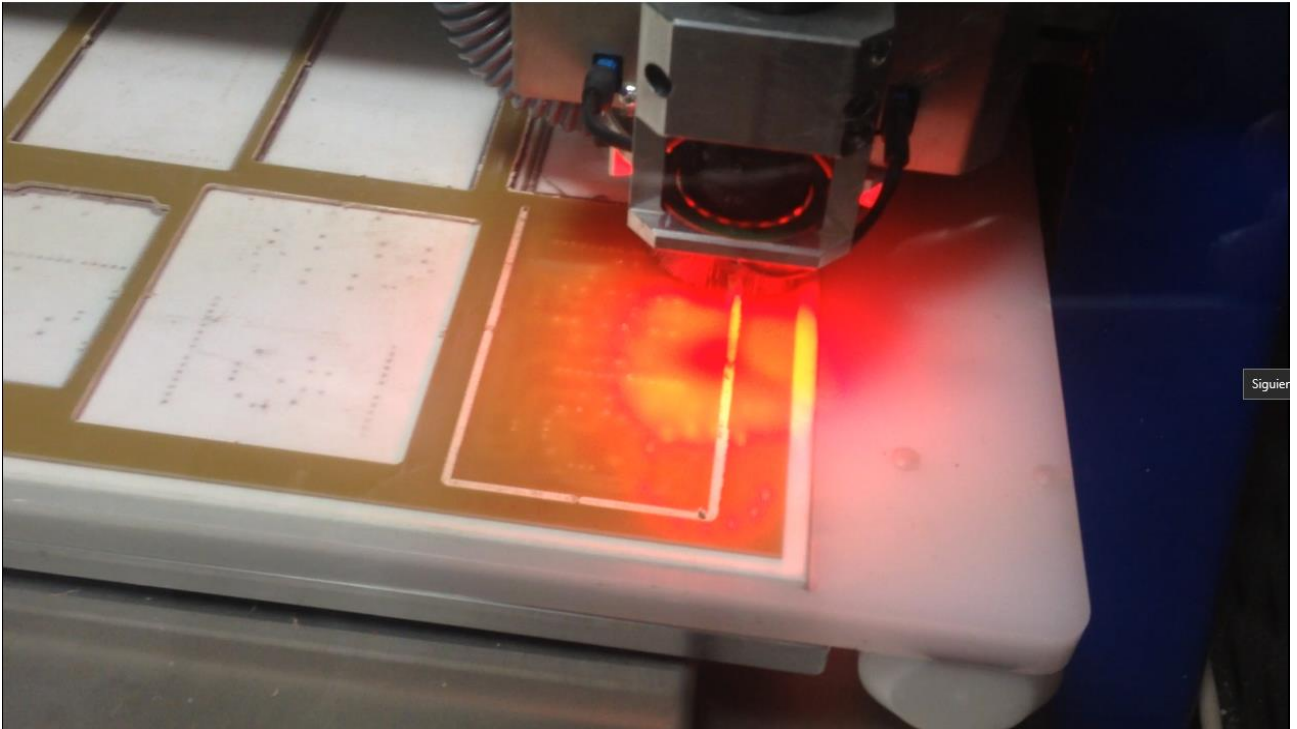
Using excellon.cam
Opening this job will load a single tab of parameters as shown below. Make sure that only holes and drills is selected in the layer list. This job will look at the drill and holes layers and create the drill data that CircuitCAM will need. No changes to the parameters are necessary. Simply, press the [Process Job] button and the data will be generated in the same directory as the .brd file loaded in Eagle.

Descargamos el zip y dentro encontramos los 2 trabajos (jobs) para generarlos, primero excellon y después los correspondientes para el modelo concreto donde será fresada la placa.

Si no modificamos el campo File, los archivos irán al directorio del proyecto, es lo más recomendable.

Una vez tenemos todo, creamos un zip que contiene todo lo necesario para generar esa placa.

Fresado y corte



Ensamblaje de componentes

En este punto iremos colocando cada uno de los componentes que tiene nuestro diseño, teniendo en cuenta polaridad, valores, etc. Cuidado también con colocar las masas adecuadamente.

Los pines importante que vayan siempre de 2 en 2 como mínimo para una mejor sujeción como shield.

Una vez colocados y seguros, comenzamos el proceso de soldadura, principalmente hay que cuidar:

Cantidad de estaño utilizada. Las pistas están muy cerca y es posible dejar caer estaño entre ellas formando puentes.

Cuidado además con los componentes que está cerca unos de otros y finalmente soldar las patillas correctamente.

Es importante observar el esquemático y board

Testado de la placa

Un programa sencillo en arduino que capture el valor digital y analógico de los puertos utilizados y los mande por el puerto de serie para poder observarlo desde el mismo monitor de Arduino.

```
#include <stdio.h>

const int inD1 = 11;
const int inD2 = 5;
const int inD3 = 3;
const int inD4 = 4;
const int inD5 = 10;
int datos[6];
char text = 0;

void setup() {

    Serial.begin(9600);

    pinMode(inD1, INPUT);
    pinMode(inD2, INPUT);
    pinMode(inD3, INPUT);
    pinMode(inD4, INPUT);
    pinMode(inD5, INPUT);

}

void loop() {

    int inA1 = analogRead(A1);
    int inA2 = analogRead(A2);
    datos[1] = digitalRead(inD1);
    datos[2] = digitalRead(inD2);
    datos[3] = digitalRead(inD3);
    datos[4] = digitalRead(inD4);
    datos[5] = digitalRead(inD5);
    char buffer [50];
    text = sprintf (buffer, "V: %d   H: %d   B1: %d   B2: %d   B3: %d   B4: %d
B5: %d", inA1, inA2, datos[1], datos[2], datos[3], datos[4], datos[5]);
    for(int l= 0; l<=text; l++)
        Serial.print(buffer[l]);
        Serial.println();
    delay(1);          // delay para estabilidad
}
```

