

# **Práctica Bloque 2: Desarrollo de PCB**

Rafael Escudero Lirio ([rafaeslir@gmail.com](mailto:rafaeslir@gmail.com))

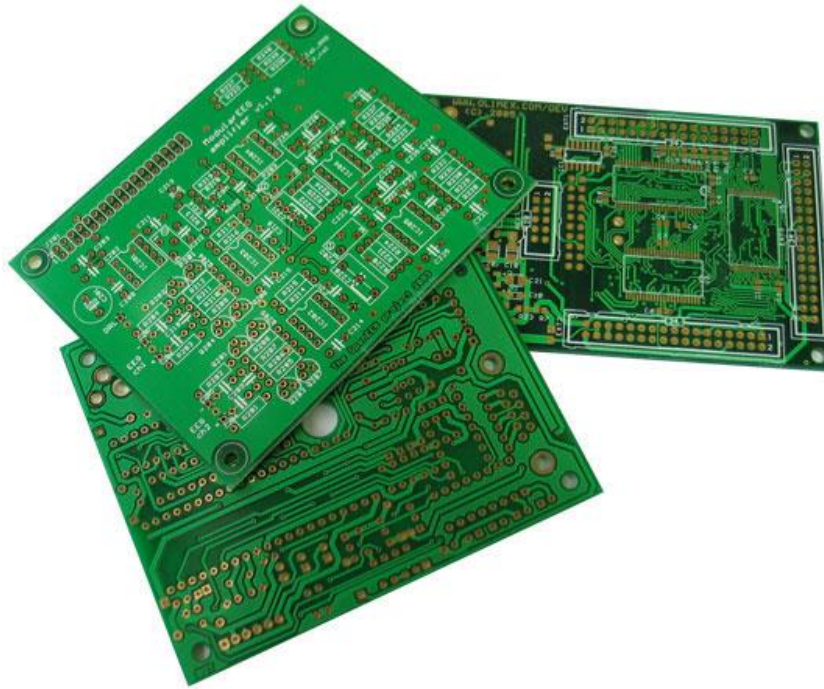
Ingeniería Informática de Computadores

## Introducción

En esta práctica se va a realizar el diseño de un PCB.

Posteriormente se va a fabricar, ensamblar y testar dicho PCB.

¿Qué es un PCB?



En electrónica, “circuito impreso”, “plaqueta de circuito impreso” (del inglés: Printed Circuit Board, PCB), es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos. Las pistas son generalmente de cobre mientras que la base se fabrica generalmente de resinas de fibra de vidrio reforzada, Pertinax, pero también cerámica, plástico, teflón o polímeros como la baquelita.

Para el diseño usaremos software especializado, en este caso la herramienta CAD EAGLE.



**EAGLE**<sup>®</sup>  
www.cadsoft.de

La PCB que se pretende fabricar va a ser una placa de expansión bien para ARDUINO o bien para ZPUINO.

Una vez fabricado el PCB se pasará a completar la práctica con las siguientes tareas:

- Ensamblaje de componentes
- Testado de la placa
- Desarrollo de software sobre Arduino/zpuino verificando la placa de expansión desarrollada
- Desarrollo de software sobre PC utilizando el mando.

## **Objetivos**

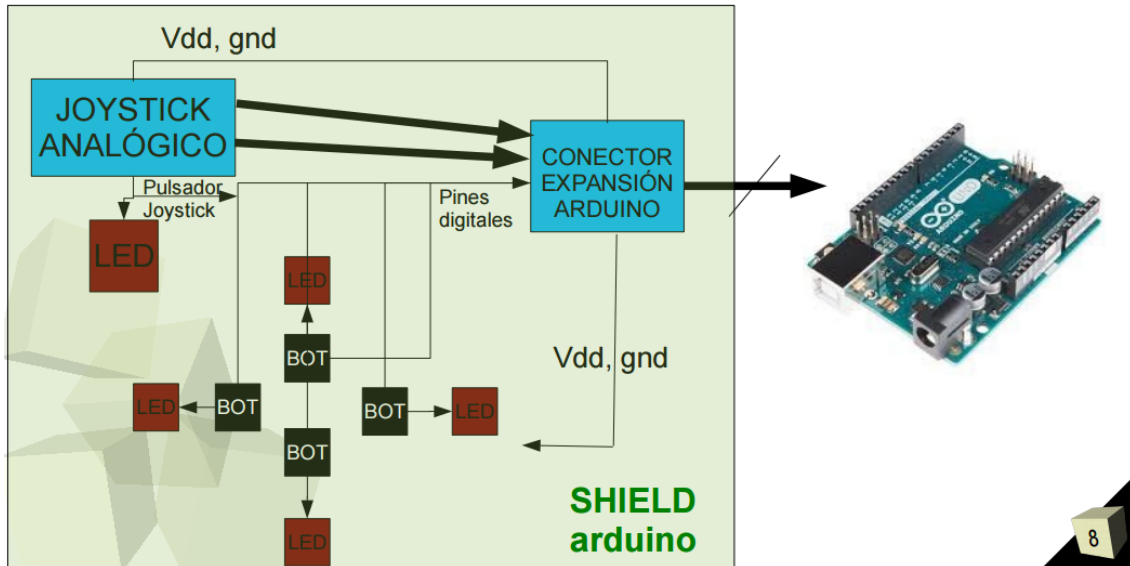
-Se pretende diseñar una placa de expansión para la placa de desarrollo arduino/papilio-zpuino que funcione de manera similar a un mando de consola: con joystick y pulsadores.

-Aprender acerca del proceso de desarrollo de una placa PCB en sus diferentes fases.

-Aprender a usar las diferentes herramientas tanto físicas como a nivel software para el desarrollo de un PCB (Herramienta CAD, Soldador, Polímetro, etc...)

## Diseño básico del sistema

El diseño final elegido será un sistema para Arduino que funcionará como shield del mismo.



Tal y como se muestra en la imagen, el sistema se conecta a los pines del Arduino UNO directamente, la señal que llegará del joystick será analógica, lo cual no supone un problema en el diseño para arduino puesto que disponemos de entradas analógicas.

### Componentes del sistema

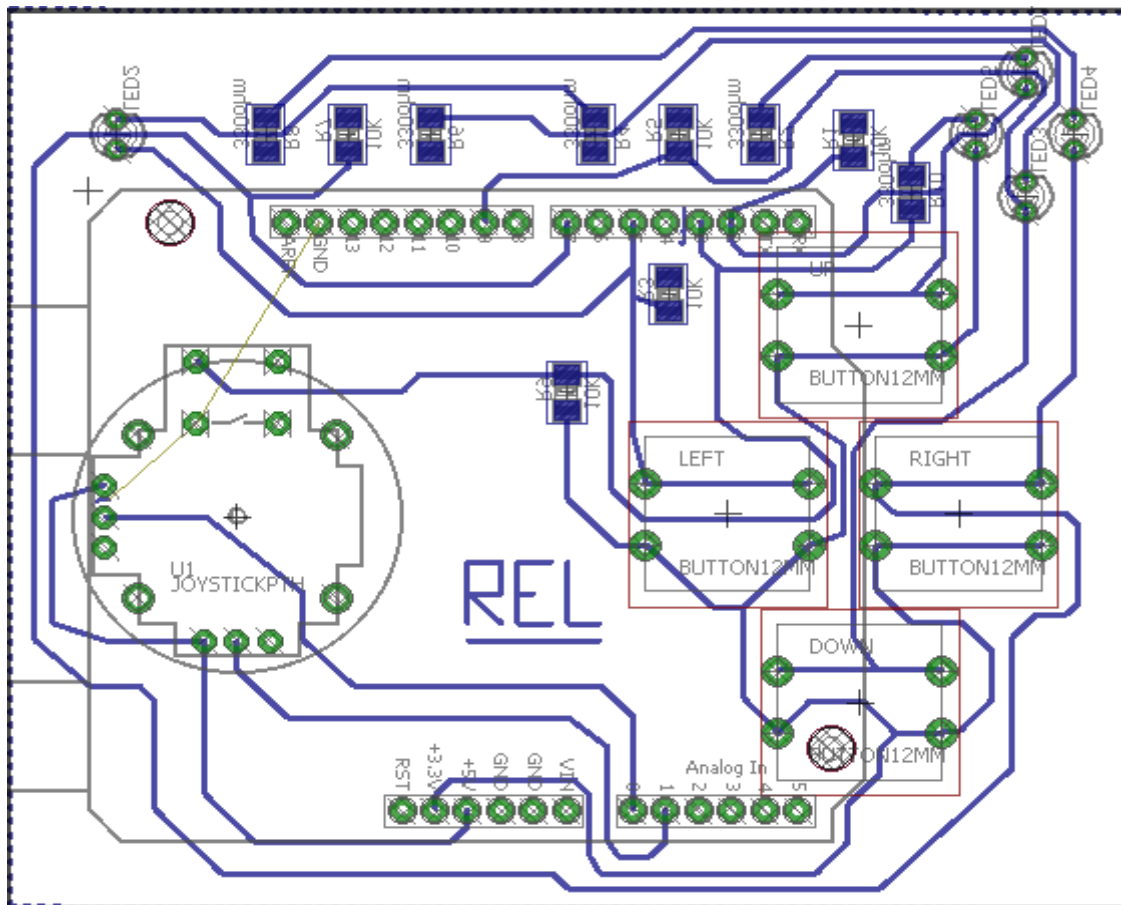
- Joystick analógico: Genera señal vertical y horizontal analógica entre VDD (5v) y GND (0v). Dispone de pulsador central. Le conectaremos LED. El pulsador debe ir conectado a un pin digital de la placa.
- Convertidor Analógico/Digital(Sólo para Zpuino): ADC. Recibirá las señales analógicas del Joystick para digitalizarlas y enviarlas a la placa a través de interfaz SPI
- Pulsadores: UP, DOWN, LEFT, RIGT. Tendrán resistencia de PULL-DOWN. Los conectaremos a LEDs. Irán conectados a pines digitales de la placa
- Conectores para los terminales de la Placa.
- Diodos LED para cada uno de los pulsadores.

El componente más a la izquierda es el joystick, que está conectado a un quinto botón (el que se acciona al pulsar el propio joystick).

Los pines de Arduino elegidos son:

- Se elige esta disposición para tener una mayor facilidad a la hora de diseñar el layout.

Para que el diseño en Eagle sea completo necesitamos crear el “Layout”, el cual sirve como esquema de la conexión y disposición de los componentes en la PCB una vez impresa.



La herramienta de EAGLE no es difícil de usar puesto que nos señala cuales son las conexiones necesarias para que nuestro sistema funcione como indica el esquemático.

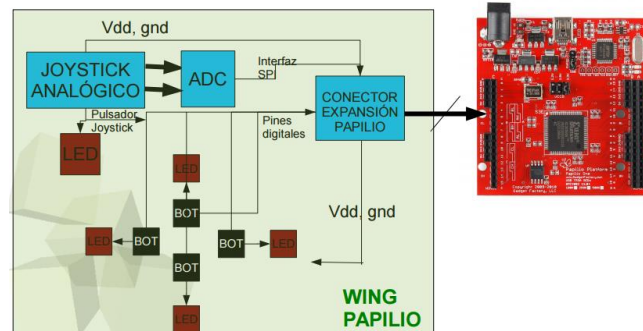
Como se muestra en la imagen a la derecha parecen los cuatro botones. Cada uno de ellos conectado en primer lugar a un LED y en segundo lugar a una resistencia de 330 ohm y de 10 Kohm cada una.

A la izquierda aparece el joystick usado, de una de las bibliotecas de sparkfun, conectado a otras dos resistencias y a un quinto LED debido que también sirve como pulsador.

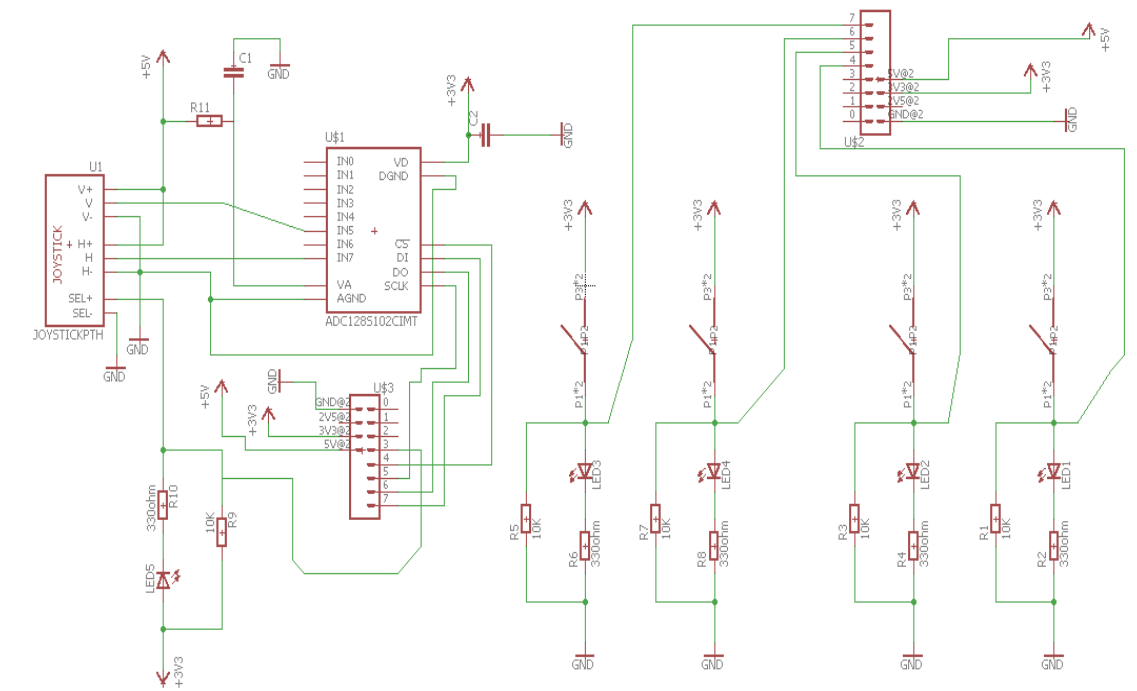
Los “Cuadrados azules” representan las diferentes resistencias y los “círculos verdes” son cada uno de los Pads, es decir, los agujeros a realizar por el taladro en la placa y que nos servirán para conectar y ensamblar los diferentes componentes y pines del shield.

Por último las “líneas azules” son las vías de la placa que funcionan a modo de cable para conectar los componentes.

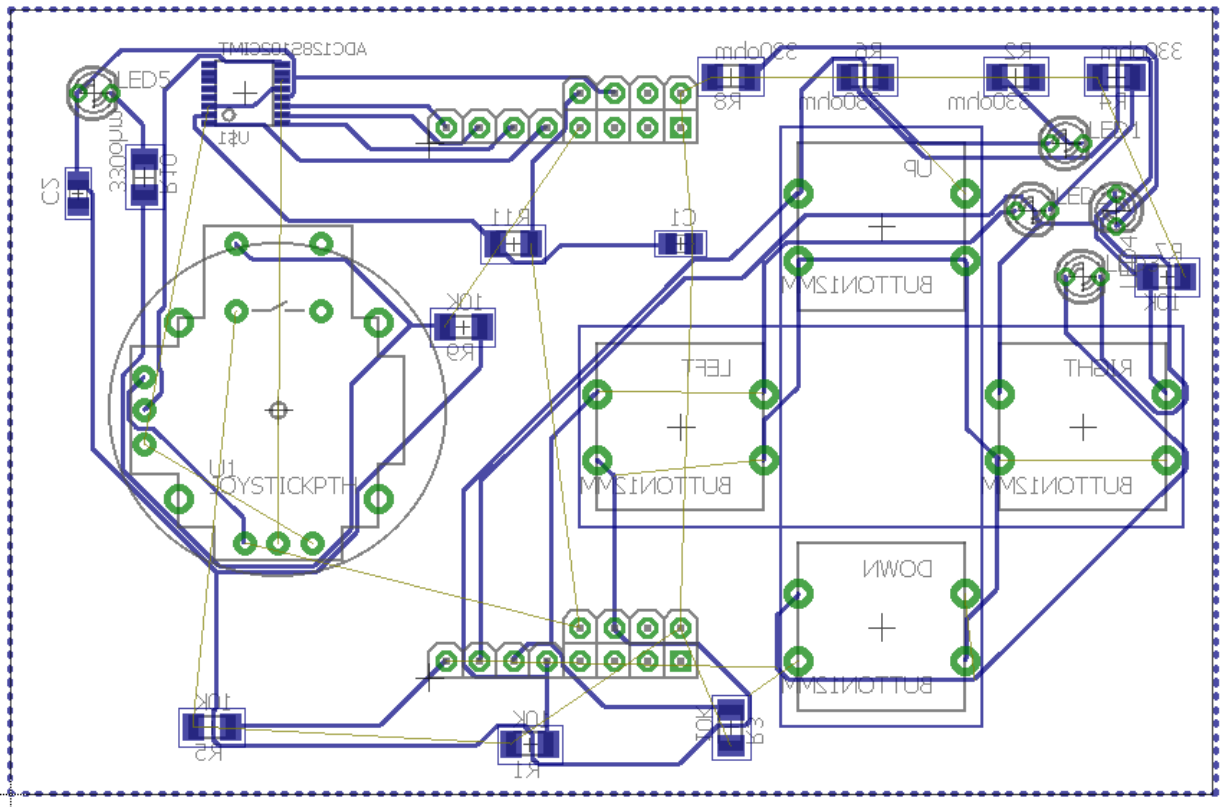
**A continuación se muestra el diseño alternativo en EAGLE de la placa de expansión para papilio.**



Se han añadido un conversor ADC y dos tiras de pines compatibles con papilio. Se observan dos partes, a la derecha la tira con los botones, a la izquierda el joystick con el conversor.



Su respectivo Layout:



Al tener más experiencia con EAGLE el PCB tiene una conexiones menos engorrosas. Aunque si es más complicada la conexión del ADC por su tamaño.

Puede observarse en los esquemas que ninguna de las pistas de tierra están conectadas, esto es debido a que usamos un plano de masa, Cubriendo las partes no conectadas de cobre que a efectos prácticos son un pin a tierra.

### Creación de los Gerbers

Mediante la opción CAM de EAGLE creamos los gerbers que son archivos con la información binaria necesaria y suficiente para imprimir nuestro PCB en una máquina.

Los Archivos requeridos son: .cmp, .bor, .dri, .drd, .sol

Los cuáles se encargan de determinada información como la posición de las vías o de los sitios en los que se debe taladrar.



## Ensamblaje

La siguiente fase de la práctica consiste en ensamblar los componentes de la placa y soldarlos, para ello usaremos:

- Soldador



A una temperatura de 400°C nos servirá para fundir el estaño y con éste soldar los componentes.

- Rollo de Estaño:



Como se acaba de mencionar para “unir” los componentes a la placa, el usado en la práctica tiene una composición en su mayoría de estaño (99%~)

- Flux



Nos ayudará a soldar la placa con mayor facilidad, al evitar en cierta medida que el estaño llegue a las partes sin cobre entre las vías y evitará la oxidación.

- Polímetro



Con el que comprobar las conexiones de la PCB.

En primer lugar, sin empezar a soldar se debe comprobar con el polímetro que todas las vías y conexiones del PCB recién impreso funcionan correctamente.

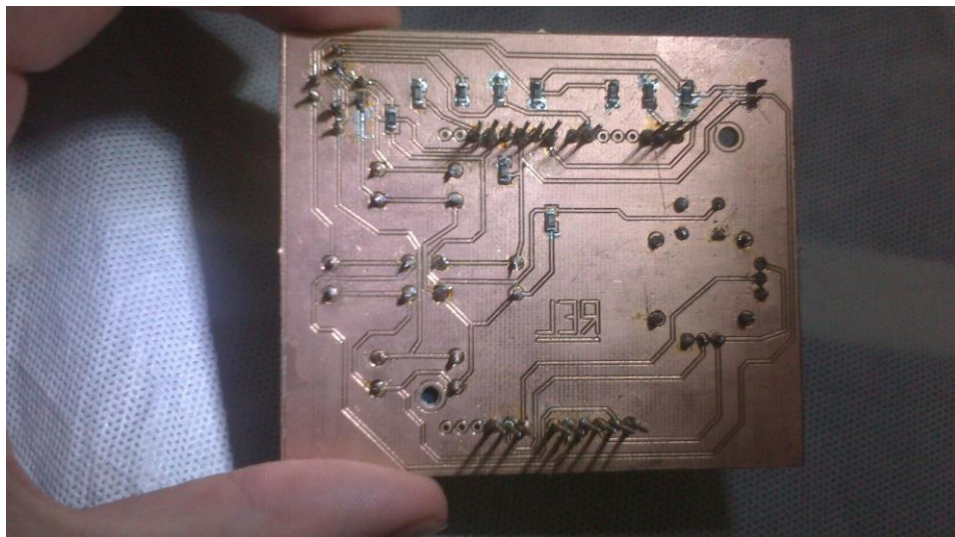
Una vez hecho esto podemos empezar con el proceso de soldadura, el orden de los componentes no es obligatorio, pero por cuestiones de facilidad se han ensamblado primero las Resistencias, LEDS, Botones, Joystick, Tiras de pines.

El proceso concreto de soldadura no es complejo pero si requiere de cierta “maña” a la hora de usar el soldador. Uno a uno iremos cogiendo los componentes, lo colocamos en su posición si tiene pines o lo sujetamos con unas pinzas si se trata de una resistencia, colocamos el hilo de estaños donde vamos a soldar y aplicamos el soldador. Después de esto comprobamos con el polímetro las conexiones y así sucesivamente.

En caso de error, o de haber un exceso de estaño podemos usar una malla de desoldadura como la de la imagen, bastaría con abrir un poco el extremo de la malla, colocarla encima de estaño y aplicar calor con el soldador para eliminar el estaño.



Una vez terminado el proceso de ensamblaje, nuestro PCB tiene este aspecto por la cara con cobre:



Y este otro por la cara con los componentes:



## Testado de la placa

El último paso es comprobar que la placa funciona correctamente, al ser un shield de arduino podemos comprobarlo esto fácilmente mediante un sencillo código en su entorno:

```
void setup() {  
  
  //Inicializamos los correspondientes pines a sus botones  
  
  Serial.begin(9600);  
  
  pinMode(2, INPUT); //UP  
  pinMode(3, INPUT); //JT  
  pinMode(5, INPUT); //LEFT  
  pinMode(7, INPUT); //RIGHT  
  pinMode(9, INPUT); //DOWN  
}
```

```

void loop() {

    // Leemos las entradas A0 para el eje Y y A1 para el eje X
    int sensorValueY = analogRead(A0);
    int sensorValueX = analogRead(A1);


    int UP = digitalRead(2);
    int JT = digitalRead(3);
    int LEFT = digitalRead(5);
    int RIGHT = digitalRead(7);
    int DOWN = digitalRead(9);

    //Mostramos Y
    Serial.println("valorY");
    Serial.println(sensorValueY);
    delay(100);

    //Mostramos X
    Serial.println("valorX");
    Serial.println(sensorValueX);
    delay(100);

    //Mostramos los botones, JT de joystick está invertido
    Serial.print(UP);
    Serial.print(LEFT);
    Serial.print(RIGHT);
    Serial.print(DOWN);
    Serial.println(JT);

}

```

Ejecutamos este código y sencillamente comprobamos que las salidas de la consola coinciden con los valores correctos.

**Cosas a mejorar:**

- En el layout de la placa se produjo un error al colocar los LEDS, de esta forma los LEDS correspondientes a Joystick y al botón izquierdo están intercambiados.
- Al no conocer bien los componentes, algunas partes del diseño son bastantes enrevesadas, por ejemplo pueden ahorrarse las interconexiones de los pulsadores y algunas de las del joystick.
- Aunque el área de media era de 6x10cm y esto se ha respetado como área total, las medidas finales han sido de unos 7x8cm, aún así el mando es relativamente más cómodo para unas manos grandes de esta manera.