

LDH

ZPUINO

Índice

Objetivos	1
Introducción	1
Desarrollo de la Practica	1
Instalación IDE	1
1º ejercicio: Diseño circuito básico sobre la FPGA: Inversor.	2
2º ejercicio: Carga del SoC Zpuino y desarrollar Sketches: ON/OFF led desde pc	3
3º ejercicio: Convertir la placa Papilio en un analizador lógico	5
4º ejercicio: Añadir periférico al Soc Zpuino: AURT.	6

Plataforma Papilio

Objetivos:

Los objetivos de estas prácticas son conocer la plataforma Papilio e instalar y trabajar con su IDE de desarrollo “DESING LAB”. Cargar el SoC Zpuino en la Papilio. Realizar Diversas tareas para afianzar los conocimientos.

Uso de [la Papilio One SPARTAN3E 500](#)

Introducción:

La plataforma Papilio son placas de desarrollo de hardware abiertas basadas en FPGAs de Xilinx. Por medio de estas FPGAs se quiere la facilidad de diseño de circuito simplificando las bibliotecas de circuitos de las FPGAs, arrastrando y soltando los circuitos que se quieran llegar a implementar en Papilio; haciendo que los proyectos en FPGAs sean más creativos y diversos sin la necesidad de aprender VHDL o Verilog. En su [web](#) encontraremos una clara explicación del proyecto de las placas Papilio y su diversidad.

El IDE de desarrollo “DESING LAB” está basado en el de Arduino. Más concretamente cogieron el IDE de Arduino y lo sobrecargaron añadiendo circuitos a la mezcla, para así poder dibujar y depurar tus propios circuitos en un solo lugar de una manera fácil. Así cada boceto tiene un circuito asociado.

El SoC Zpuino que implementaremos en la Papilio One. Es un SoC implementado en VHDL (soft core) basado en el microprocesador ZPU de Xilinx. Las especificaciones y diseño de Zpuino son abiertas pudiendo añadir periféricos al SoC según necesidades. Se puede trabajar con Zpuino de manera similar a Arduino debido a las cualidades del IDE ya mencionadas anteriormente y como mencionan en la [web](#) de la plataforma Papilio.

Desarrollo de la Práctica:

Instalación del IDE:

Nos descargamos el IDE del servidor del profesor por rapidez: <http://10.1.15.78/~bellido>

Seguimos los pasos de esta [GUIA](#). Que en definitiva es descomprimir el fichero bajado y en el directorio que se encuentra el archivo con el terminal de Ubuntu ejecutamos:

```
sudo ./Ubuntu-setup.hs
```

1- Diseño circuito básico sobre la FPGA: Inversor.

Desde Desing Lab vamos a diseñar nuestro circuito inversor de una manera fácil y rápida. Para ello nos vamos a ayudar de un tutorial que nos indica paso a paso lo que debemos hacer. [TUTORIAL](#).

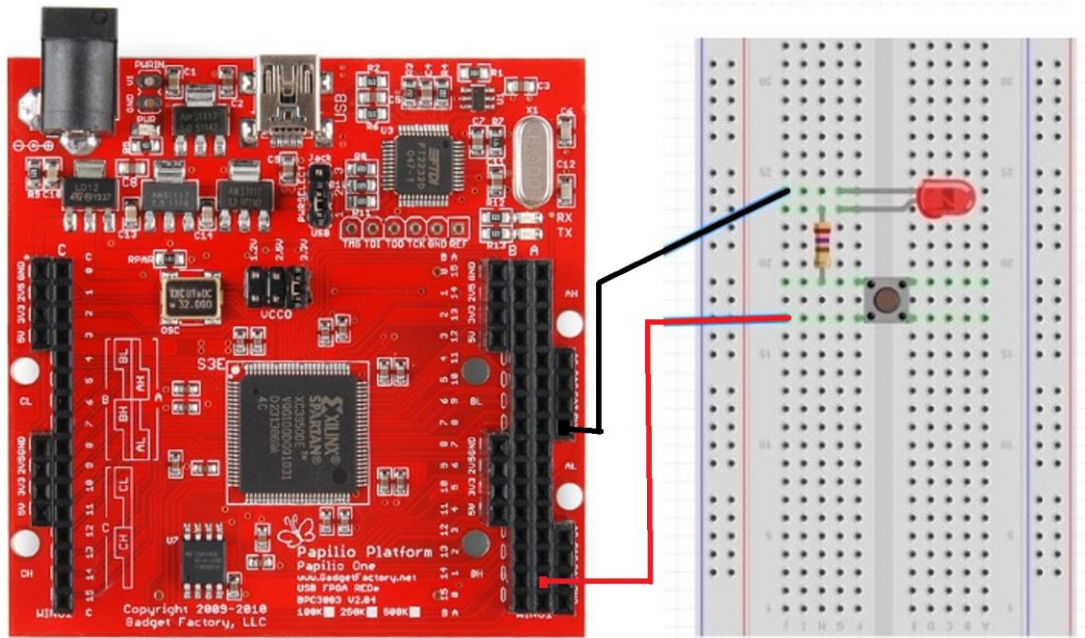
Comprobaremos el funcionamiento del inversor mediante un interruptor y un led. Conectando el interruptor a la patilla A0 de la placa Papilio.

Después de hacer el circuito y generar los ficheros de programa para usar en la FPGA en el propio IDE. Hay que corregir un error que se produce en el nombre a la hora de generar los ficheros, que consiste en renombrar uno de ellos.

- **Carpeta:** <proyecto>/circuit/500K/
- 1.- Borrar **papilio_one_500k.bit**
- 2.- Renombrar **Papilio_One_500K.bit** a **papilio_one_500k.bit**

De esta manera ya se puede comprobar el funcionamiento del inversor creado.

El paso de la errata hay que realizarlo cada vez que se crea o cambia un circuito y se generan los ficheros para poder usar en la FPGA de la Papilio One.



2- Carga del SoC Zpuino y desarrollar Sketches: ON/OFF led desde pc

En este ejercicio consiste en cargar en la Papilio One el SoC de Zpuino, y hacer que por medio del puerto serie podamos encender y apagar un led desde el pc. El encendido y apagado del led el pc se hará por medio de la ejecución de un código Python. Este ejercicio se parece mucho a aquel que hicimos con Arduino de la mismatematica, tanto que el código Python de entonces se puede reutilizar ahora.

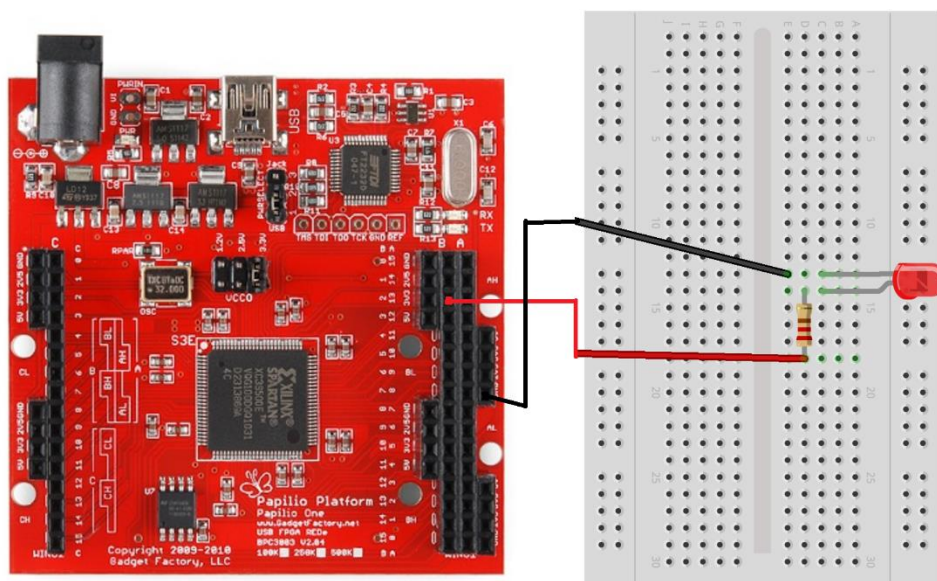
Para explicar la carga de SoC y los pasos a seguir lo haremos por medio de tutorial que nos llevara de la mano: [TUTORIAL](#)

Código Python

```
import serial
Zpuino = serial.Serial('/dev/ttyUSB0', 9600);
print ("Starting o Inicializando")
while 1:
    comando = raw_input('Introduce un comando H o L: ')
    Zpuino.write (comando)
    if comando == 'H':
        print('LED ENCENDIDO')
    elif comando == 'L':
        print('LED APAGADO')
Zpuino.close()
```

Código Zpuino

```
int led = 13;  
  
void setup(){  
    pinMode (led, OUTPUT);  
    Serial.begin(9600);  
}  
  
void loop(){  
    if(Serial.available()){  
        char c = Serial.read();  
        if (c == 'H'){  
            digitalWrite(led, HIGH);  
        }else if( c == 'L'){  
            digitalWrite(led, LOW);  
        }  
    }  
}
```



3- Convertir la placa Papilio en un analizador logico

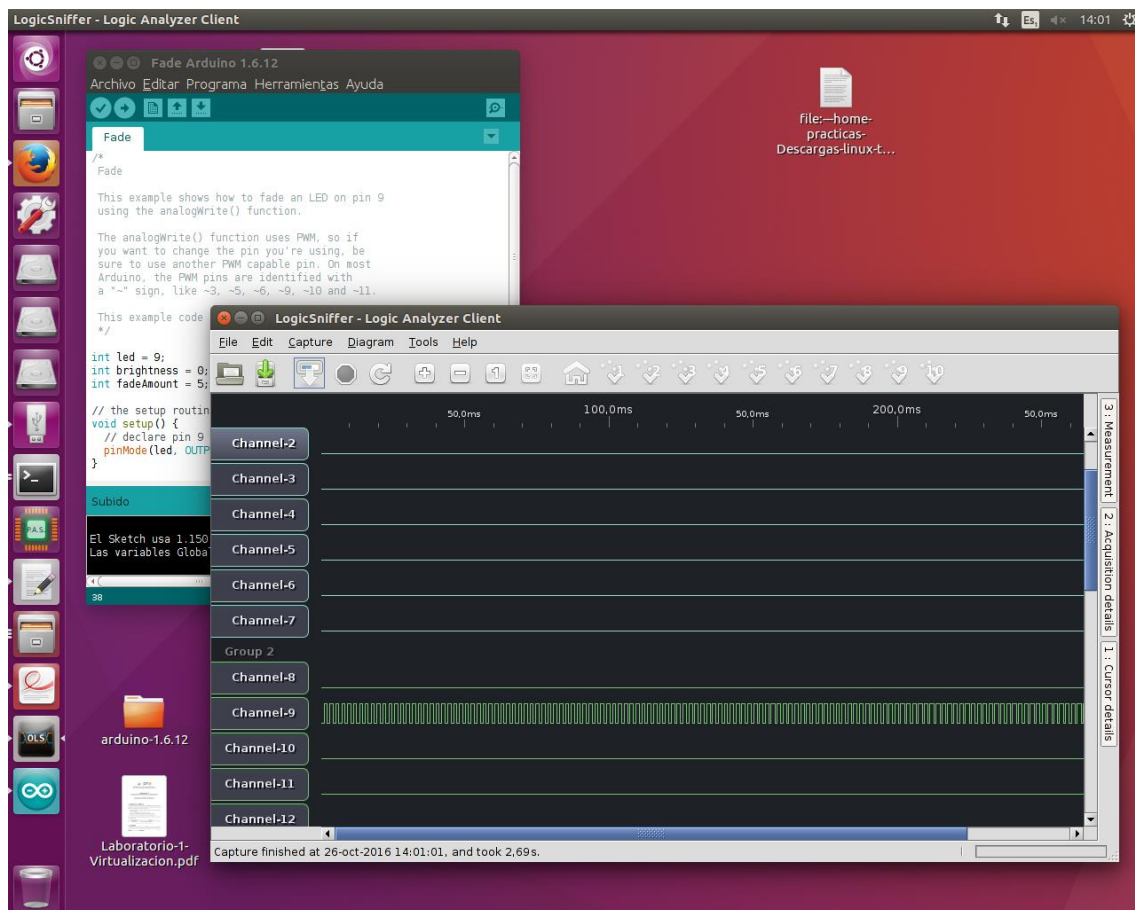
Lo primero que tenemos que hacer es coger una placa Arduino que desde donde vamos a mandar la señal que la Papilio One va analizar como analizador lógico.

Configuramos Arduino para mandar una señal PWM, lo que hay que hacer es cargar el ejemplo fade que viene en el IDE de Arduino. Y una vez cargado conectar a la Papilio como analizador lógico.

Para transformar nuestra Papilio en un analizador lógico nos vamos a servir de una guía. [GUÍA](#)

Hubo un problema con los PC's del laboratorio. El cliente software del analizador lógico tiene conflicto con Java. Y para solucionarlo lo hacemos mediante un [Script](#). Para Ejecutar el script se descomprime y en terminal ponemos `./run.sh`

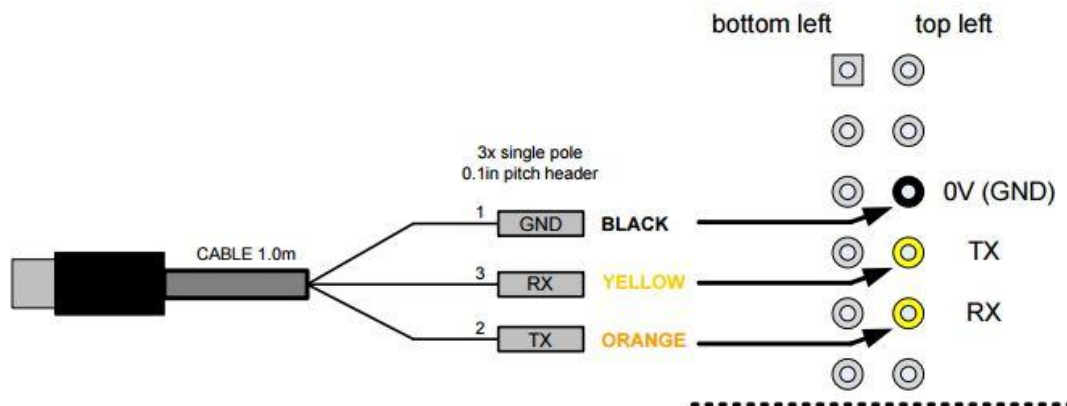
Una vez tenemos el analizador lógico corriendo le conectamos la señal PWM para que se visione.



4- Añadir periférico al SoC Zpuino: UART

A través de la nueva UART (puerto Serie) vamos a manejar el encendido/apagado de un Led desde el PC. Como se ve es muy parecido al ejercicio 2 de Zpuino sólo que ahora daremos las instrucciones por medio del nuevo puerto serie.

Para añadir el nueva UART nos guiaremos de este [TUTORIAL](#). Para hacer la conexión con la nueva UART la haremos por medio del Clable TTL-232R Rpi de la manera que se muestra en la imagen, sabiendo cuales son las patillas RX y TX de nuestra nueva UART en los pines de la Papilio One.



El Código Python que hace falta para mandar ordenes desde el PC es el mismo que el del ejercicio 2 asegurándose de que el puerto serie es el adecuado.

Código Zpuino

```
HardwareSerial mySerial1 (Wishbone (5));  
  
int led = 13;  
  
void setuo(){  
    pinMode (led, OUTPUT);  
    mySerial1.begin(9600);  
}  
  
void loop(){  
    if(mySerial1.available()){  
        char c = mySerial1.read();  
        if (c == 'H'){  
            digitalWrite(led, HIGH);  
        }else if( c == 'L'){  
            digitalWrite(led, LOW);  
        }  
    }  
}
```

Montaje del Circuito:

