Proyectos

1. PyBraille
2. ROBOT_6_DoF
3. prueba_sudoku
4. Traductor-a-braille
5. pyVHDL
6. UART32
7. prueba_sudoku
8. calculator
9. Import_mat_files
10. kicad_pcb
11. PMOD_board
12. PySampler
13. Alf-V
14. HDLHelper
15. BlockSim

1. PyBraille

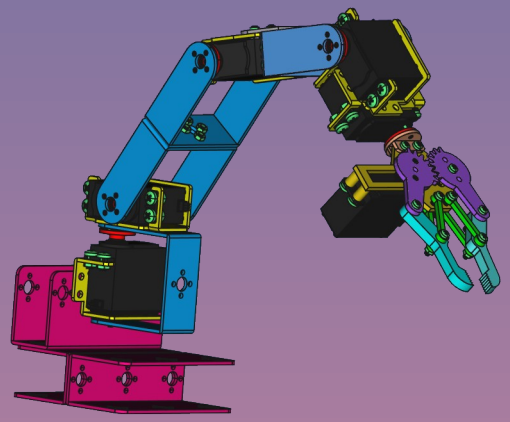
Librería de traducción a Braille escrita en **Python**

Enlace: https://github.com/DRubioG/pyBraille

2. ROBOT_6_DoF

Proyecto para el diseño, modelado y futura impresión 3D de un robot de 6 grados de libertad utilizando **FreeCad**.

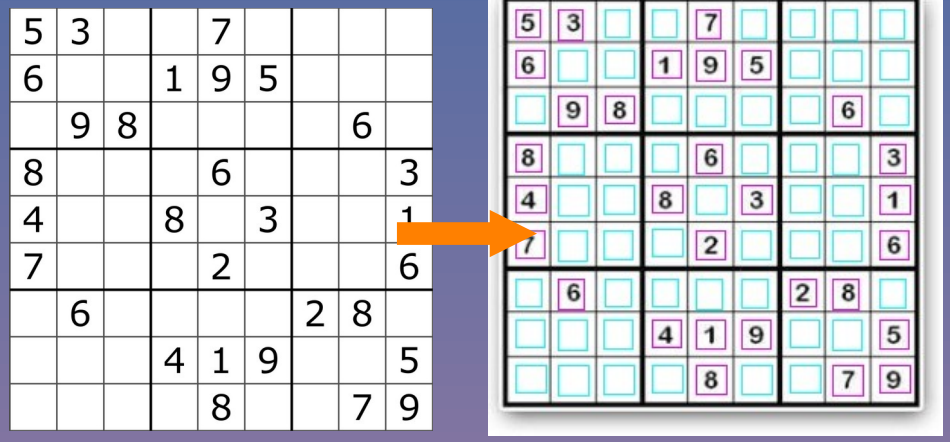
Enlace: https://github.com/DRubioG/ ROBOT_6_DoF



3. prueba_sudoku

Proyecto para la detección de casilla con números en sudokus utilizando **OpenCV** y **Python**

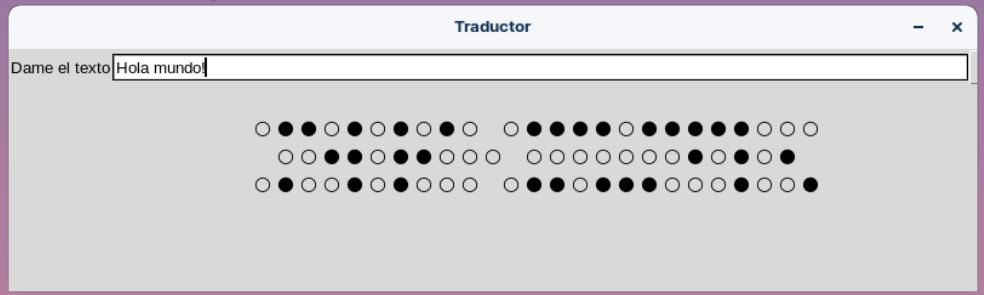
Enlace: https://github.com/DRubioG/ prueba_sudoku



4. Traductor-a-braille

Proyecto para la traducción de texto a Braille con una interfaz gráfica utilizando **Python** y **Tkinter**

Enlace: https://github.com/DRubioG/Traductor-a-braille



5. pyVHDL

Librería en Python para escribir código en **VHDL** usando funciones de **Python**

Enlace: https://github.com/DRubioG/pyVHDL

```python
import pyVHDL
v=pyVHDL("Prueba.vhd")
v.use("numeric_std")
v.generic("generico", "integer", 32)
v.port_in("puerto_entrada1")
v.port_in("puerto_entrada2", 23)
v.port_in("puerto_entrada3", 23, invert=1)
v.port_in("puerto_entrada4", bits=4, LSB=2)
v.port_out("puerto_salida")
v.port_inout("puerto_entrada_salida")
v.constant("cero", 23, 0)
v.signal("senal", 8)
```

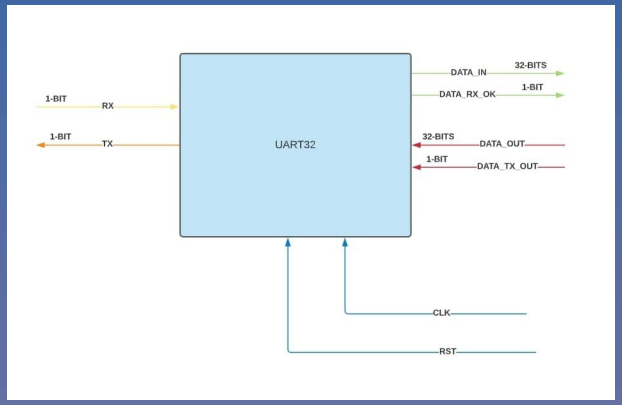
Equivalente:

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use iee.numeric_std.all;

entity Prueba is
generic(
    generico : integer := 32
);
port(
    puerto_entrada1 : in std_logic;
    puerto_entrada2 : in std_logic_vector(22 downto 0);
    puerto_entrada3 : in std_logic_vector(0 to 22);
    puerto_entrada4 : in std_logic_vector(6 downto 2);
    puerto_salida : out std_logic;
    puerto_entrada_salida : inout std_logic
);

end Prueba;

architecture arch_Prueba of Prueba is
signal senal : std_logic_vector(7 downto 0);
constant cero : unsigned(22 downto 0) := to_unsigned(0, 23);
begin


end architecture;
```

6. UART32 (en desarrollo)

Protocolo de comunicación basado en **UART** desarrollado en **VHDL**.

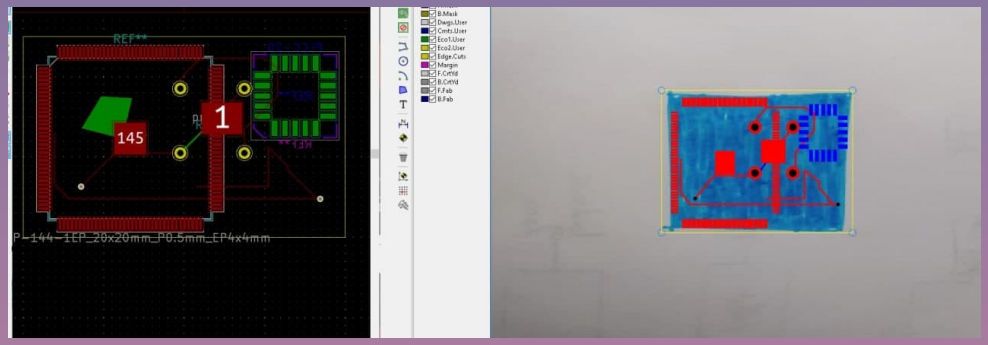
Enlace: https://github.com/DRubioG/ UART32



7. prueba_sudoku

Proyecto para la detección de casilla con números en sudokus utilizando **OpenCV** y **Python**

Enlace: https://github.com/DRubioG/ prueba_sudoku



8. calculator (en desarrollo)

Proyecto para la implementación de una calculadora en una FPGA usando **VHDL**.

Enlace: https://github.com/DRubioG/ Calculator

9. Import_mat_files

Proyecto para importar como variables en **Python** ficheros .mat de Matlab.

Explicación en mi blog: https://soceame.wordpress.com/2023/06/23/como-importar-un-mat-en-pythony-hacer-que-se-convierta-en-una-variable/

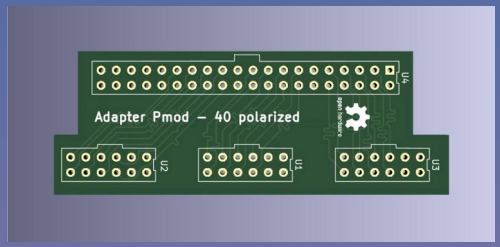
Enlace: https://github.com/DRubioG/Import_mat_files

```python
def load(file, *args):
    '''
    This function loads .mat files to Python like Python variables.
    Input:
        - file : the .mat file you want to load. Doesn't matter if ends in '.mat' or not.
        - specific_variables : this options load only specific variables from .mat file.
    '''

    if file[-4:] != ".mat":
        file += ".mat"

    import scipy.io
    mat = scipy.io.loadmat(file)
    for k,_ in mat.items():
        if k != "__header__" and k != "__globals__" and k != "__version__":

            # specific variable
            if len(args) != 0:
                for i in args:
                    if i == k:
                        if mat.get(k).shape[0] == 1:        # when list has only a row
                            if mat.get(k).shape[1] == 1:    # when object has one value
                                globals()[f"{k}"] = mat.get(k)[0][0].tolist()
                            else:
                                globals()[f"{k}"] = mat.get(k)[0].tolist()
                        else:
                            globals()[f"{k}"] = mat.get(k).tolist()

            else:
                # non-specific variable
                if mat.get(k).shape[0] == 1:
                    if mat.get(k).shape[1] == 1:
                        globals()[f"{k}"] = mat.get(k)[0][0].tolist()
                    else:
                        globals()[f"{k}"] = mat.get(k)[0].tolist()
                else:
                    globals()[f"{k}"] = mat.get(k).tolist()
```

10. kicad_pcb

Proyecto de desarrollo de una PCB en **KiCad** para adaptar los 40 pines de una placa de Alinx para conseguir adaptarlo con 3 Pmods.

Enlace: https://github.com/DRubioG/kicad_pcb

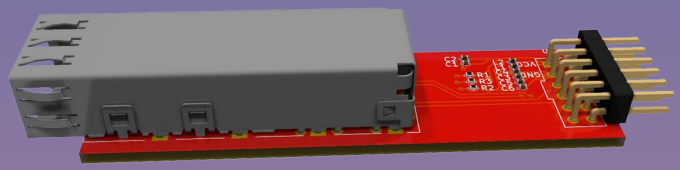


11. PMOD_board (en desarrollo)

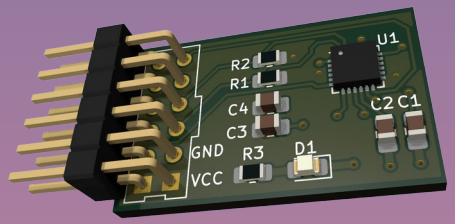
Proyecto para el desarrollo en **KiCad** dispositivos para conectar a conectores PMOD.

-SFP_MOD →

-IMU_PMOD



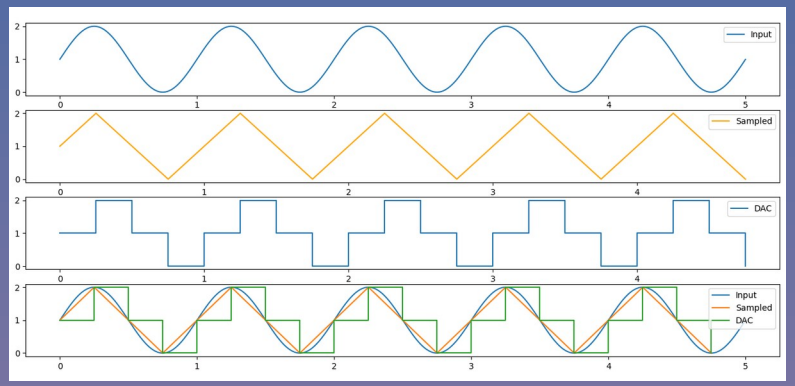
Enlace: https://github.com/DRubioG/PMOD_boards



12. PySampler (en desarrollo)

Proyecto para la simulación en **Python** del comportamiento de **ADCs** y **DACs**.

Enlace: https://github.com/DRubioG/pySampler



13. Alf-V (en desarrollo)

Proyecto de desarrollo de arquitecturas en **VHDL** para **FPGAs** basada en RISC-V

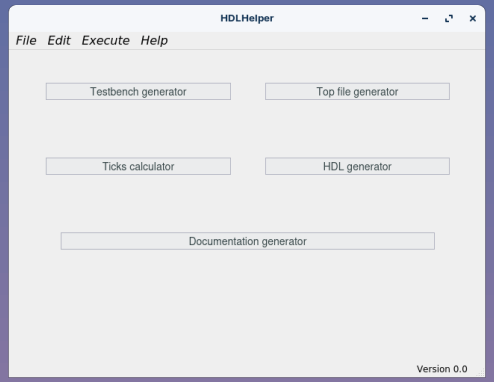
Enlace: https://github.com/DRubioG/ALF-V

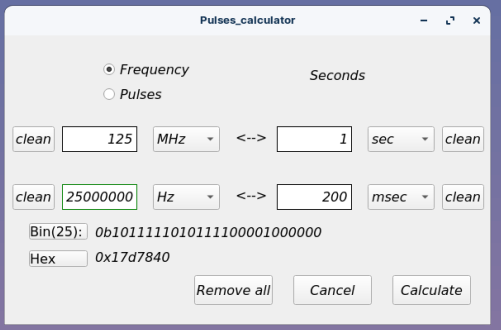
14. HDLHelper (en desarrollo)

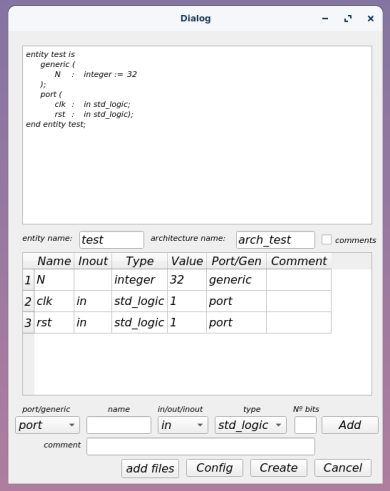
Proyecto de desarrollo de un conjunto de herramientas para facilitar el desarrollo de hardware para **FPGAs** en **VHDL** y **Verilog** utilizando **Python** y **PyQt**. Herramientas como:

- Testbench generator: generador de testbenchs automático
- Ticks calculator: una calculadora de pulsos
- HDL generator: generador de ficheros sobre la marcha
- Documentation generator: generador de un PDF con los comentarios del código

Enlace: https://github.com/DRubioG/HDLHelper







15. BlockSim (en desarrollo)

Proyecto de desarrollo de un herramienta en Python para la simulación de lazos de control, además, también genera el código en C para introducirlo en un microcontrolador.

Enlace: https://github.com/DRubioG/BlockSim