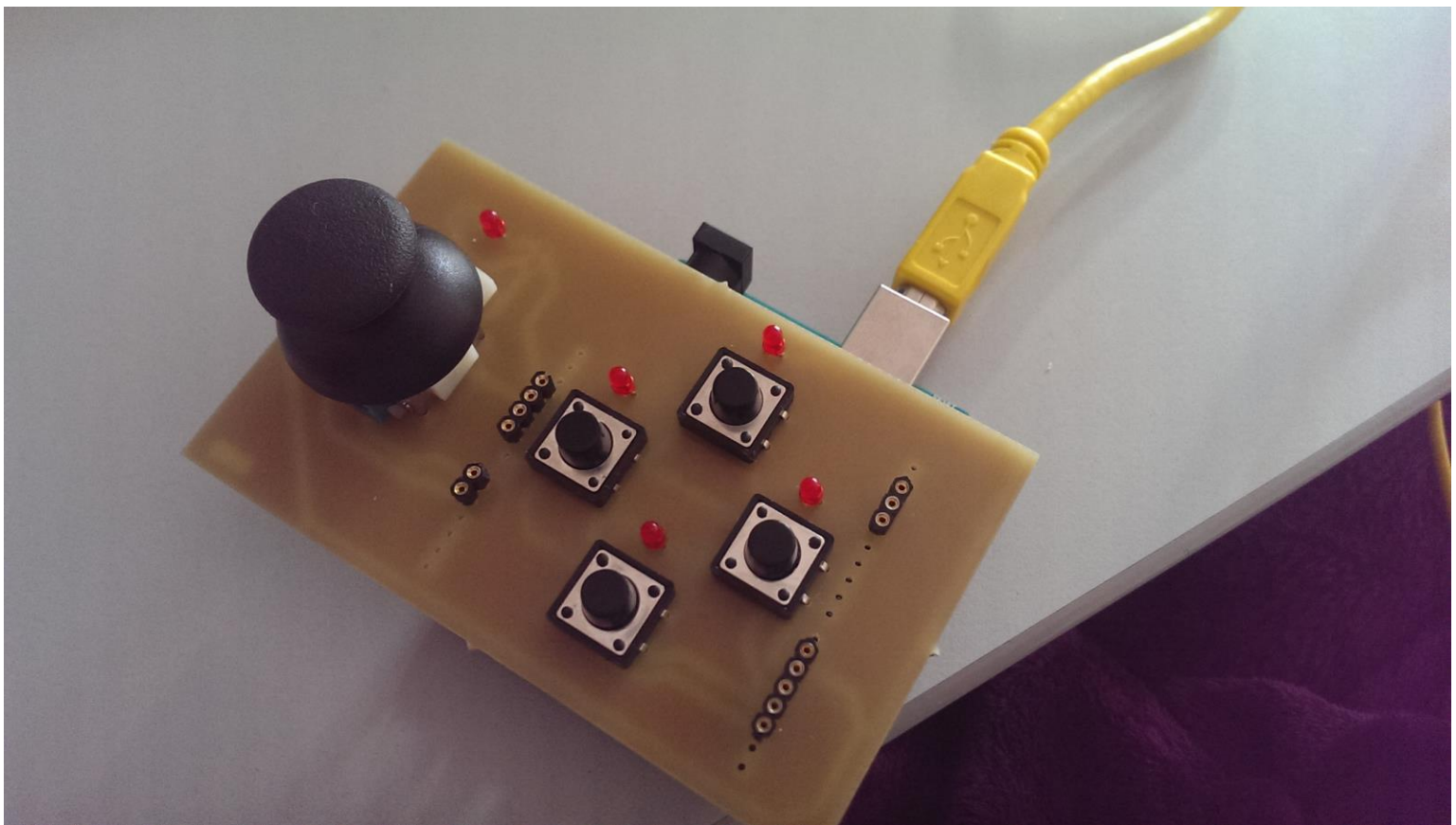


# Diseño de un mando para juegos



Antonio Manuel Núñez Domínguez

# ÍNDICE

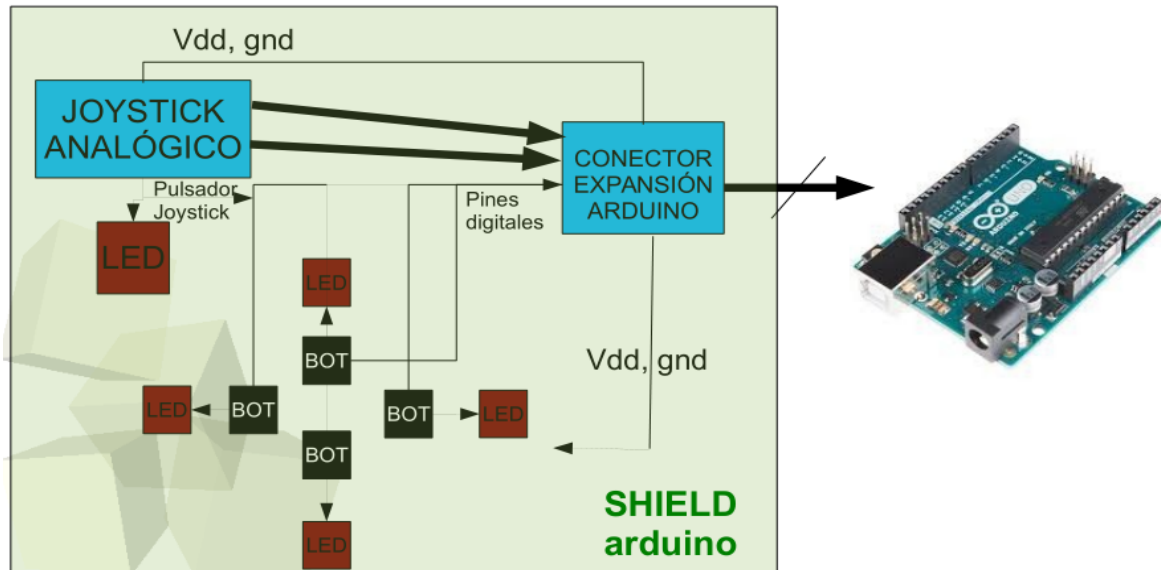
---

· Especificaciones del sistema .....	3
- Partes del sistema electrónico .....	4
· Implementación en PCB del sistema electrónico .....	7
· Fase de diseño .....	7
- Diseño Eagle .....	7
- Diseño Kicad .....	13
- Naked-Proto EURO CIRCUITS .....	15
· Fase de fabricación .....	17
· Fase de ensamblaje .....	19
· Fase de test y desarrollo de Software .....	24

## *Especificaciones del sistema electrónico*

---

En este proyecto, se recogen los pasos realizados para el diseño, fabricación, ensamblaje y test de una placa de expansión para la placa de desarrollo Arduino, para que funcione como el mando de una consola.



**Especificaciones del sistema electrónico**

### Recomendaciones:

- Área máxima: 10cm x 6cm.
- Diseño a una sola cara.
- Tamaño del cable a usar: 0.4064mm (0.016 inch).

## *Partes del sistema electrónico*

---

Las partes principales del sistema electrónico son:

- **Joystick analógico:** Genera señal vertical y horizontal, analógica entre VDD(5V) y GND(0V). Dispone de un pulsador central, que se conectará a un LED. El pulsador debe ir conectado a un pin digital de la placa.
- **Pulsadores:** UP, DOWN, LEFT, RIGHT. Tendrán resistencia de PULL-DOWN. Se conectarán a LEDs. Estos pulsadores deben de ir conectados a pines digitales de la placa.
- **Conectores para los terminales de la placa, tipo wing.** Para este diseño, se emplearán conectores que colocaremos en una posición para que encaje en la placa de desarrollo Arduino.

## *Listado de componentes del sistema electrónico*

---

- Joystick: Thumb Joystick Sparkfun
  - Número de unidades: 1



- Pulsadores: Momentary Pushbutton Switch – 12mm Square
  - Número de unidades: 4



- **Resistencia de 10K y 330 ohm: SMD 1206**
  - Número de unidades: 5 SMD de 10K y 5 SMD de 330 ohm



- **Tiras de pines:**

- Número de unidades: Número de pines que se vayan a conectar.



- **LEDs TH. De 3mm:**

- Número de unidades: 5

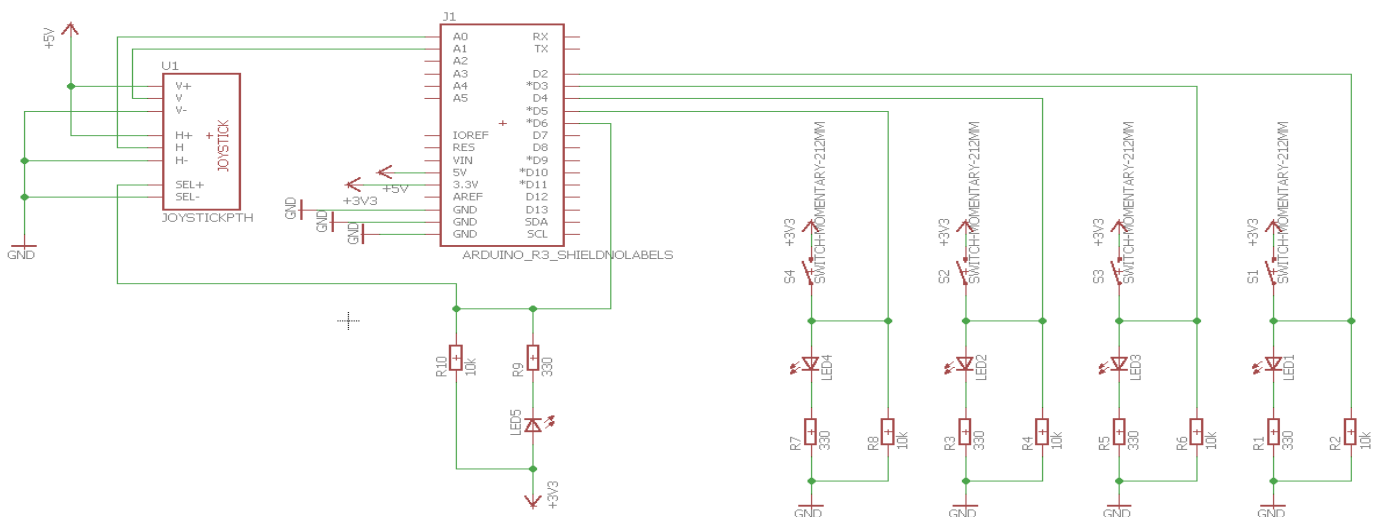


### Fase de diseño

En esta fase se crea el **esquemático** y el **layout** del sistema electrónico. El siguiente diseño se ha realizado en la plataforma EAGLE.

### Diseño en Eagle

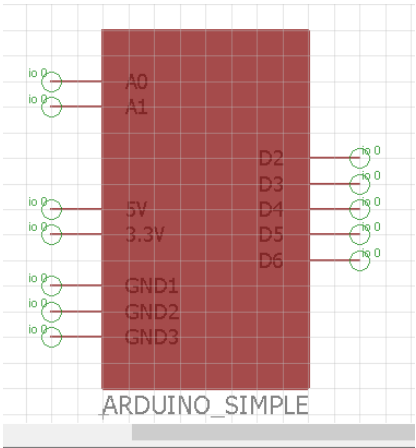
- **Esquemático:** Es el diagrama del circuito electrónico donde aparecen los diferentes componentes electrónicos y las interconexiones entre ellos.



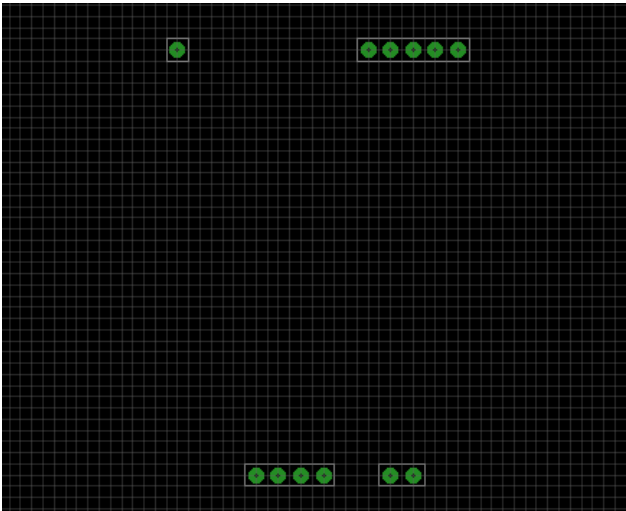
También se ha querido realizar un diseño en el que se utilizaran los mínimos pines posibles para conectar la placa de desarrollo Arduino, y así facilitar la tarea de enrutamiento en la board.

No se tenía una huella para la Arduino que se necesitaba, por lo que se procedió a crear el componente en una librería:

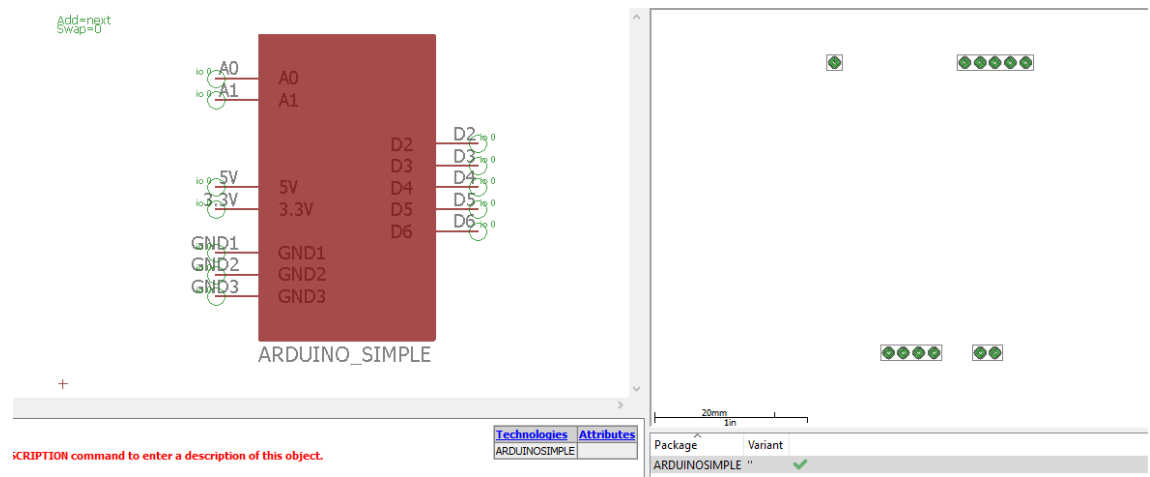
• **Símbolo:**



• **Package:**

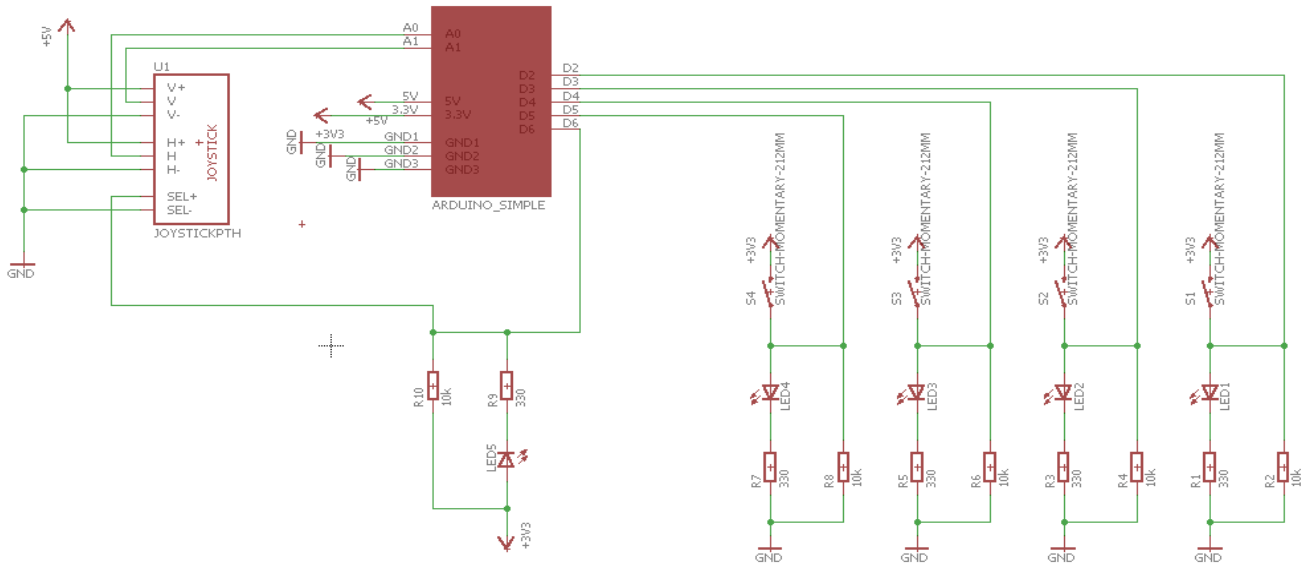


• **Device:**





Finalmente, este diseño no se ha utilizado porque se decidió que convenía tener más pines para ofrecer una mayor estabilidad cuando se conectara la PCB diseñada a la placa de desarrollo Arduino.

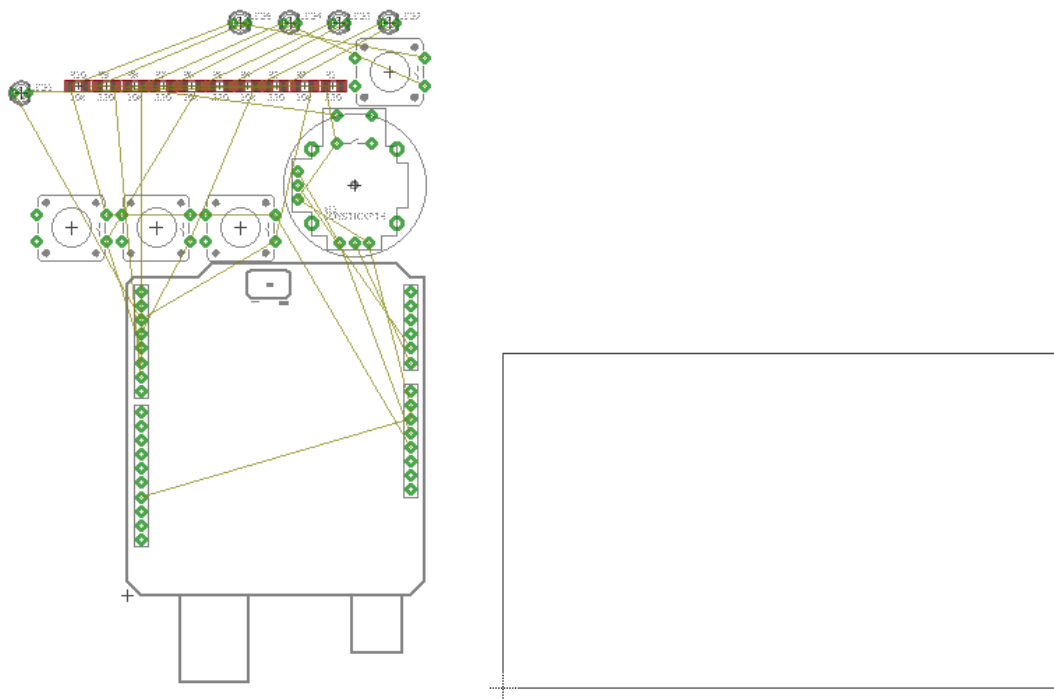


## *Diseño del Layout o Artwork*

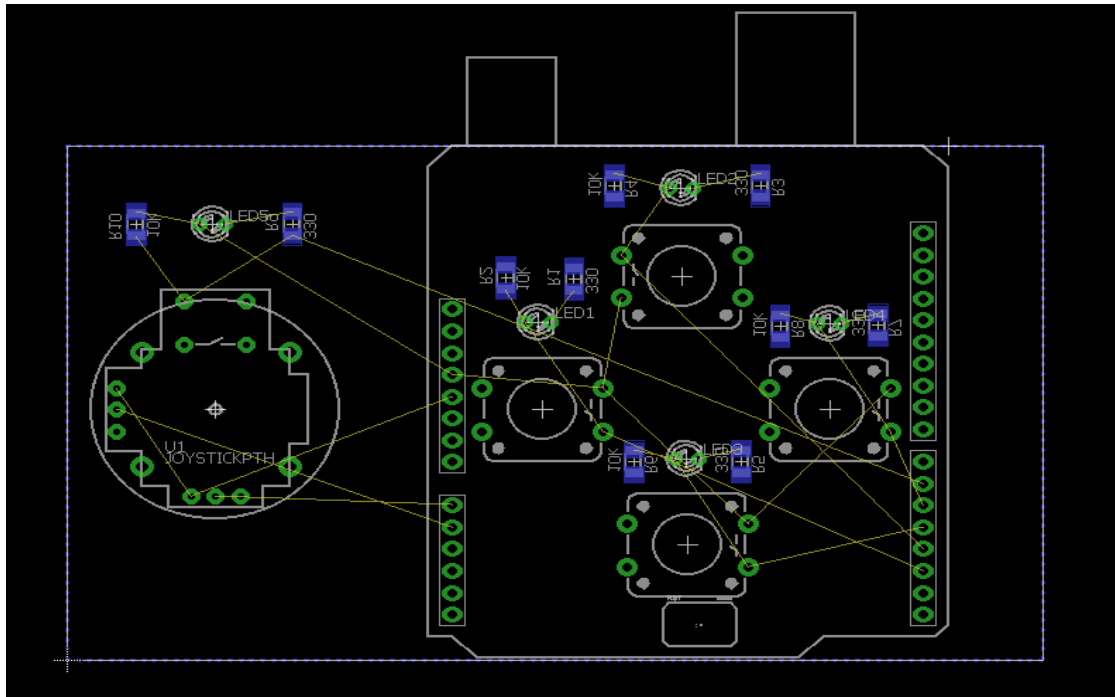
---

- **Layout:** es el dibujo donde aparecen los componentes electrónicos con sus huellas a tamaño real (**footprint**) en la posición que va a ocupar en la PCB final y los caminos de interconexión entre los pines.

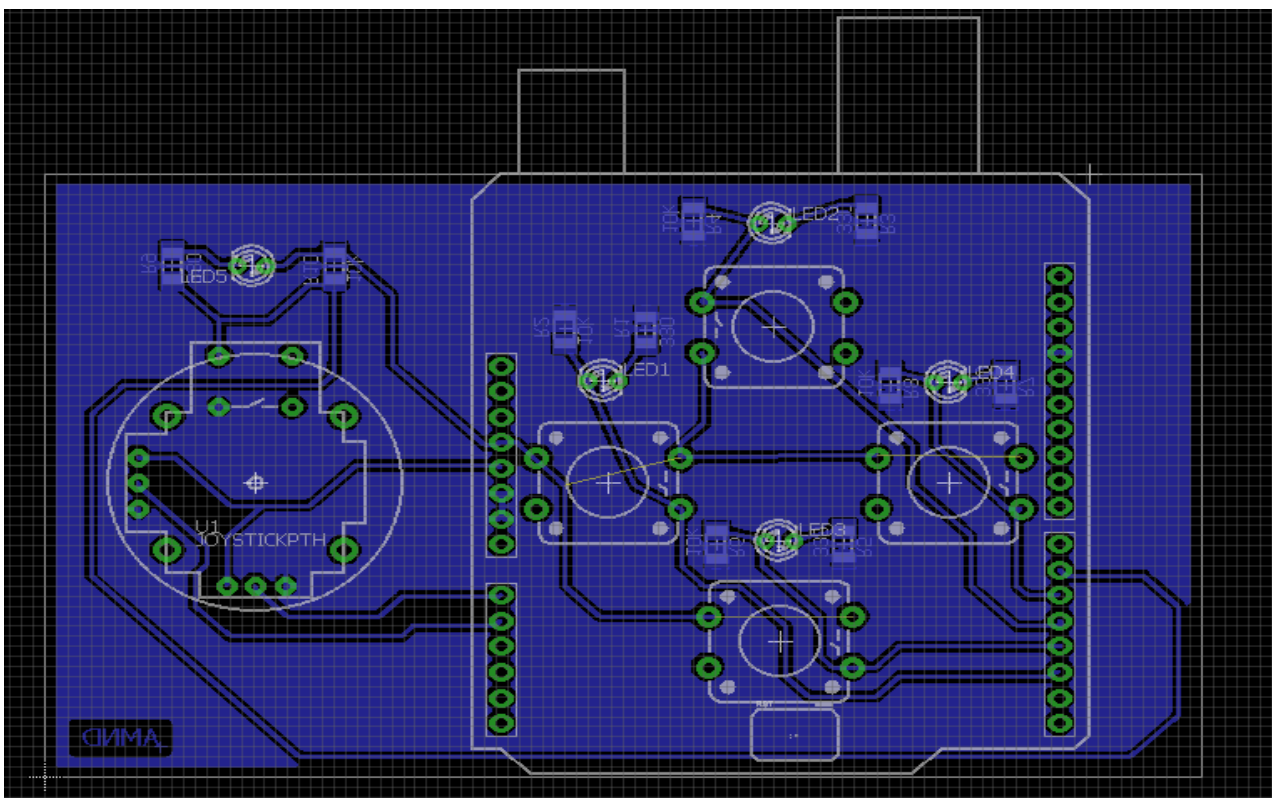
Una vez realizado el esquemático, se procede a realizar la board y el proceso de enrutamiento. Partiendo de la figura siguiente, donde se muestran los componentes del sistema electrónico y un rectángulo, que representa la placa PCB, donde se colocarán los componentes. Lo primero que hay que hacer es modificar las dimensiones de este rectángulo, para que esté a la medida pedida en los requisitos del proyecto. En este caso, una placa de 10cm x 6cm.



En el siguiente paso se procede a colocar los componentes en el lugar deseado sobre la placa:



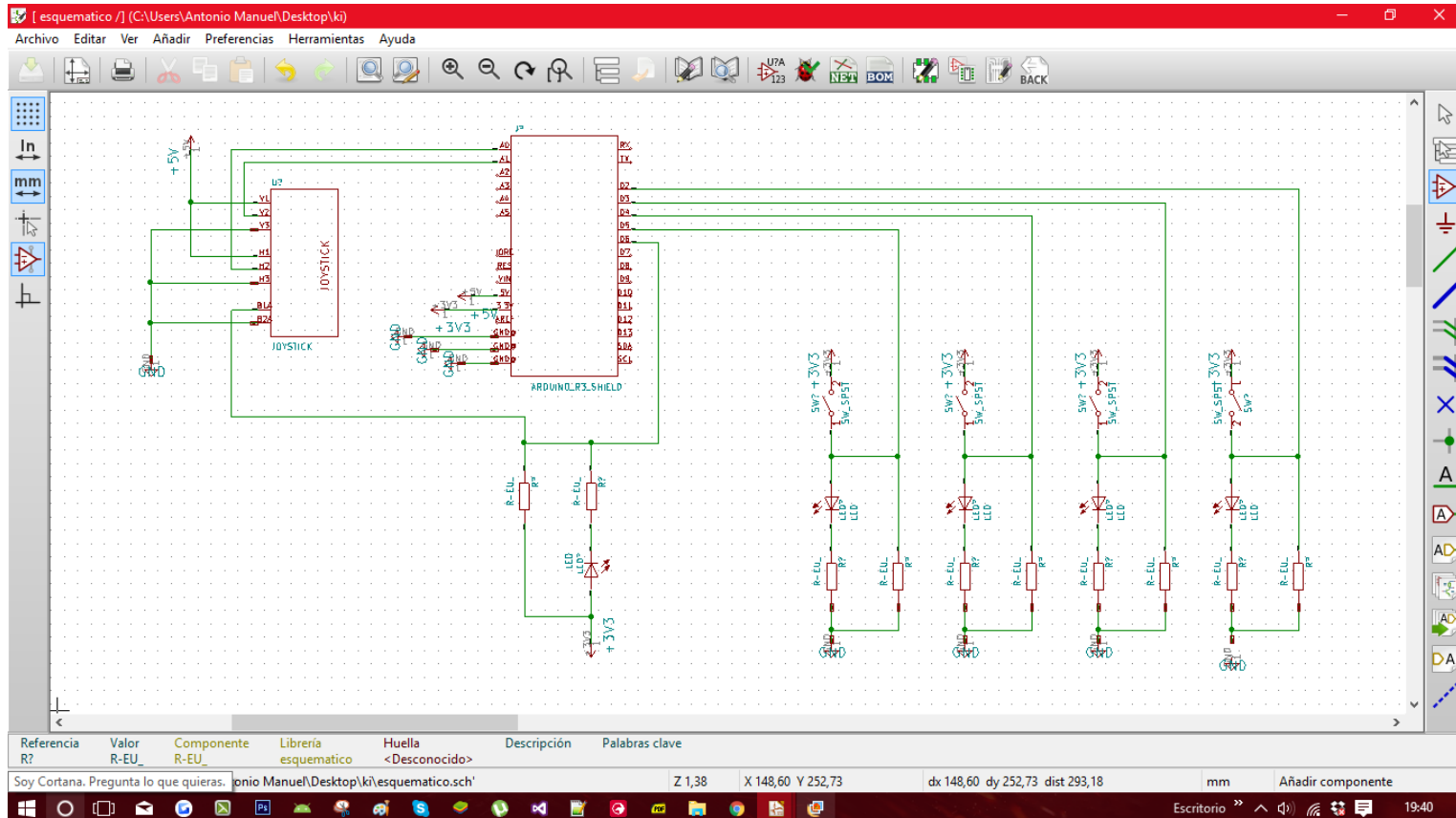
Una vez colocados los componentes, se procede a realizar el enrutado de pistas. Se puede hacer de forma automatizada, pero en este caso se ha realizado manualmente.



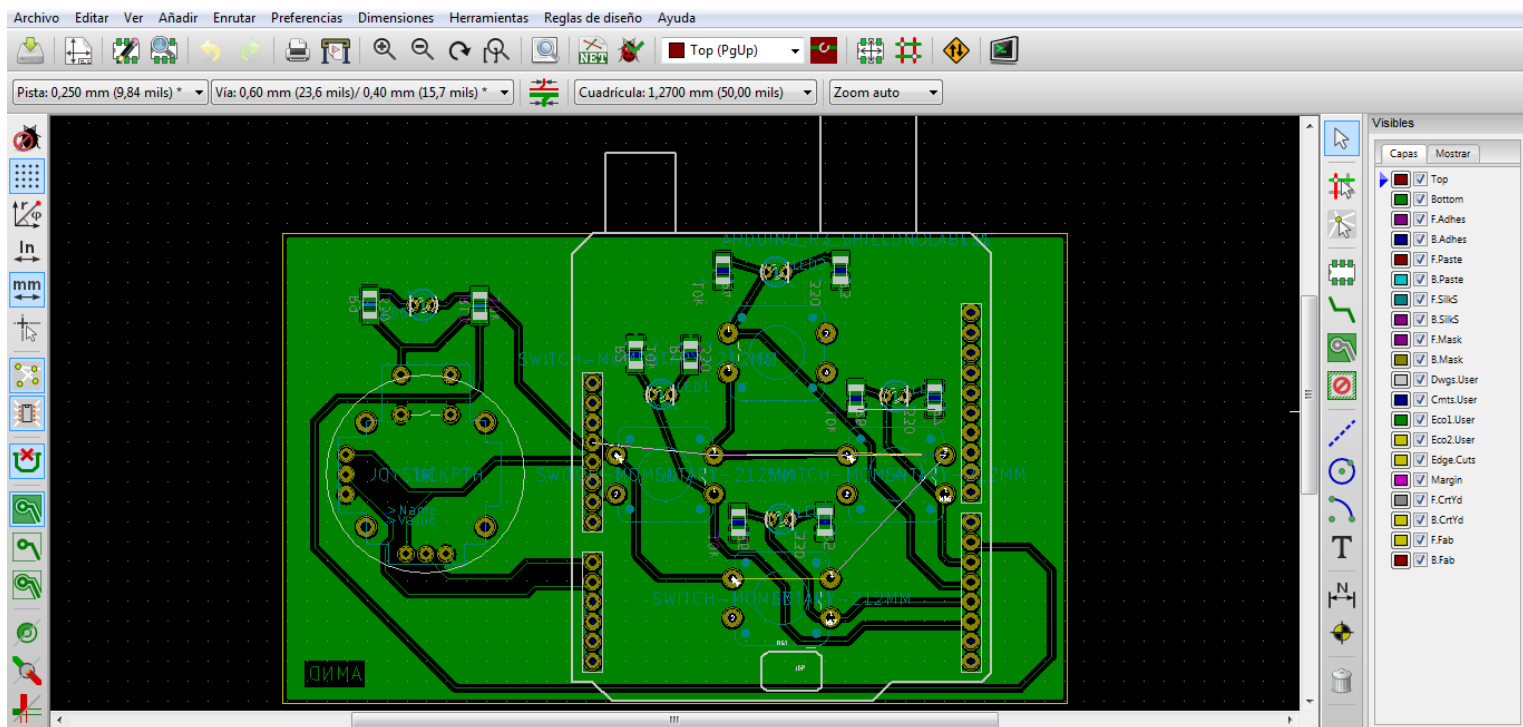
Ahora que se ha realizado en el enrutado de pistas, se comprueba de que no hay fallos, haciendo uso del **DRC (Design Rule Check)**. Si todo está correcto, se procede a la generación de los **GERBERS**, que son archivos que poseen información geométrica sobre las distintas capas de la PCB. Estos archivos son necesarios para fabricar la PCB.

# Diseño en Kicad

Primero se realizó el diseño del esquemático. Hubo que añadir las librerías que se habían utilizado en Eagle para el Joystick y para Arduino.



Una vez realizado el esquemático, se realizó la board y el enrutado de pistas.



# Proceso Naked-Proto de EURO CIRCUITS

PCB Visualizer® v1.4-24-170119 ©2016 - Eurocircuits n.v.

PCB Configurator PCB Checker

Board name joy (B0932326) Data set: Customer data

### DRC - DFM Information

DRC information DFM information

Layer	Required	Measured
<b>Outer layer trackwidth (OL-TW)</b>		
Bottom copper	0.250 mm	0.406 mm
<b>Outer layer isolation distance (OL-TT-PP)</b>		
Bottom copper	0.250 mm	0.406 mm

### Fault view

Outer layer trackwidth (OL-TW) - Bottom copper

Current issue

Trackwidth

More information can be found [here](#).

### Board buildup

Top view

Total material thickness: 1.59 mm

### Bird's Eye View

### Summary

Service: NAKED proto

Estimated shipment date: 01-02-2017

Quantity: 1 PCB

Board surface / Order surface: 0.60 dm² / 0.60 dm²

Prices: Gross\*

Single PCB: € 32.67

Total boards: € 32.67

Express transport: € 4.45

Total: € 37.12

\* The gross prices include 21.00% VAT.

[Save changes](#)

For more 'Advanced options', switch to our standard PCB calculator offering 'PCB proto' and 'STANDARD pool' service options.

[Advanced options](#)

### Alternatives

Customized matrix

1 PCB	2 PCBs	5 PCBs
7 working days	7 working days	7 working days
Gross*	Gross*	Gross*
€ 32.67	€ 18.74	€ 9.35

PCB Visualizer® v1.4-24-170119 ©2016 - Eurocircuits n.v.

PCB Configurator PCB Checker

Board name joy (B0932326) Data set: Customer data

### DRC - DFM Information

DRC information DFM information

Layer	Values
<b>Plating</b>	
Bottom copper	0.96
<b>Solderpaste surface</b>	
Bottom solderpaste	57.72 mm²
<b>Not-connected soldermask-free pads - Potential fiducials</b>	
Bottom copper	0
<b>Copper free of soldermask</b>	
Bottom copper	81.02%
<b>Copper surface</b>	
Top copper	0.0000 dm²
Bottom copper	0.4861 dm²

### Fault view

Plating - Bottom copper

Current issue

Plating index : 0.96

The plating index measures the uniformity of copper density on the board. A completely

### Plating

Bottom plating index 0.96



### Summary

Service: NAKED proto

Estimated shipment date: 01-02-2017

Quantity: 1 PCB

Board surface / Order surface: 0.60 dm² / 0.60 dm²

Prices: Gross\*

Single PCB: € 32.67

Total boards: € 32.67

Express transport: € 4.45

Total: € 37.12

\* The gross prices include 21.00% VAT.

[Save changes](#)

For more 'Advanced options', switch to our standard PCB calculator offering 'PCB proto' and 'STANDARD pool' service options.

[Advanced options](#)

### Alternatives

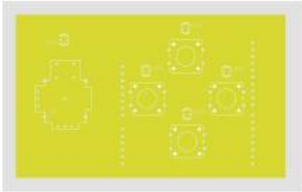
Customized matrix

1 PCB	2 PCBs	5 PCBs
7 working days	7 working days	7 working days
Gross*	Gross*	Gross*
€ 32.67	€ 18.74	€ 9.35

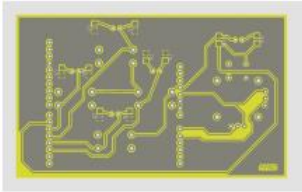
PCB Visualizer

PCB images

Top view :

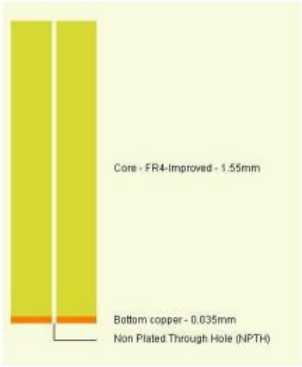


Bottom view:



Buildup & Mechanical plan

Board buildup :  
:

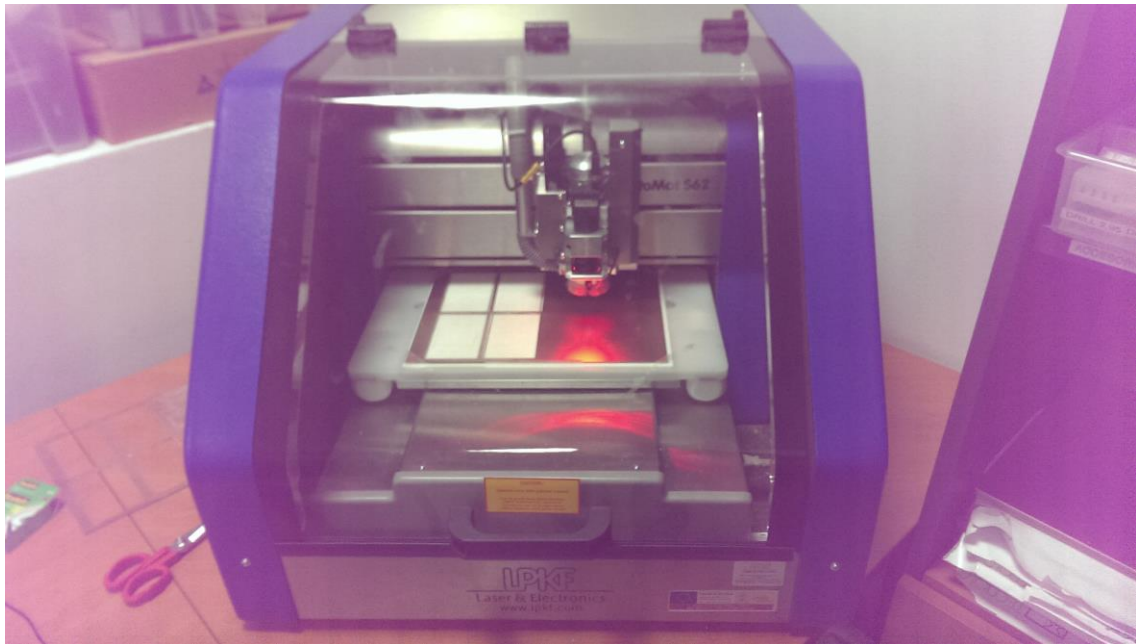




## *Fase de fabricación*

---

La fabricación de la PCB se va a realizar mediante la técnica de fresado. Consiste en una máquina tipo plotter que hace un “dibujo” sobre la placa empleando fresas que eliminan el cobre de la misma. La máquina que se ha utilizado es una **ProtoMat S62**.



Una vez fabricada, se procede a comprobar mediante un polímetro si existen cortos, y de manera visual, si existen pistas rotas.



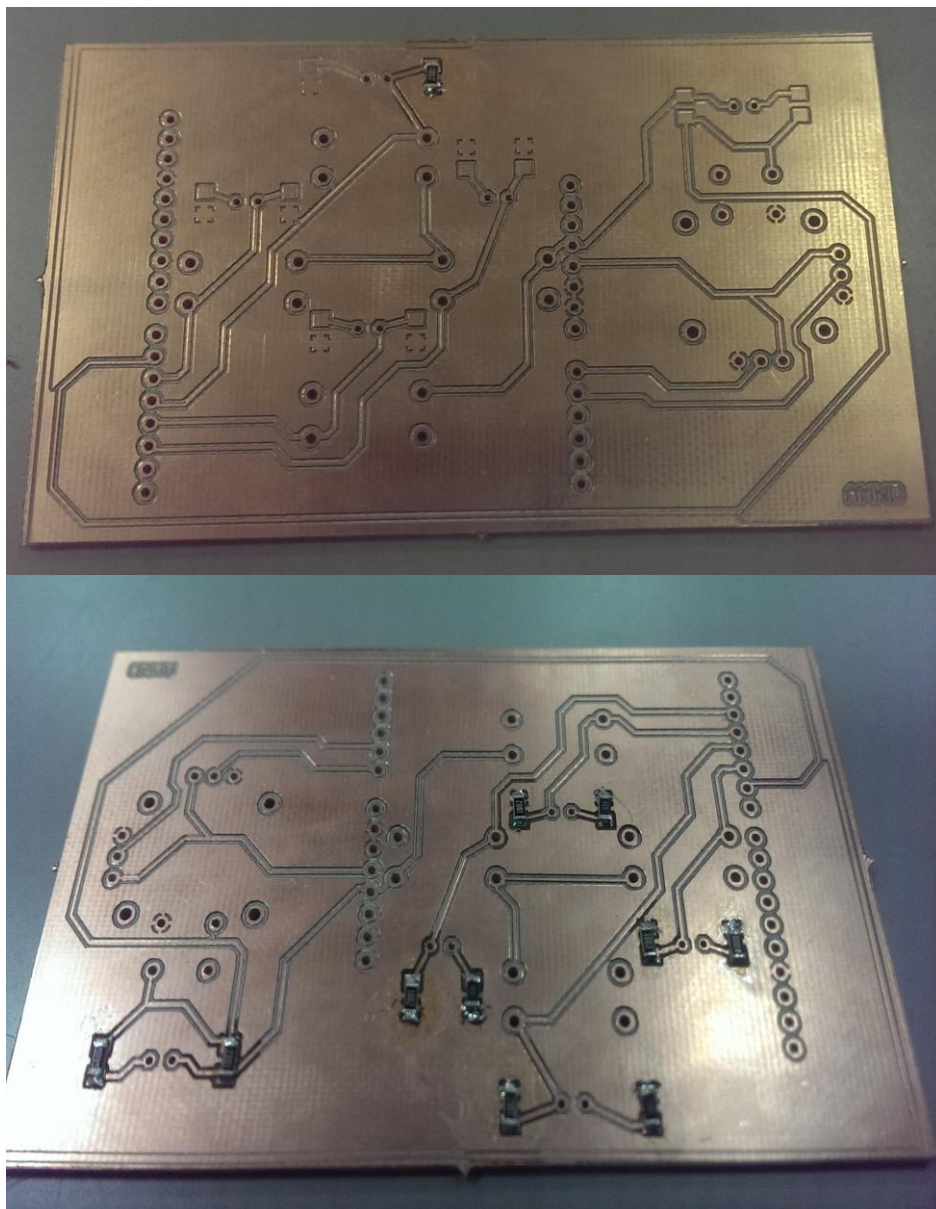
## *Fase de ensamblaje*

---

En la fase de ensamblaje, se procede mediante la técnica de soldadura a fijar los componentes a la PCB. Las resistencias que se emplean en este proyecto son de montaje superficial, mientras que todos los demás componentes son **Through Hole**.

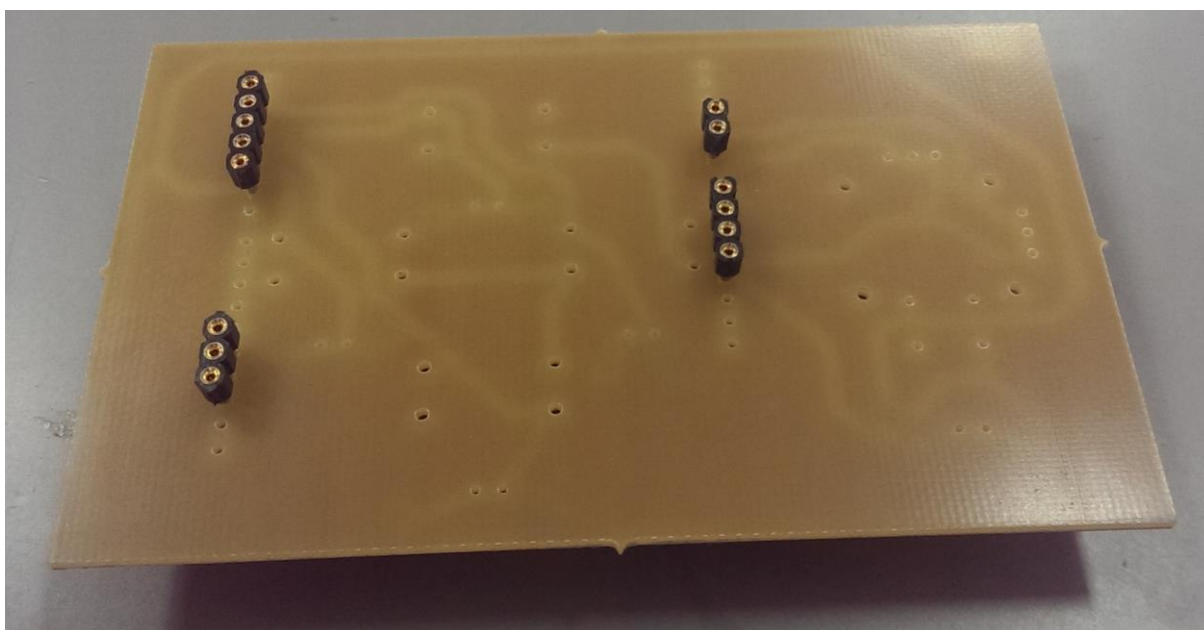
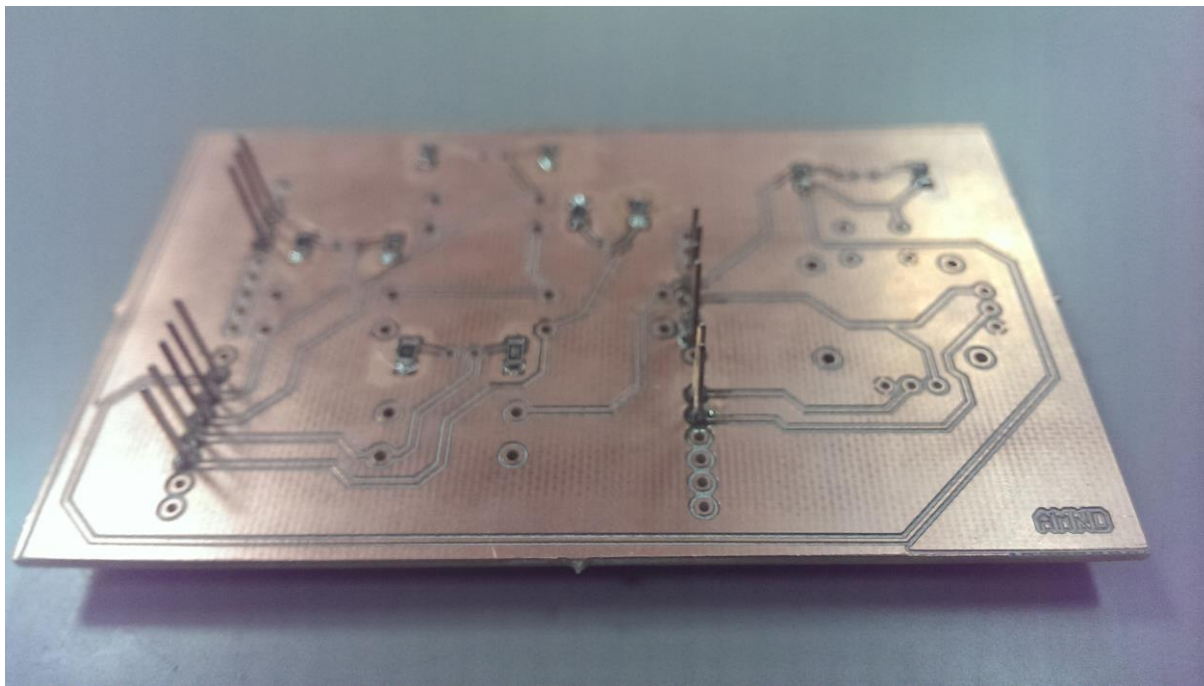
Para soldar, se ha utilizado una estación de soldadura, que contiene un soldador y una consola de control de temperatura. En el soldador se ha utilizado una punta fina, ya que los componentes que se han soldados tienen espacios pequeños entre pads. También se ha utilizado flux cada vez que se ha soldado un componente.

Primero se comenzó soldando las resistencias:

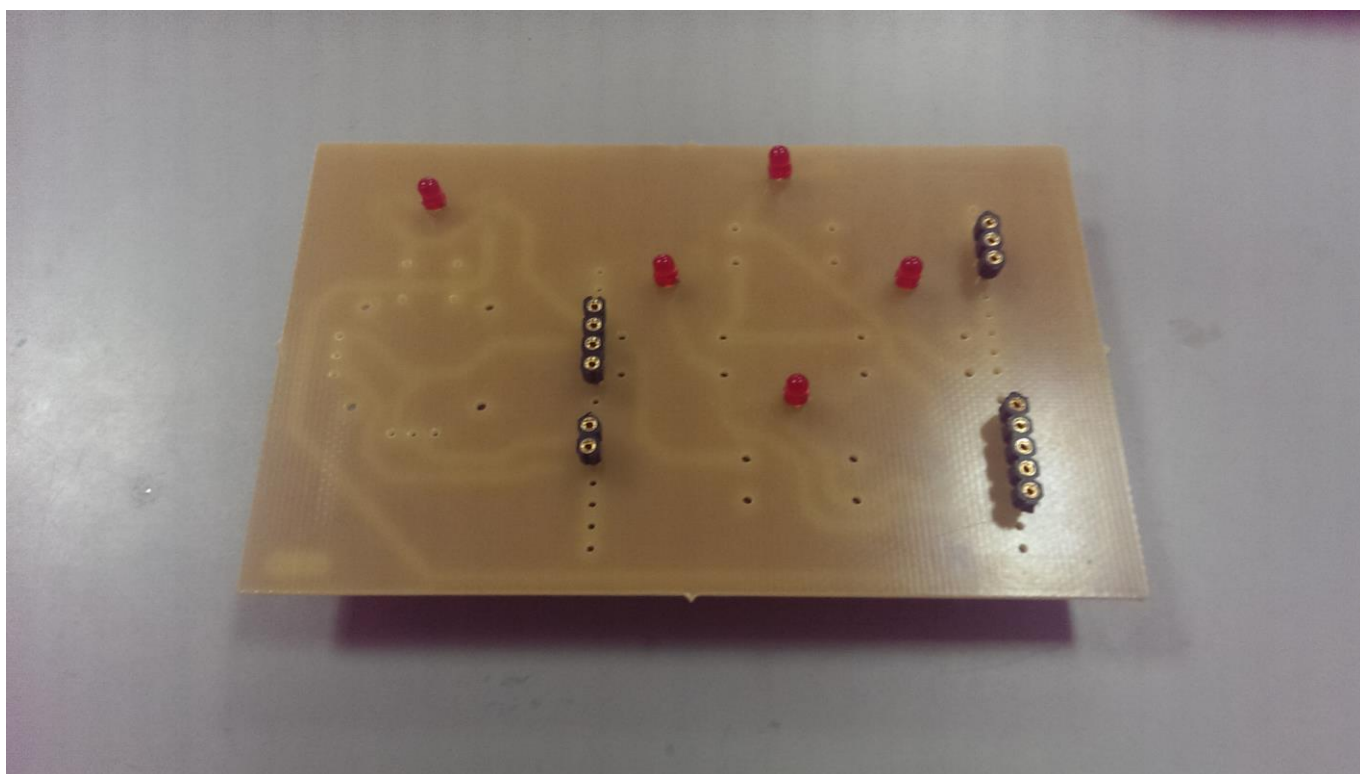
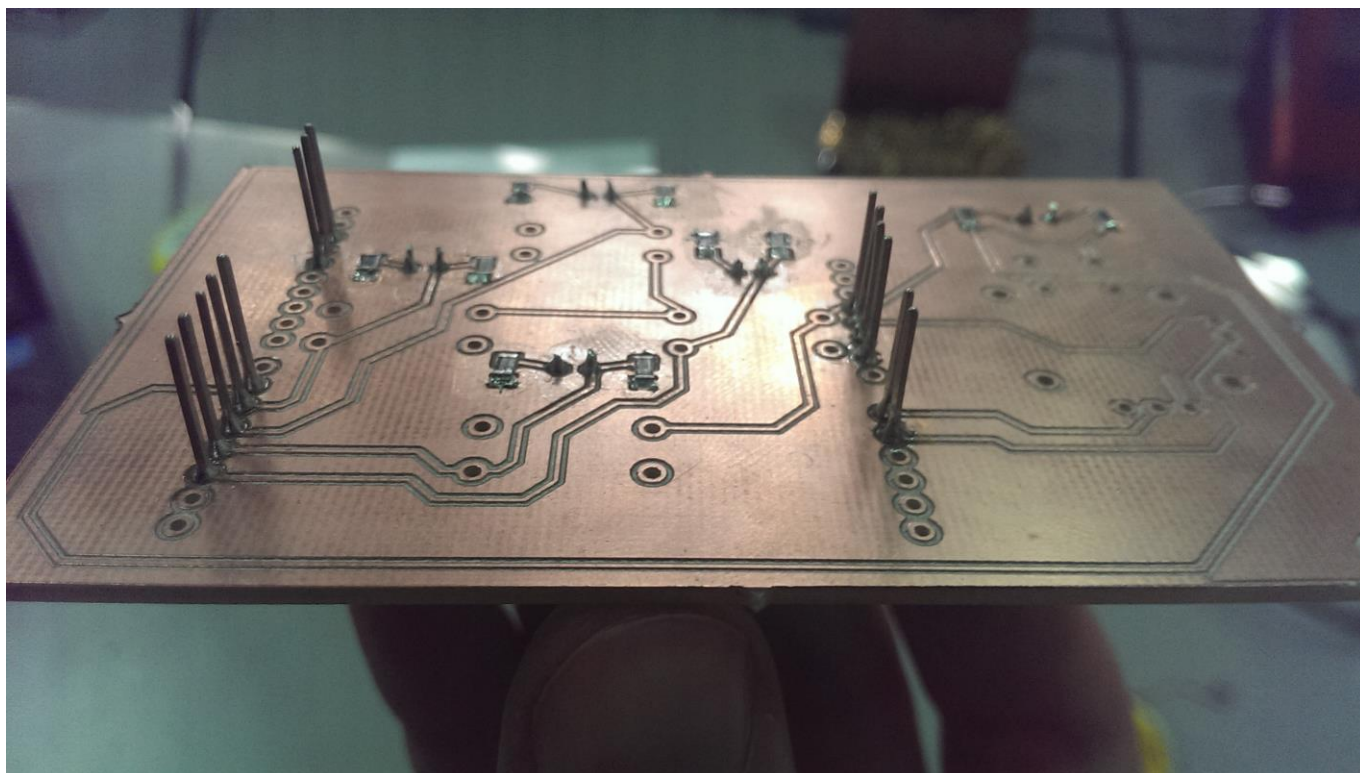




A continuación, los pines de expansión:

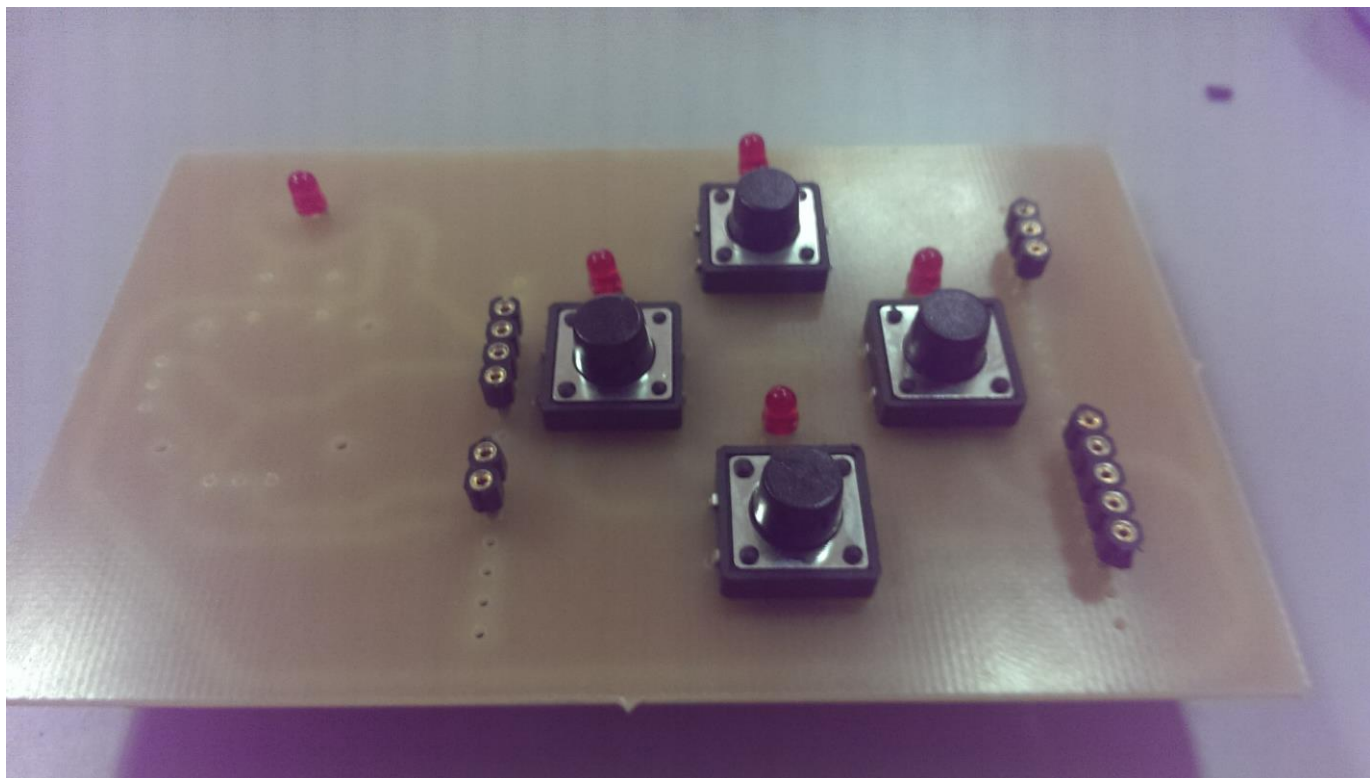
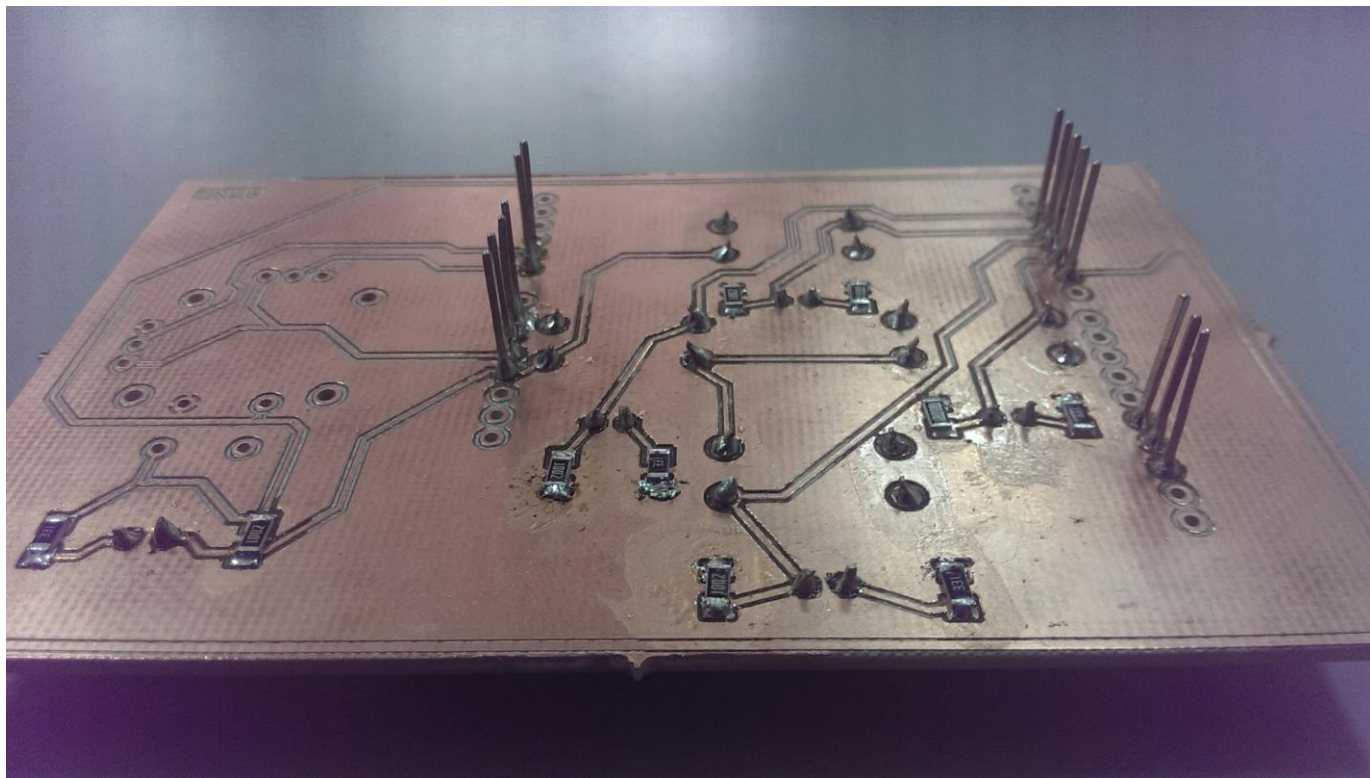


Los LEDs:

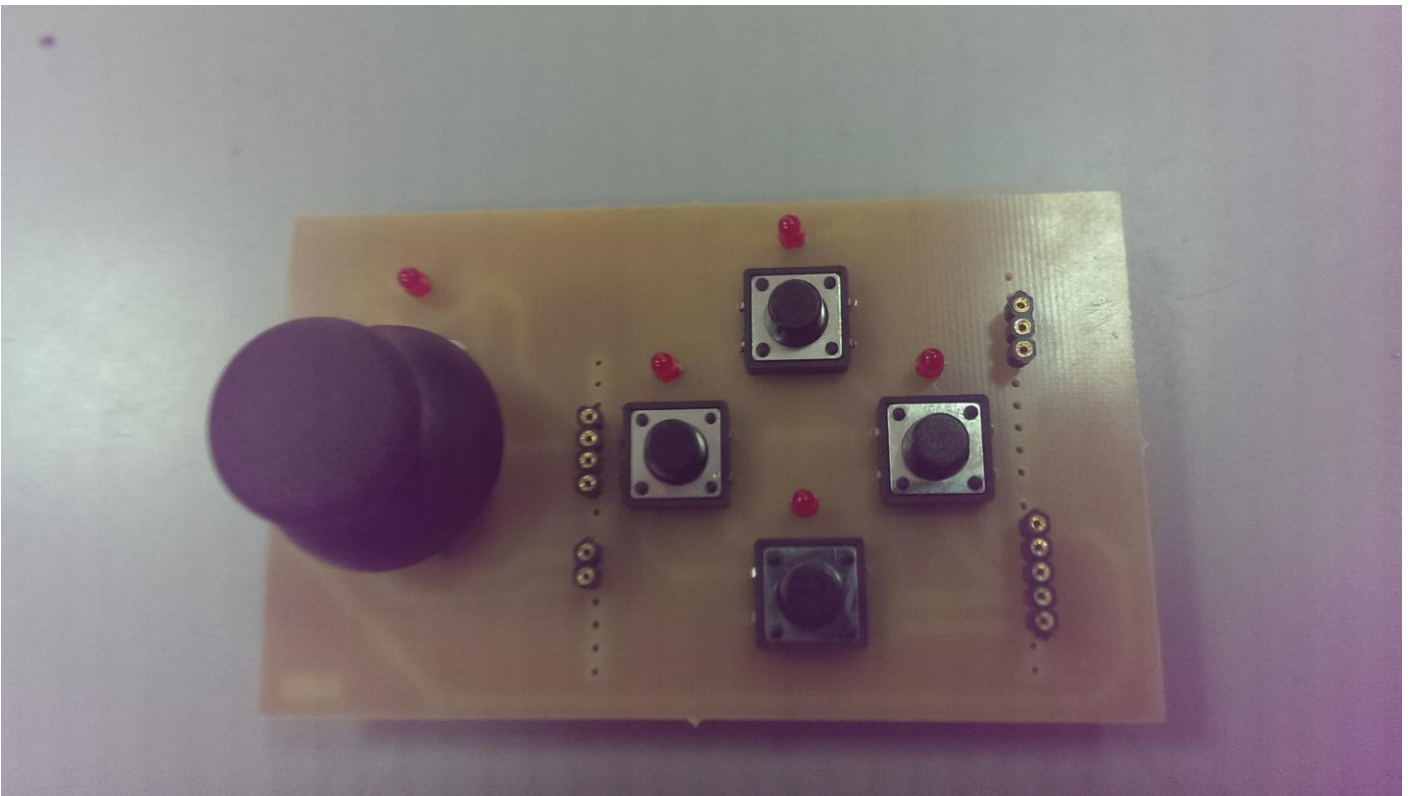
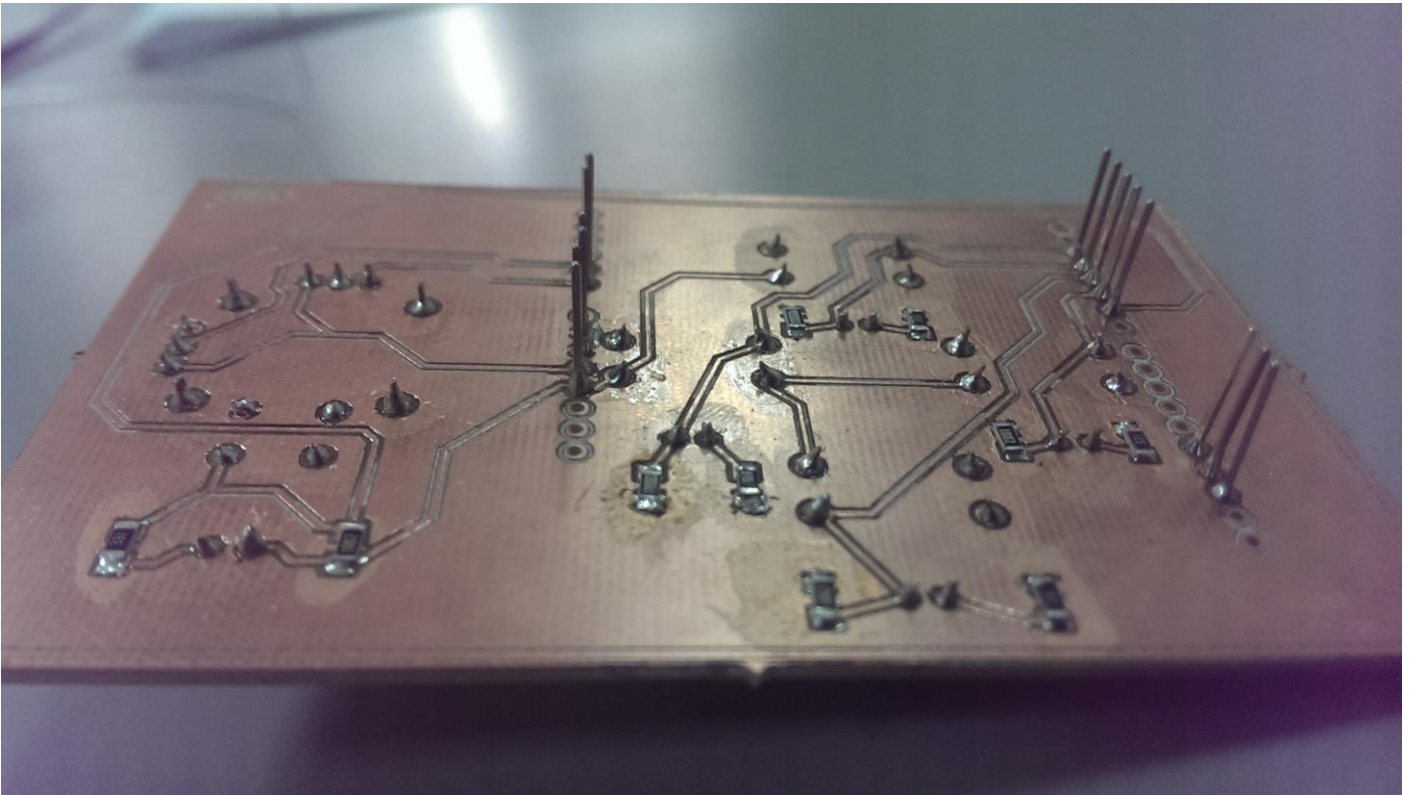




Los botones:



Y por último, el joystick:



## *Fase de Test y desarrollo de Software*

---

Cuando se construyó la PCB, se realizó un testeo de la PCB, en la que de manera visual, y con un polímetro, se buscaban pistas rotas o cortocircuitos.

Ahora que ya se tienen los componentes, se necesita hacer un test para comprobar el correcto funcionamiento de la placa.



Partiendo de un proyecto dado en Arduino, se modificó unas líneas de código, para asignar los pines correctamente a los botones, y en la lectura analógica, modificar la dirección a la que se movía el joystick, ya que estaba invertida:

```
//Asocial los pines digitales a los botones del mando
controllerData.triangleOn = !digitalRead(4);
controllerData.circleOn = !digitalRead(5);
controllerData.squareOn = !digitalRead(2);
controllerData.crossOn = !digitalRead(3);
controllerData.selectOn = !digitalRead(6);
controllerData.startOn = !digitalRead(6);

controllerData.leftStickX = map( (analogRead(A0)) , 0, 1023, 255, 0);
controllerData.leftStickY = map( (analogRead(A1)) , 0, 1023, 255, 0);
```



Estos son los archivos que se han utilizado para el Software en Arduino

Nombre	Fecha de modifica...	Tipo	Tamaño
 joy.ino	20/01/2017 11:58	Arduino file	2 KB
 UnoJoy.h	18/01/2017 12:57	C/C++ Header	10 KB

El código completo es el siguiente:

```
joy$ UnoJoy.h
#include "UnoJoy.h"

void setup() {
    setupPins();
    setupUnoJoy();
}

void loop() {
    dataForController_t controllerData = getControllerData();
    setControllerData(controllerData);
}

void setupPins(void) {
    for (int i = 2; i <= 12; i++) {
        pinMode(i, INPUT);
        digitalWrite(i, HIGH);
    }
}

dataForController_t getControllerData(void) {

    dataForController_t controllerData = getBlankDataForController();

    //Asocia los pines digitales a los botones del mando
    controllerData.triangleOn = !digitalRead(4);
    controllerData.circleOn = !digitalRead(5);
    controllerData.squareOn = !digitalRead(2);
    controllerData.crossOn = !digitalRead(3);
    controllerData.selectOn = !digitalRead(6);
    controllerData.startOn = !digitalRead(6);

    controllerData.leftStickX = map((analogRead(A0)), 0, 1023, 255, 0);
    controllerData.leftStickY = map((analogRead(A1)), 0, 1023, 255, 0);

    return controllerData;
}
```

Se compiló el código y se subió a la placa Arduino. Entonces, se realizó una primera comprobación del funcionamiento de los botones y del joystick con la aplicación **UnoJoyProcessingVisualizer**:

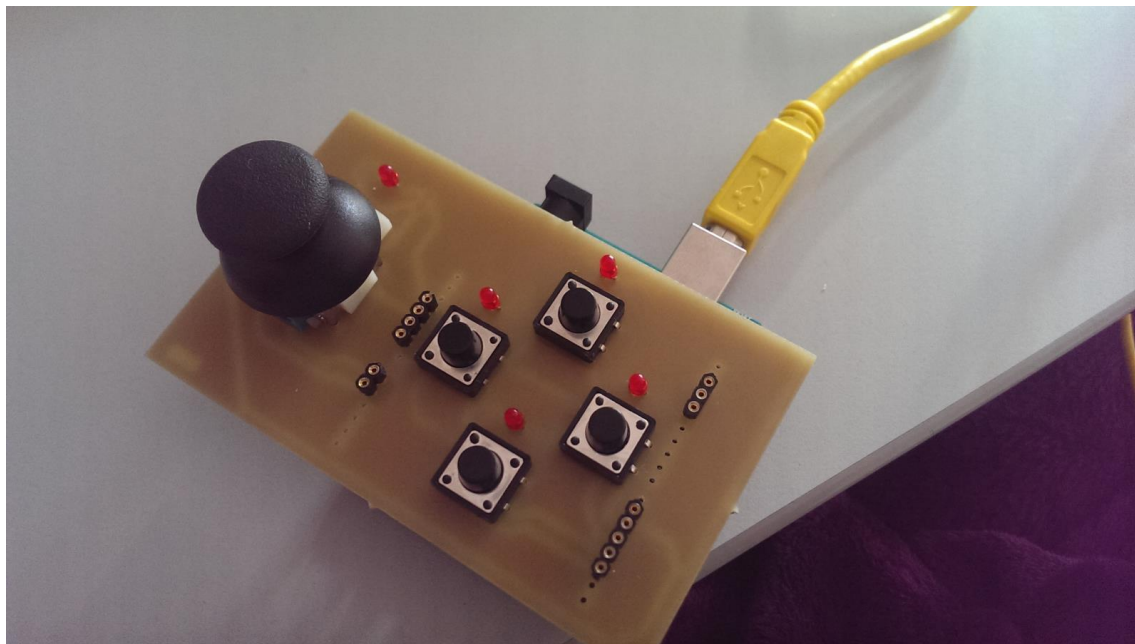


Una vez realizada esta comprobación, se procedió a poner la placa Arduino en modo **DFU (Device Firmware Upgrade)**, provocando un cortocircuito entre dos pines específicos de la placa. Ahora que se está en este modo, se tienen varias opciones:

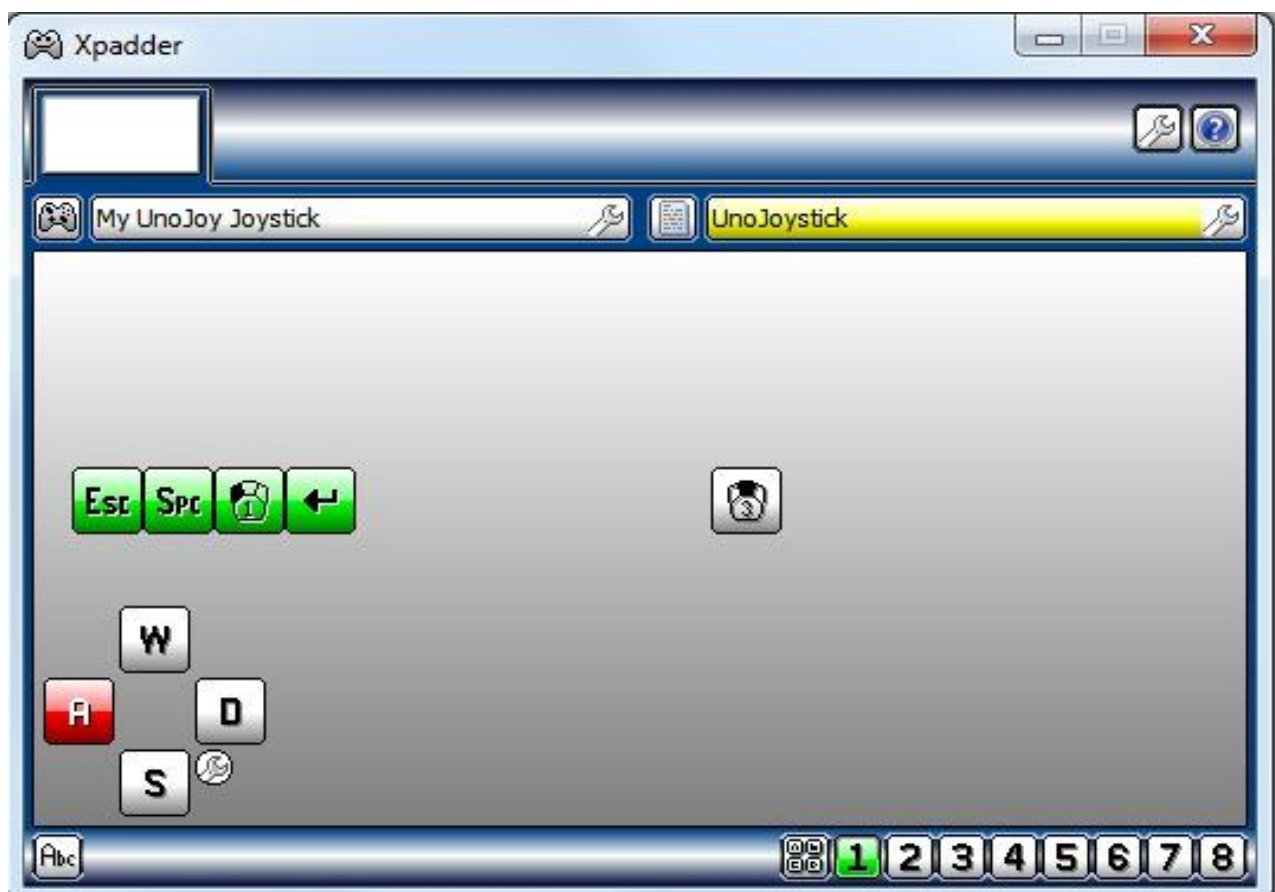
1.- Realizarlo manualmente, utilizando el programa **FLIP** de **Atmel**, mediante el cual se cargan los archivos hexadecimales que convertirán la placa Arduino en un Joystick.

2.- Una manera más automatizada. Se seleccionan un archivo .bat, que instalará unos drivers. Después, se vuelve a seleccionar otro archivo .bat, que convertirá la placa en un Joystick.

Una vez la placa Arduino sea reconocida como un Joystick, es hora de conectar la PCB a la placa:



Se conecta la placa al PC, y mediante el programa Xpadder, se configura la función de cada botón y de cada dirección del Joystick:



En un principio se configuró para que el Joystick funcionara como cámara (girar el personaje), pero al ser la sensibilidad tan alta, era injugable, por lo que se decidió asignar a los controles del joystick las teclas **W, S, A, D**. El botón de la PCB que se corresponde con la **X** en la **PlayStation**, se configuró como la tecla **Espacio**, que se suele utilizar para saltar. El botón **círculo**, se configuró con el **botón izquierdo del ratón**, que es el disparo. El botón **triángulo**, como **cambiar de arma**, y el botón **cuadrado** como **Escape**, para **salir al menú principal**.

Una vez configurados los botones, es hora de probar el mando en un juego, como se puede ver en el siguiente vídeo clicando [aquí](#).