Miguel Ángel Lara Guarino

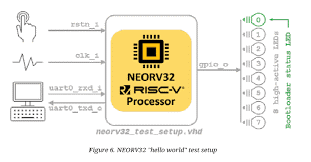
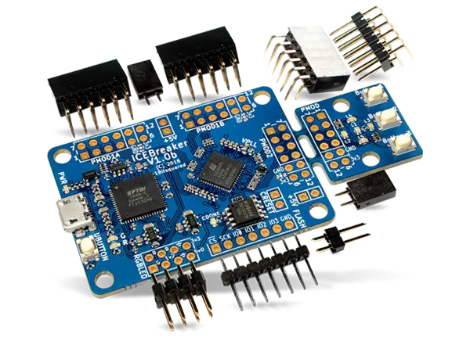
Alfonso Manchado Pérez

**Icono

Descripción generada automáticamente**

**SEPA 2023:**

**Memoria de Prácticas y Proyecto**



SoftCore

En esta memoria, nos adentramos en el emocionante mundo de la FPGA Lattice IceBreaker, una placa de desarrollo poderosa y compacta, y exploramos en detalle su integración con el procesador Nerov32 como SoftCore. Este procesador de 32 bits, diseñado específicamente para su implementación en FPGA, proporciona una sólida base de cómputo y control, ampliando las capacidades de la IceBreaker y ofreciendo una plataforma adaptable para numerosas aplicaciones.

A lo largo de este informe, comentaremos las diferentes prácticas realizadas durante la asignatura y acabando con la explicación de nuestro proyecto final, donde la combinación de la FPGA IceBreaker y el procesador Nerov32 ha demostrado ser excepcionalmente efectiva.

Pero, antes de nada, ¿por qué el uso de un procesador dentro de una FPGA? La inclusión de un procesador en una FPGA (Field-Programmable Gate Array) ofrece una versatilidad única al permitir la personalización del hardware para aplicaciones específicas, combinando la lógica personalizada con el procesamiento de propósito general. Es decir, tenemos por un lado el poder de paralelizar múltiples tareas con la FPGA y por el otro, el desarrollo secuencial por parte del procesador que nos permite tener más control sobre el sistema.

**Índice**

[1. Práctica 1: Toma de contacto y uso de los leds 3](#_Toc153653511)

[**Hardware** 3](#_Toc153653512)

[**Software** 3](#_Toc153653513)

[2. Práctica 2: Uso del Teclado numérico mediante el GPIO 4](#_Toc153653514)

[**Hardware** 4](#_Toc153653515)

[**Software** 4](#_Toc153653516)

[3. Práctica 3: Uso de un periférico Wishbone 4](#_Toc153653517)

[**Hardware** 4](#_Toc153653518)

[**Software** 4](#_Toc153653519)

[4. Proyecto Final 4](#_Toc153653520)

[**Hardware** 4](#_Toc153653521)

[**Software** 4](#_Toc153653522)

# Práctica 1: Toma de contacto y uso de los leds

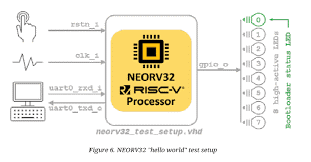
Para esta primera práctica, entraremos en contacto con el manejo de las herramientas que se nos proporcionan para realizar la síntesis, implementación y generación del bitstream por parte de los códigos VHDL y la compilación para la parte del procesador Neorv32.

Los programas que nos permitirán efectuar estos pasos son:

* Síntesis 🡪 Yosys + Ghdlsynth
* Place & Route 🡪 Nextpnr
* Bitstream 🡪 Icestorm
* Build 🡪 GNU Make

## **Hardware**

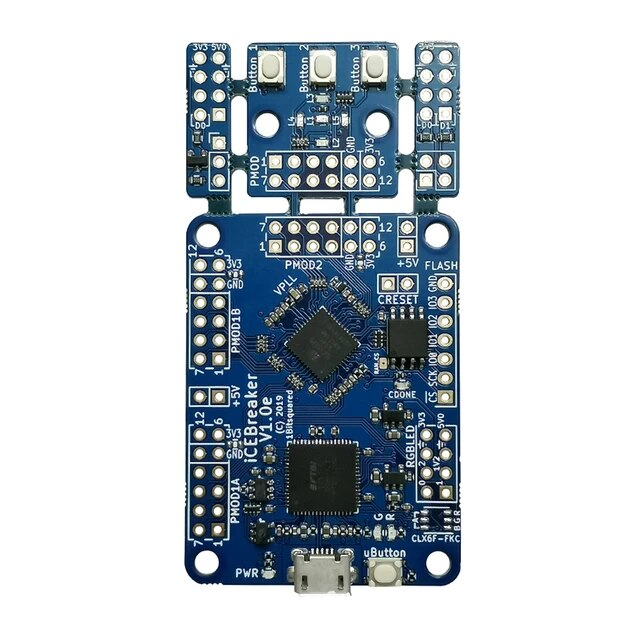
Para la parte del hardware, se hará uso de los pines de comunicación GPIO para crear una conexión directa entre los Leds y el soft processor Neorv32. Además, se agregará una señal de reset controlada por el soft processor y, finalmente, se comunicará también mediante el GPIO, el valor del botón pulsado.



Gpio\_o

Gpio\_i

Leds (4 downto 0)



Reset signal

**ICE40UP5k**

**FPGA**

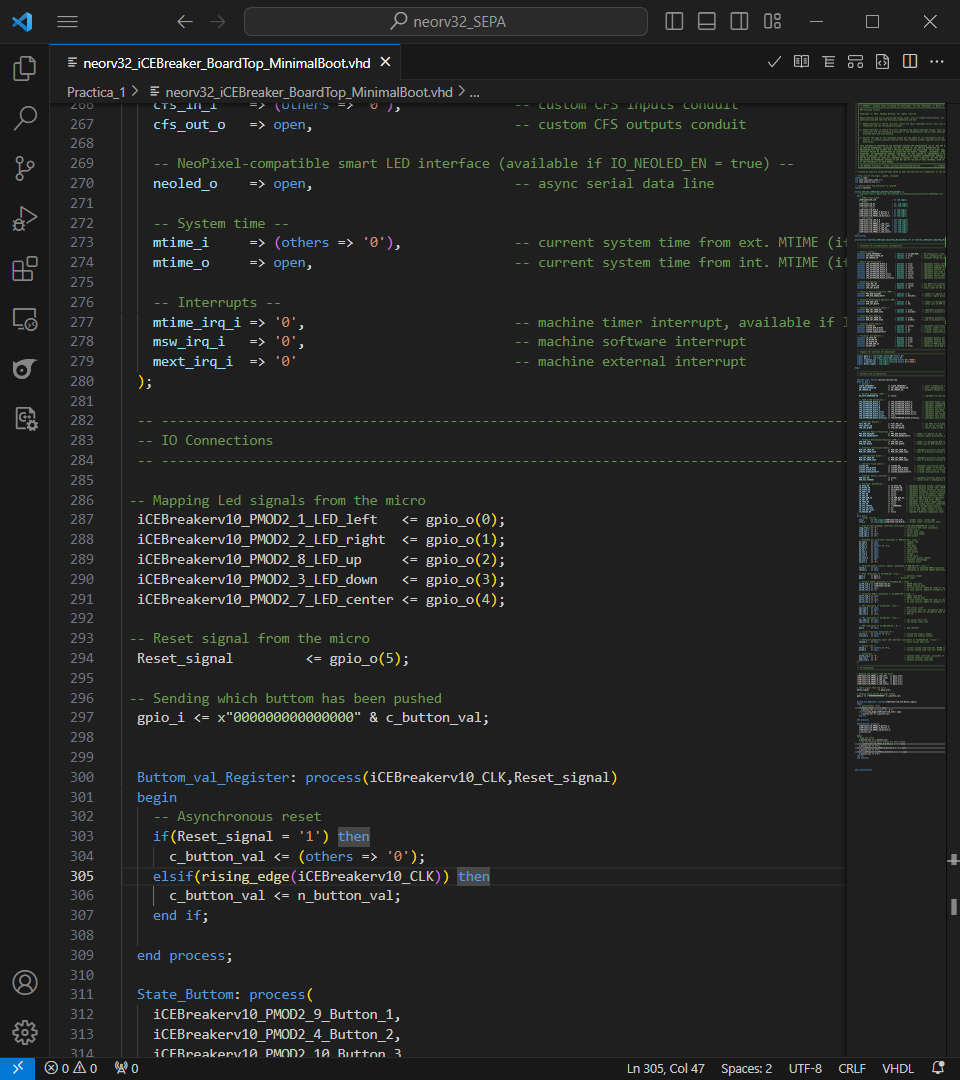
Button (2 downto 0)

1. Texto

   Descripción generada automáticamenteAñadimos las señales necesarias:
2. Asignamos a la señal ‘c\_button\_val’ el valor del botón que haya sido pulsado:

Texto

Descripción generada automáticamente

1. Asignamos valor a las salidas:

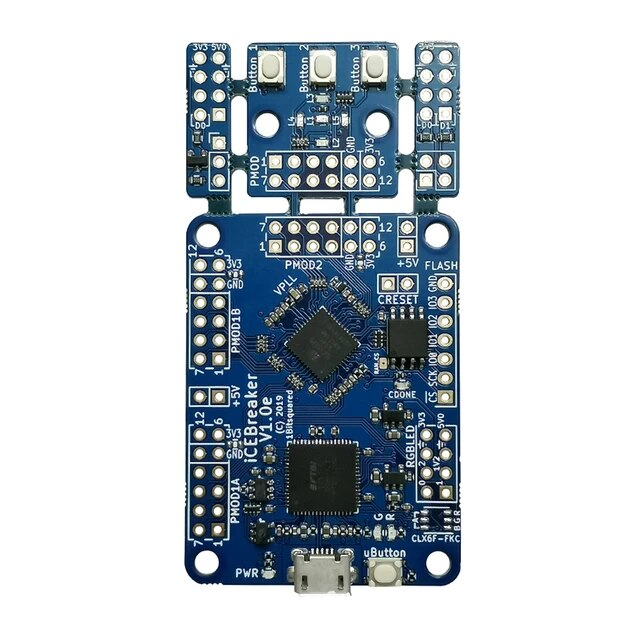
## **Software**

# Práctica 2: Uso del Teclado numérico mediante el GPIO

((Alfon explica la intro))

## **Hardware**

Para la parte del hardware, se hará uso de los pines de comunicación GPIO para crear una conexión directa entre el teclado y el soft processor Neorv32. Además, se instanciará una entidad llamada “peripheral\_teclado.vhd” la cual realizará el proceso de lectura del teclado.



s\_leds [4:0]

**ICE40UP5k**

**FPGA**

Diagrama

Descripción generada automáticamente

Gpio\_o

Gpio\_i

s\_reset

Un control color blanco

Descripción generada automáticamente

c\_button\_val [2:0]

Peripheral\_teclado.vhd

Key\_value [2:0]

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza bajaPara realizar la lectura del del teclado en hardware, se ha diseñado la siguiente entidad en VHDL:

Para poder realizar la lectura sobre el teclado, utilizaremos un contador de 2 bits para realizar la secuencia: ‘1110’, ‘1101’, ‘1011’ y

## **Software**

# Práctica 3: Uso de un periférico Wishbone

## **Hardware**

## **Software**

# Proyecto Final

## **Hardware**

## **Software**

# Anotaciones adicionales