PASCO Florian

Hi there! 👋

🎓 I’m Florian Pasco, a **Master’s student in Engineering** specializing in **Electronics, Computer Science, and Mechatronics**.

👨‍💻 I am passionate about creating and sharing **open-source projects** that are reusable and easy to understand, helping others grow in the fields of **electronics, programming, and mechanics**.

🔧 Recently, I developed a **full-stack project** using **React, Next.js, Django, Python, and PostgreSQL**, showcasing my skills in modern web and backend development.

📚 Beyond technology, I enjoy **self-training**, exploring **personal development**, and learning from resources like the **French Polar Institute**.

🏕️ I love traveling and discovering new places—whether it’s **hiking**, **biking**, or going on **road trips in my car**!

🎯 I’ve also applied my skills in **reverse engineering**, successfully diagnosing and repairing **20+ electronic units**. Additionally, I designed and worked on a **thermal resistance control board** with feedback from a thermal resistance sensor for precise regulation.

HV2LV-PowerJST

🔋 Regulator 4.8V-15V -> 3.3V

HV2LV-PowerJST is an open-source PCB designed to step down **4.8V - 15V** to a stable **3.3V** output using an **AMS1117 voltage regulator**. The board features **JST-PH connectors** for easy integration into embedded projects, IoT devices, and prototyping setups.

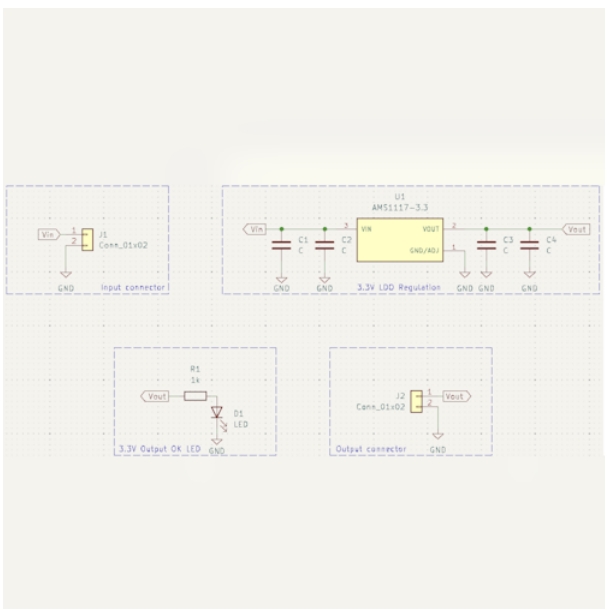
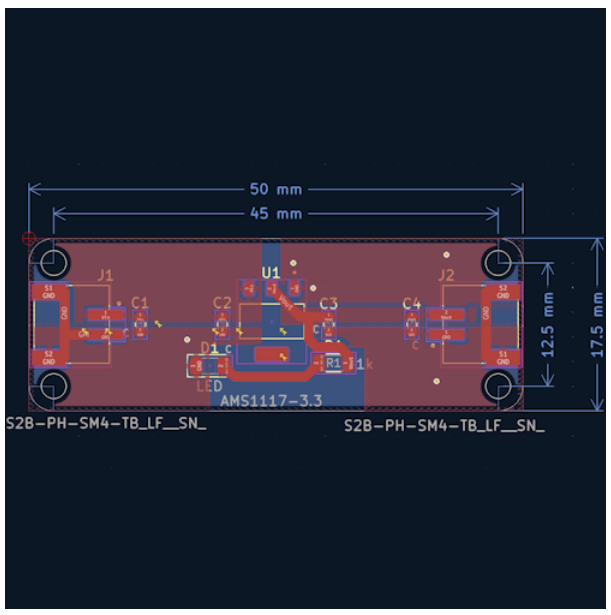
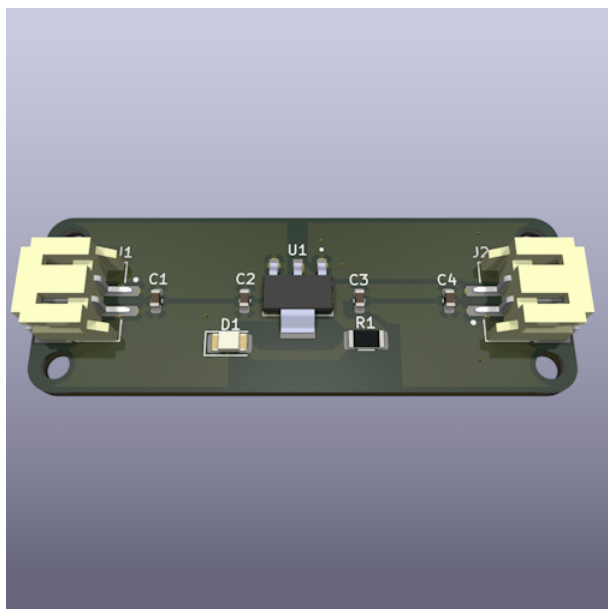
🎯 Purpose

- 🔌 **Efficient power conversion**: Converts input voltages from **4.8V to 15V** down to a fixed **3.3V**.
- 📏 **Compact and Adafruit-compatible**: Designed to fit within Adafruit's standard PCB footprint.
- 🛠️ **Open-source and customizable**: Modify and adapt the design to suit your specific needs.

📝 Features

🏷️ Feature	🔍 Description
⚡ **Input Voltage**	4.8V - 15V
🔋 **Output Voltage**	3.3V (fixed)
🛠️ **Regulator**	AMS1117-3.3
🔌 **Connector 1**	JST-PH (Input Voltage)
🔌 **Connector 2**	JST-PH (3.3V Output)
🔧 **Capacitors**	4 decoupling capacitors for stability
🖥️ **PCB Design**	Open-source & customizable
🌍 **Use Cases**	Powering 3.3V embedded systems, IoT devices, and sensors

📐 PCB Design Preview

📜 Schematic	🖥️ PCB Layout	🏗️ 3D
		

HeaterControl-Shield

🔥 STM32 Shield for Heat Control

HeaterControl-Shield is an open-source electronic board designed to control a **heating resistor**, measure the **consumed current**, and detect the **temperature** near the heating resistor. This board is shaped as an **Arduino shield** and optimized for use with an **STM32 Nucleo**.

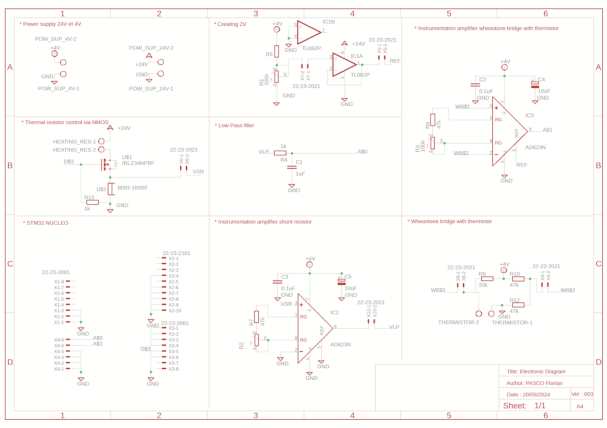
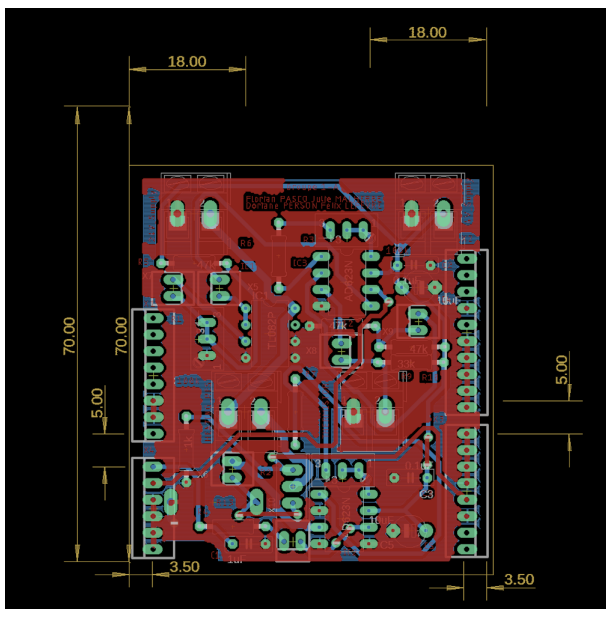
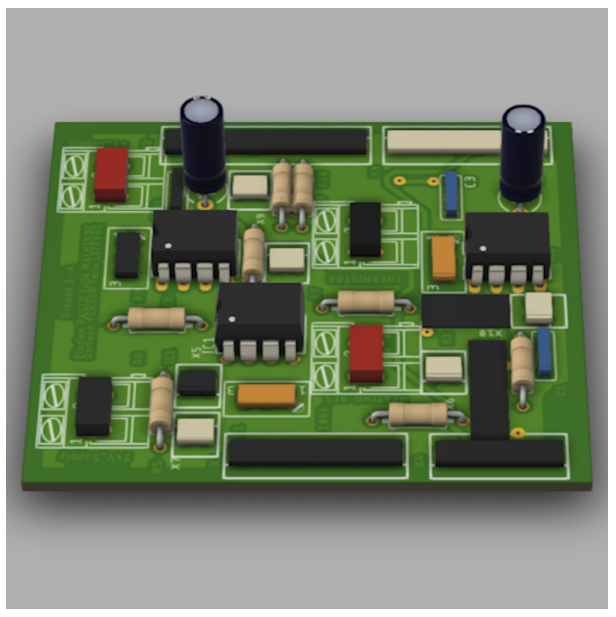
🎯 Main Features

- 🔥 **Heating resistor control** via an **N-channel MOSFET**.
- ⚡ **Current consumption measurement** using a **shunt resistor** and an **operational amplifier**.
- 🌡️ **Temperature sensing** using a **thermistor** integrated into a **Wheatstone bridge**, amplified by an **instrumentation amplifier**.
- 🛠️ **Open-source and customizable design** to adapt to specific project needs.

📝 Technical Specifications

🏷️ Feature	🔍 Description
🔌 **Power Input**	5V - 12V
🔥 **Heating Control**	N-channel MOSFET for power control
⚡ **Current Measurement**	Shunt resistor + Operational amplifier
🌡️ **Temperature Sensing**	Wheatstone bridge + Instrumentation amplifier
🔄 **Interface**	Compatible with **Arduino Shield** and **STM32 Nucleo**
🖥️ **PCB Design**	Open-source & customizable
🌍 **Applications**	Thermal control projects, embedded systems, temperature regulation

📐 PCB Preview

📜 Schematic	🖥️ PCB Layout	🏗️ 3D View
		

🔗 Main Connections

Pin	Function
VIN	Main power input
GND	Ground
HEAT_CTRL	PWM signal to activate heating
CURR_SENSE	Amplified output of current measurement
TEMP_SENSE	Amplified output of temperature measurement

MicroUSB2JST

🔌 MicroUSB -> JST-SH & JST-PH

MicroUSB2JST is an open-source PCB that acts as a bridge between a Micro USB port and JST connectors (JST-SH and JST-PH). This module is designed to simplify connections for embedded projects, prototyping, and power distribution.

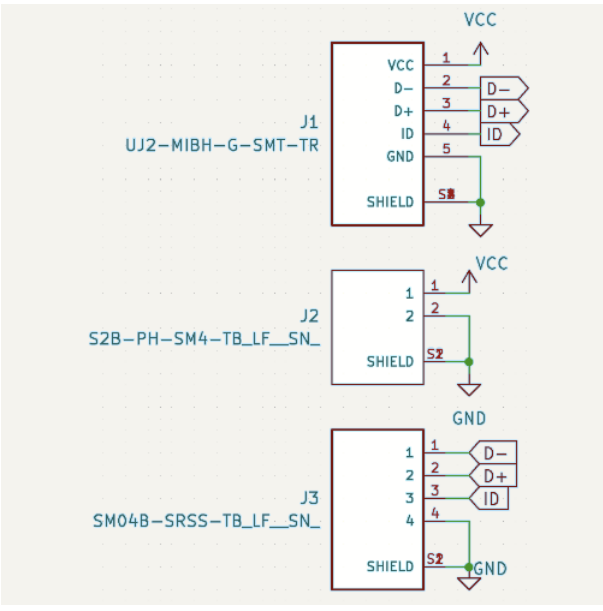
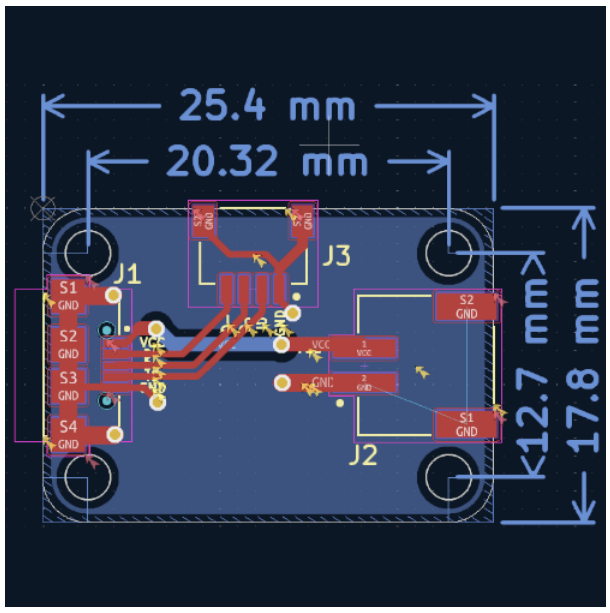
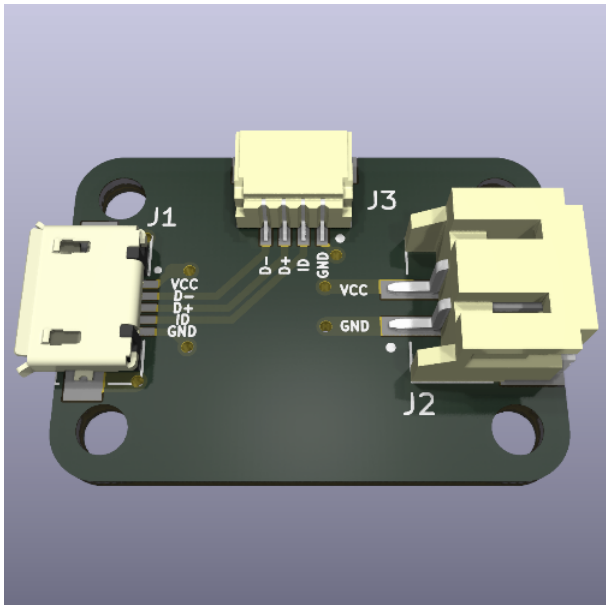
🎯 Purpose

- 🔌 **Convenient power and data interface**: Easily connect USB power or data lines to JST-equipped devices.
- 🩹 **Compact and Adafruit-compatible**: Designed to fit within Adafruit's standard PCB footprint.
- 🛠️ **Open-source and customizable**: Modify and adapt the design to suit your specific needs.

📝 Features

🏷️ Feature	🔍 Description
🔌 **Connector 1**	Micro USB (⚡ power & 🔄 data)
🔌 **Connector 2**	JST-PH (⚡ higher current capacity)
🔌 **Connector 3**	JST-SH (🩹 small form factor)
🖥️ **PCB Design**	🆓 Open-source & 🖍️ customizable
🎯 **Use Cases**	⚡ Power distribution, 🛰️ sensor connections, 🤖 embedded systems

📐 PCB Design Preview

📜 Schematic	🖥️ PCB Layout	🏗️ 3D
		

StateMachineSafe

🔐 Safe control via state machine!

StateMachineSafe is a project aimed at designing and simulating a state machine to manage the opening and closing of a safe. This machine is built using D flip-flops and logic gates, ensuring precise logical control of the mechanism.

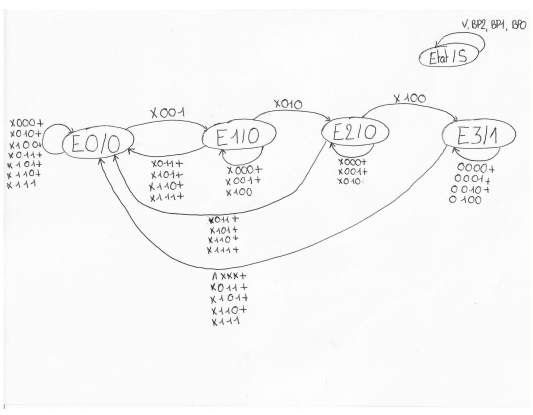
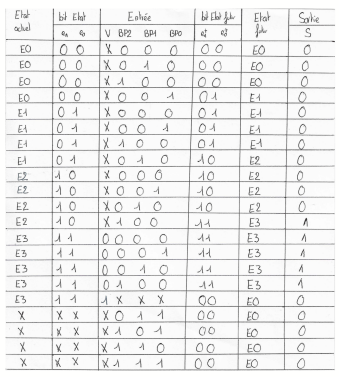
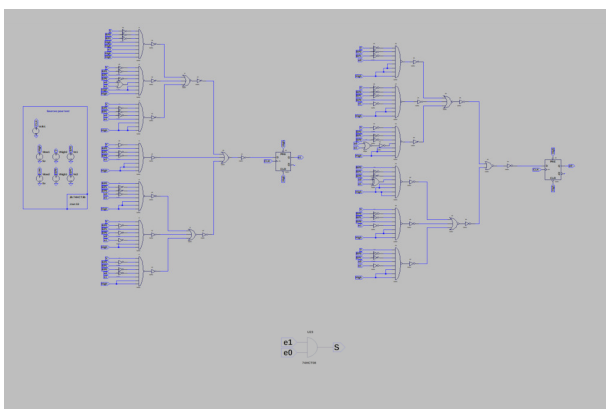
🎯 Purpose

- 📜 **State Machine Design**: Development of a state graph defining the safe's behavior.
- 🧮 **Transition Table and Karnaugh Map**: Derivation of logical equations necessary for system operation.
- 🔬 **LTSpice Simulation**: Verification and validation of the logical circuit through LTSpice simulation.
- 🔑 **Secure Access Management**: Implementation of a reliable mechanism ensuring the safe's opening and closing according to a defined sequence.

📝 Features

🏷️ Feature	🔍 Description
🔄 **State Machine**	Modeling of the control system with a state graph
📊 **Transition Table**	Definition of transitions between states
📏 **Karnaugh Map**	Simplification of logical equations
🛠️ **Hardware Implementation**	Use of D flip-flops and logic gates
🖥️ **LTSpice Simulation**	Verification of behavior through simulation

📐 System Architecture

🎭 State Graph	🗂️ Transition Table	⚡ Logic Circuit
		

AMS1117DC3V3

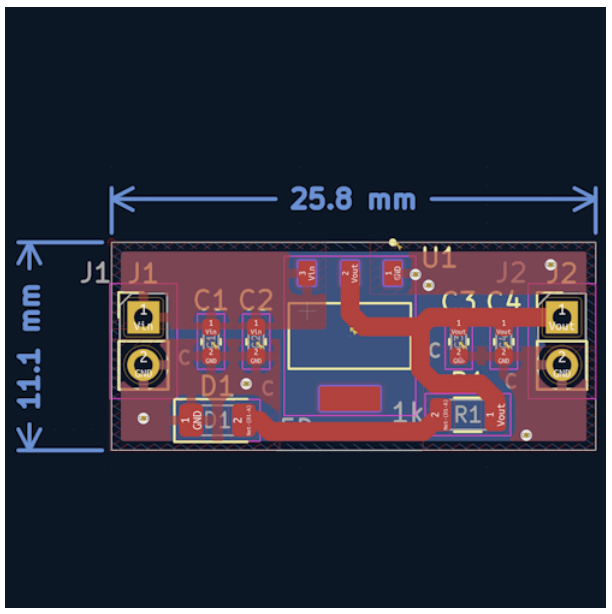
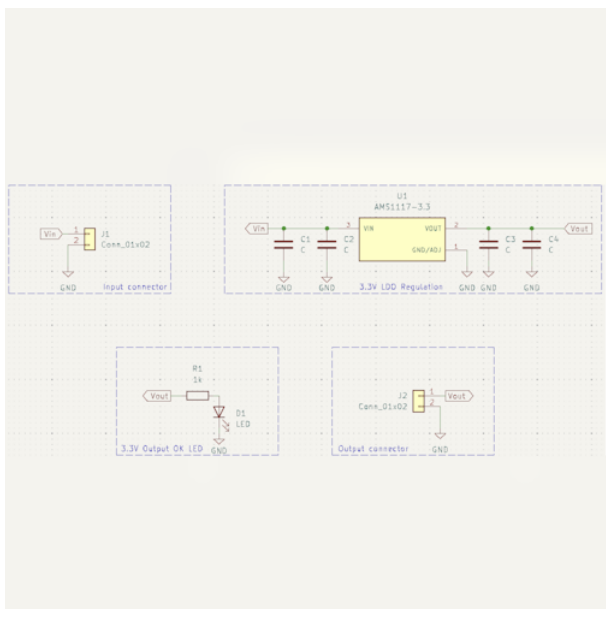
⚡ AMS1117 3.3V Buck reverse-engineered

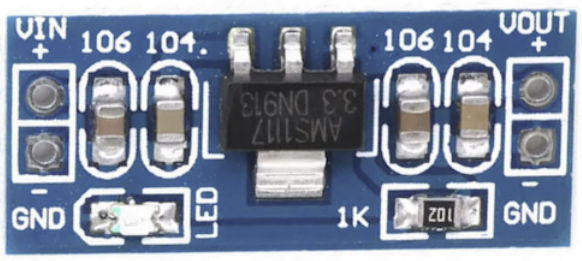
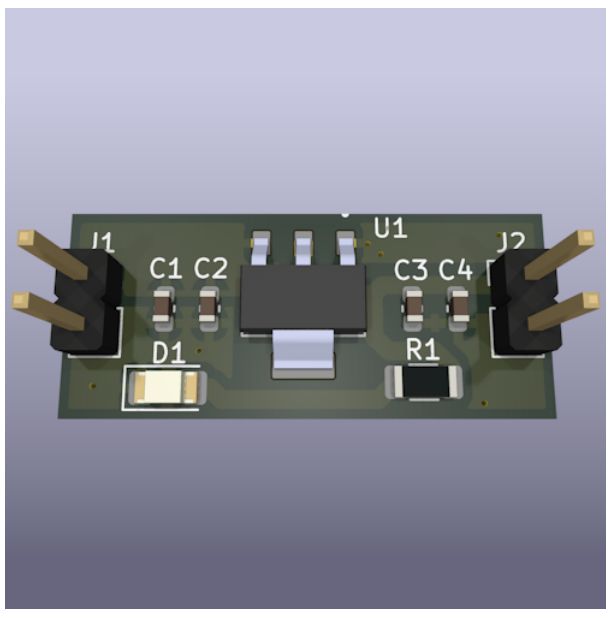
An open-source reverse-engineered version of the AMS1117 3.3V DC-DC buck converter module, based on the original component available here.
This project aims to provide insights into DC-DC voltage regulation and offer a customizable alternative for power management in embedded systems.

🎯 Purpose

- 🔍 **Reverse engineering**: Understanding the design and functionality of the AMS1117-based voltage regulator.
- 🛠️ **Skill development**: Enhancing expertise in PCB design and power electronics.
- 🔄 **Future adaptation**: Leveraging this knowledge to develop custom voltage regulation solutions for embedded applications.

📝 Features Comparison: Original vs. Reverse-Engineered

Feature	Original Module	Reverse-Engineered Version
🖥️ PCB Design	Proprietary	Open-source & customizable
🔌 Input Voltage	4.8V - 15V	4.8V - 15V
⚡ Output Voltage	3.3V (fixed)	3.3V (fixed)
📦 Max Current	1500 mA	1500 mA
🏷️ Regulator Chip	AMS1117-3.3	AMS1117-3.3
🖇️ Mechanical Drawing		
📄 Reverse-Engineered Schematic	N/A	

Feature	Original Module	Reverse-Engineered Version
📷 Photo		

🛠️ How to Use

📌 Wiring Guide

Pin	Description
VIN	Input Voltage (4.8V - 15V)
GND	Ground
VOUT	Regulated 3.3V Output

CP2102USB2UART

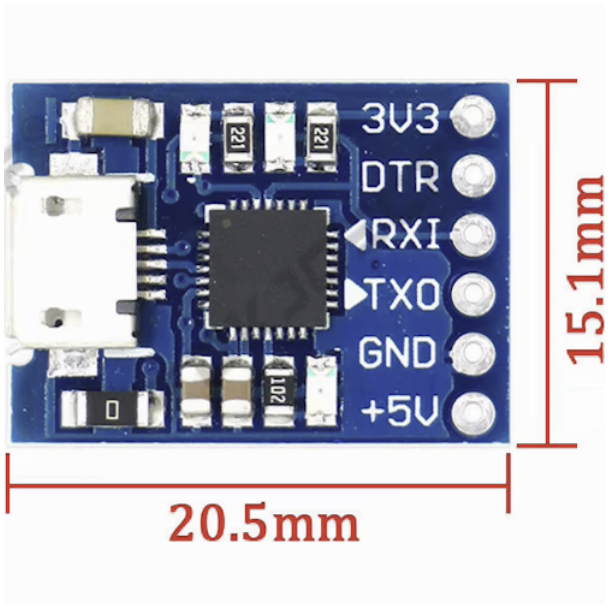
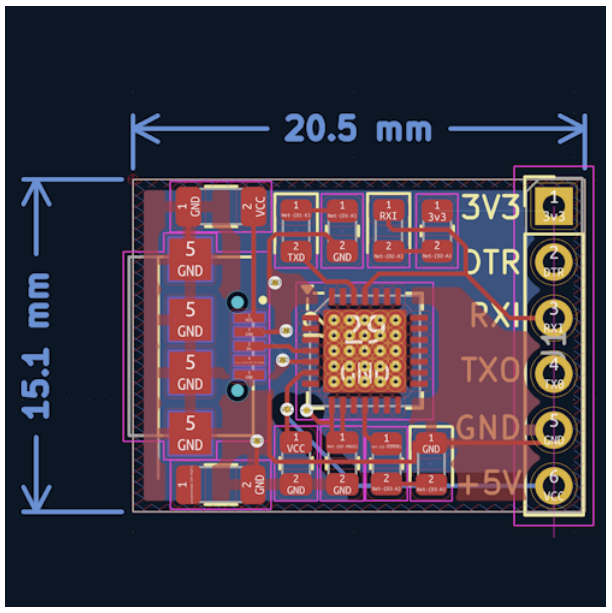
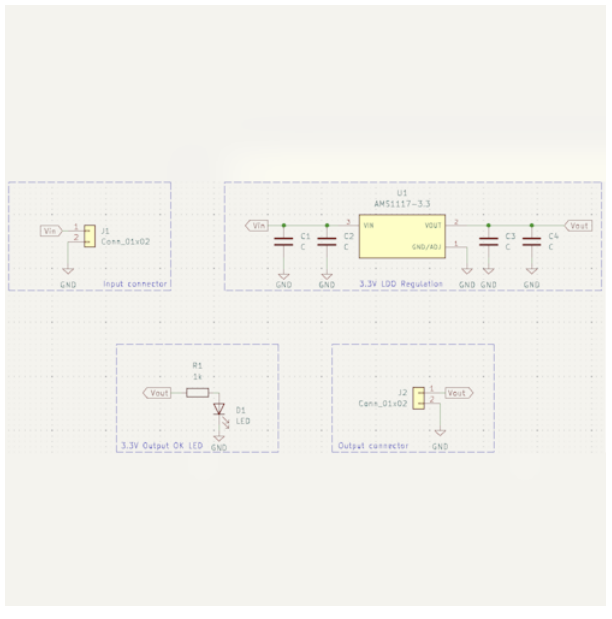
🔌 CP2102 USB to UART reverse-engineered

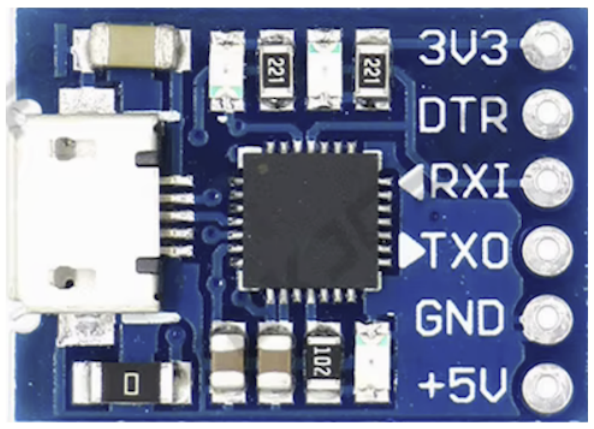
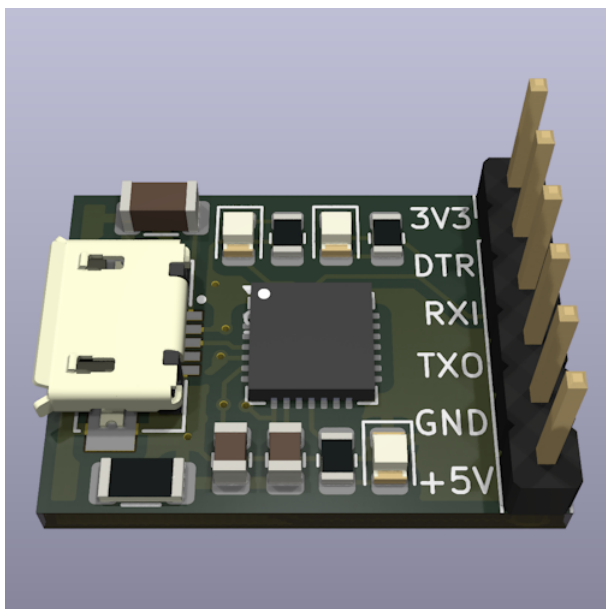
An open-source reverse-engineered version of the CJMCU CP2102 USB to UART TTL adapter, based on the original component available here. This project aims to provide insights into USB-to-serial communication and offer a customizable alternative for embedded system developers.

🎯 Purpose

- 🔍 **Reverse engineering**: Understanding the design and functionality of the CP2102USB2UART.
- 🛠️ **Skill development**: Enhancing expertise in PCB design and USB-to-serial communication.
- 🔄 **Future adaptation**: Leveraging this knowledge to develop custom USB-to-UART solutions for embedded systems.

📝 Features Comparison: Original vs. Reverse-Engineered

Feature	Original Module	Reverse-Engineered Version
🖥️ PCB Design	Proprietary	Open-source & customizable
🔌 USB Connector	Type-C & Micro USB	Type-C & Micro USB
🧩 Chipset	CP2102	CP2102
📌 Pin Mapping	6-Pin UART TTL	6-Pin UART TTL
⚡ Supported Voltage	3.3V / 5V	3.3V / 5V
🤲 Mechanical Drawing		
📝 Reverse-Engineered Schematic	N/A	

Feature	Original Module	Reverse-Engineered Version
📷 Photo		

🛠️ How to Use

📌 Wiring Guide

CP2102 Pin	Description
TXD	Transmit Data
RXD	Receive Data
GND	Ground
3V3	3.3V Power Output
5V	5V Power Output
DTR	Data Terminal Ready

MicroUSB2DIP

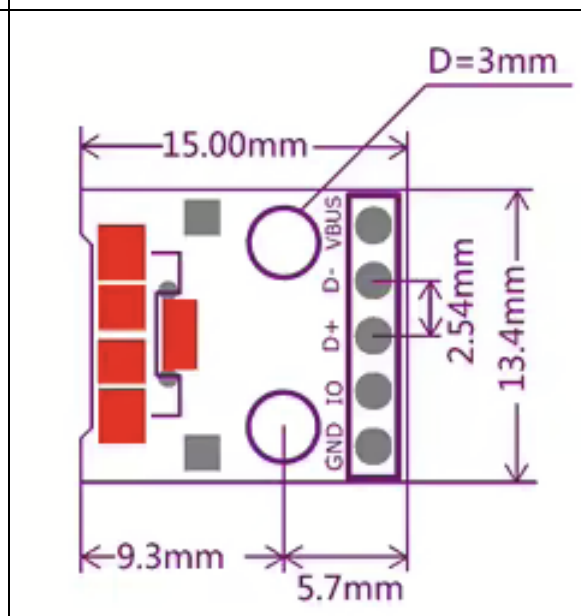
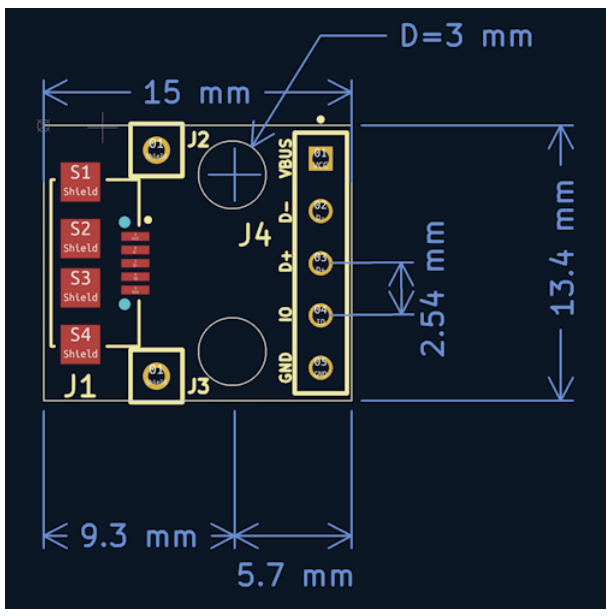
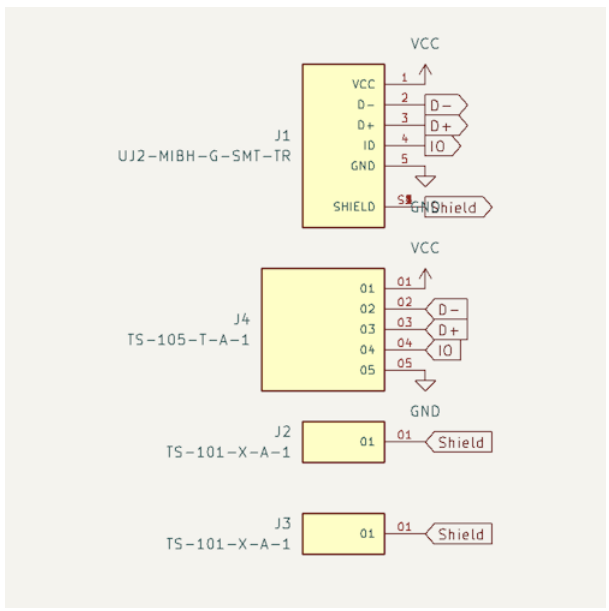
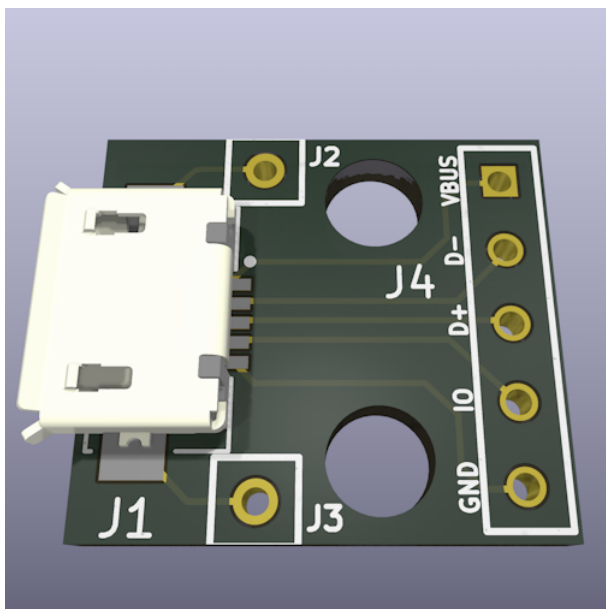
🔌 Micro USB to DIP reverse-engineered

This is an open-source reverse-engineered version of a Micro USB to DIP adapter, based on the original component available here. The goal of this project was to practice reverse engineering as a learning exercise and to prepare for a future adaptation in a larger project.

🎯 Purpose

- 🔍 **Reverse engineering**: Understanding the design and functionality of this micro USB to DIP connector.
- 🛠️ **Skill development**: Enhancing expertise in PCB design and hardware analysis skills.
- 🔄 **Future adaptation**: Leveraging this knowledge for embedded applications.

📝 Features Comparison: Original vs. Reverse-Engineered

Feature	Original Module	Reverse-Engineered Version
🖥️ PCB Design	Proprietary	Open-source & customizable
🔌 Connector Type	Micro USB	Micro USB
📌 Pin Mapping	Standard DIP	Standard DIP
👐 Mechanical Drawing		
📝 Reverse-Engineered Schematic	N/A	
📷 Photo		

🛠️ How to Use

📌 Wiring Guide

Pin	Function
VBUS	+5V
D-	Data -
D+	Data +
ID	Mode detect (A: GND, B: Open)
GND	Ground

AntiVuvuzelaFilter

🎶 Noise filter for clear audio 🎙️

Anti-Vuvuzela Filter is an open-source project dedicated to **second-order analog filters** and beyond. This project was initially developed to design an **“anti-vuvuzela” filter**, aiming to attenuate the distinctive and persistent sound of vuvuzelas while preserving the clarity of commentators’ voices during the **2010 FIFA World Cup**.

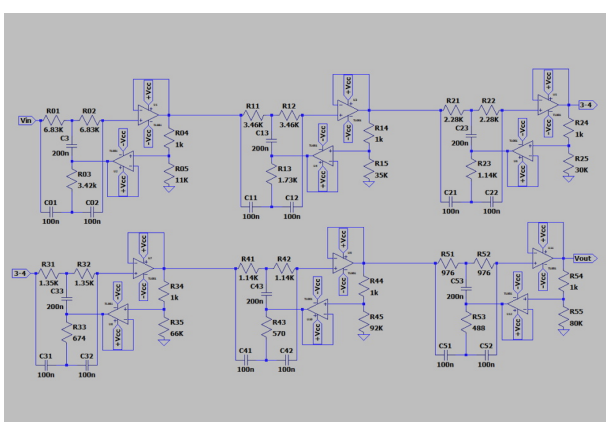
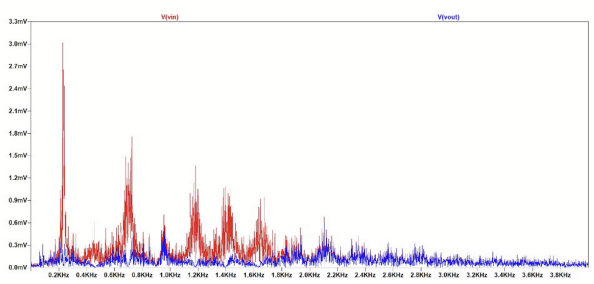
🎯 Purpose

- 🔇 **Targeted Noise Reduction**: Specifically designed to **attenuate vuvuzela noise** while maintaining the intelligibility of speech.
- 🎚️ **Second-Order Analog Filtering**: Utilizing advanced filtering techniques for efficient noise cancellation.
- 🛠️ **Open-source and Customizable**: Modify and adapt the design for other audio filtering applications.

📝 Features

🏷️ Feature	🔍 Description
🎼 **Filter Type**	Second-order analog filter
🎯 **Target Frequency**	233 Hz (typical vuvuzela frequency)
🎙️ **Voice Preservation**	Maintains speech clarity
🔧 **Components**	Resistors, capacitors, and operational amplifiers
🖥️ **Simulation Tools**	Jupyter Notebook, LTSpice
🛠️ **Real-world Testing**	Assembled and tested in real conditions
🔌 **Input**	Analog audio signal
🔊 **Output**	Cleaned audio signal with reduced vuvuzela noise
🌎 **Use Cases**	Audio signal processing, speech enhancement, noise reduction

📐 Simulation & Testing

🛠️ LTSpice Circuit	📜 Simulation
	

DtmfCodeAnalyzer

🎶 Analyze and filter DTMF signals

DtmfCodeAnalyzer is an open-source project designed to analyze and filter DTMF (Dual-Tone Multi-Frequency) signals used in conventional telephony. The project provides functionalities to detect and identify keypresses from audio recordings, as well as remove DTMF tones to isolate voice signals.

🎯 Purpose

- 🎵 **DTMF Signal Recognition**: Identify the key pressed based on the audio recording of its emitted frequencies.
- 🔇 **DTMF Noise Removal**: Extract and suppress DTMF tones from an audio sample to reveal underlying speech.
- 📊 **Mathematical Approach**: Use vector projection in an Euclidean space to determine the closest frequency matches.

📝 Features

🏷️ Feature	🔍 Description
🎛️ **DTMF Frequencies**	Combination of two distinct tones per key
🔠 **Key Identification**	Detects and determines the key pressed
🔉 **Noise Filtering**	Removes DTMF tones while preserving speech
📉 **Mathematical Model**	Projects signals into a vector space for analysis
🎤 **Audio Processing**	Works with recorded audio samples
🖥️ **Open-Source**	Fully customizable and modifiable

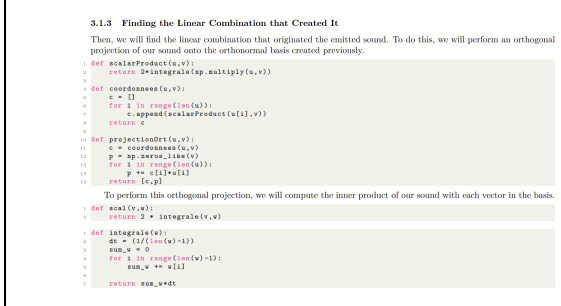
📐 Signal Processing Approach

🎵 Frequency Vector Representation

```python
def function_u(i,t):
    return np.sin(t*w[(i)%7]*2*np.pi+(i//7)*np.pi/2)

def createOrthonormalBasis(sample_rate,signal_freq):
    u = []
    t_ech=np.linspace(0.0,1.0,fs) # intervalle d'echantillonnage
    for i in range(14):
        u.append(np.array(function_u(i,t_ech)))
    return u
```

🧮 Euclidean Projection



3.1.3 Finding the Linear Combination that Created It

Then, we will find the linear combination that originated the emitted sound. To do this, we will perform an orthogonal projection of our sound onto the orthonormal basis created previously.

```python
def scalarProduct(u,v):
    return 2*integrale(np.multiply(u,v))

def coordonnees(u,v):
    c = []
    for i in range(len(u)):
        c.append(scalarProduct(u[i],v))
    return c

def projectionOrt(u,v):
    c = coordonnees(u,v)
    p = np.zeros_like(v)
    for i in range(len(u)):
        p += c[i]*u[i]
    return [c,p]
```

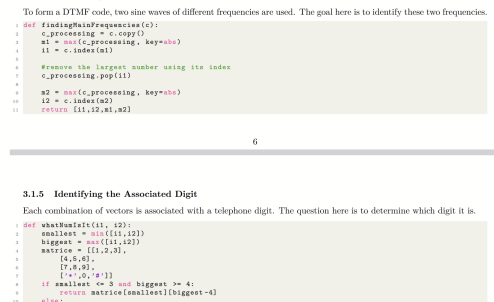
To perform this orthogonal projection, we will compute the inner product of our sound with each vector in the basis.

```python
def scal(v,w):
    return 2 * integrale(v,w)

def integrale(w):
    dt = (1/(len(w)-1))
    sum_w = 0
    for i in range(len(w)-1):
        sum_w += w[i]

    return sum_w*dt
```

🎛️ Filtering



3.1.4 Identifying the Two Main Frequencies

To form a DTMF code, two sine waves of different frequencies are used. The goal here is to identify these two frequencies.

```python
def findingMainFrequencies(c):
    c_processing = c.copy()
    m1 = max(c_processing, key=abs)
    i1 = c.index(m1)

    #remove the largest number using its index
    c_processing.pop(i1)

    m2 = max(c_processing, key=abs)
    i2 = c.index(m2)
    return [i1,i2,m1,m2]
```

6

3.1.5 Identifying the Associated Digit

Each combination of vectors is associated with a telephone digit. The question here is to determine which digit it is.

```python
def whatNumIsIt(i1, i2):
    smallest = min([i1,i2])
    biggest = max([i1,i2])
    matrice = [[1,2,3],
        [4,5,6],
        [7,8,9],
        ['*',0,'#']]
    if smallest <= 3 and biggest >= 4:
        return matrice[smallest][biggest-4]
    else:
        return "La combinaison linéaire ne permet pas de déterminer une touche"
```

AntUpRising

Global Game Jam 2023

AntUpRising is an engaging 2D platformer created during the Global Game Jam 2023. Step into the shoes of an ant and dive into a vibrant world where your mission is to bring food back to the queen ant. But beware! Strange roots have taken over the anthill, causing the ants to become aggressive and unpredictable.

🎯 Purpose

- 🍽️ **Food Retrieval**: Collect resources to nourish the queen ant.
- ⚔️ **Avoid Threats**: Dodge the crazed ants and other dangers along your journey.
- 👑 **Reach the Queen**: Overcome obstacles to make your way to the queen's chamber.

📝 Features

- 🌍 **Vibrant 2D Environment**: Immerse yourself in a colorful and detailed world.
- 🤸 **Dynamic Gameplay Mechanics**: Interact with enemies and tackle various challenges.
- 🎮 **Immersive Gaming Experience**: Enjoy an exciting adventure through the tunnels of the anthill.

👥 Team Members

- Mael Madec
- Florian Pasco
- Romain Cloâtre
- Théo De Morais
- Romain Fauvel

BoardMapper

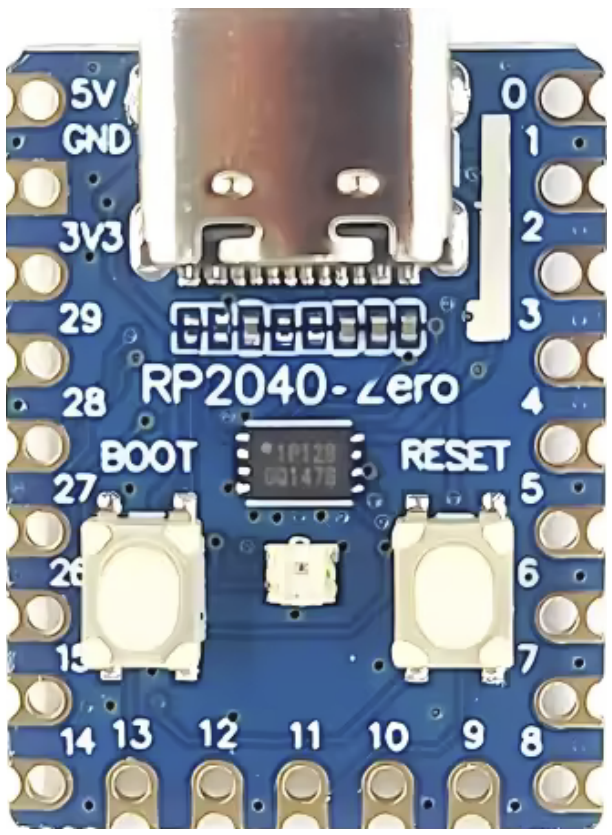
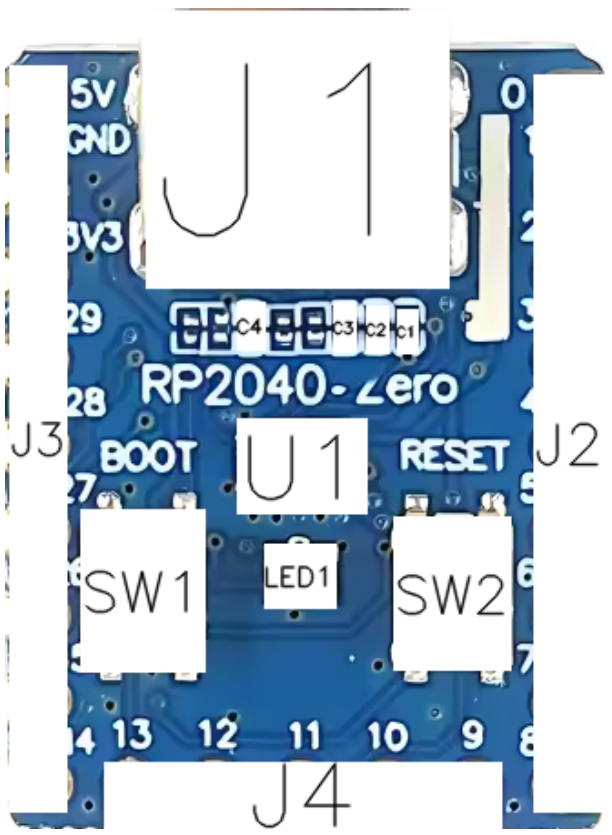
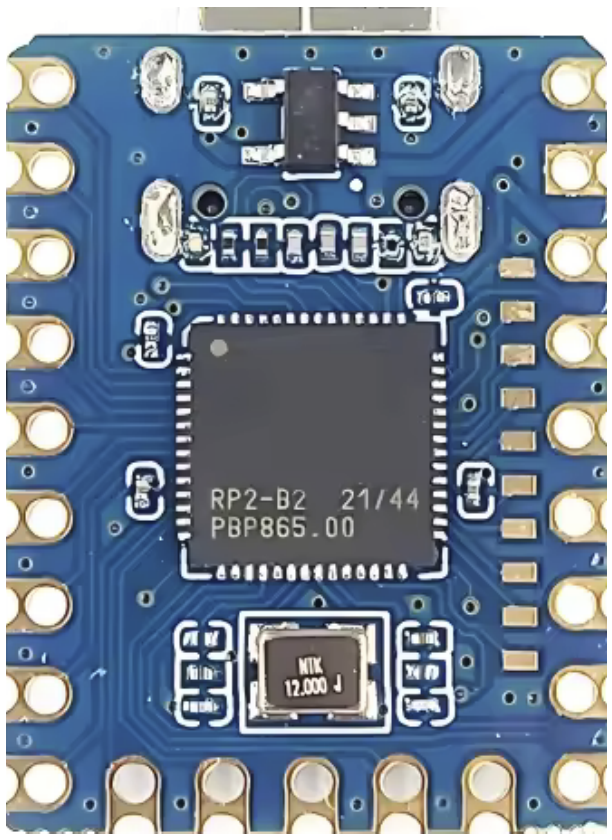
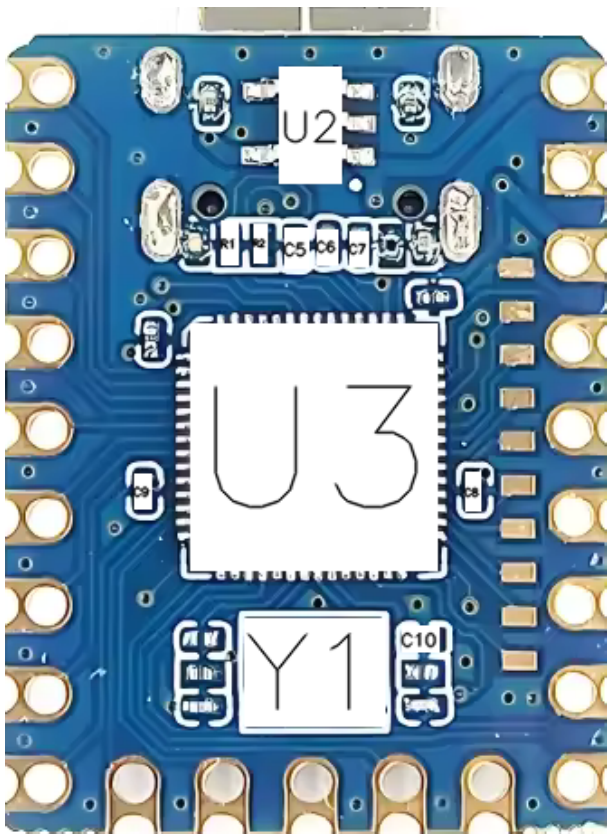
🛠️ PCB placement map generator

BoardMapper is an open-source tool designed to automatically generate PCB layout. It labels component references (e.g. U1, R1, C1) directly on the circuit image, facilitating component identification for reverse engineering purposes.

🎯 Purpose

- 🤖 **Automation**: Eliminates the need for manual placement annotation on PCB layouts.
- ⏱️ **Efficiency**: Saves time for engineers and makers working on PCB assembly and debugging.
- 🔍 **Clarity**: Provides a clear visual reference for debugging, testing, and manufacturing.
- 💻 **Cross-Platform**: Works on Windows, Linux, and macOS systems.

📝 Annotation

Position	Original	Annotated
Top		
Bottom		

📋 Requirements

- **Python**: Version 3.6 or higher
- **Required Libraries**:
  - `opencv-python` (for image processing)
  - `lxml` (for XML parsing)

🚀 Installation Instructions

🛠️ Setup

1. **Clone the repository** or **Download the project** to your local machine.

2. **Labeling the PCB**:
   - **Step 1**: Take a photo of both the top and bottom layers of the chosen PCB.
   - **Step 2**: Place the `top.png` and `bottom.png` images into the `input` folder.
   - **Step 3**: Install the latest version of LabelImg.
   - **Step 4**: Open `top.png` in LabelImg and draw bounding boxes around each component. Label each component according to its type:
     - **R**: Resistor
     - **C**: Capacitor
     - **L**: Inductor
     - **F**: Fuse
     - **POT**: Potentiometer
     - **D**: Diode
     - **LED**: LED
     - **Q**: Transistor (BJT, MOSFET)
     - **U**: Integrated Circuit (IC)
     - **J**: Connector
     - **K**: Relay
     - **SW**: Switch
     - **Y**: Quartz / Resonator
     - **SP**: Speaker
     - **ANT**: Antenna

     LabelImg Shortcuts:

     - 🖍️ **W**: Draw a new rectangular bounding box (RectBox)
     - ❌ **D**: Delete the last drawn bounding box
     - 💾 **Ctrl + S**: Save the annotation as an XML file
     - ⏪ **Ctrl + Z**: Undo the last action
     - 📋 **Ctrl + C**: Copy a bounding box
     - 📏 **Ctrl + V**: Paste a copied bounding box
     - 🔍 **Ctrl + A**: Select all bounding boxes
     - 🔄 **Ctrl + R**: Rotate the image (for better labeling)
     - 🚫 **Esc**: Cancel the current operation or close a dialog box
   - **Step 5**: After labeling the `top.png`, save the annotation as `top.xml`.
   - **Step 6**: Repeat the labeling process for the `bottom.png` and save it as `bottom.xml`.
   - **Step 7**: Place both `top.xml` and `bottom.xml` into the `input` folder.
3. **Running the Tool**:
   - **Windows**: Double-click on `setup_and_run.bat` to automatically run the script. The tool will read the XML annotations, draw bounding boxes on the images, and save the annotated images.
   - **Linux/macOS**: You can run the script from the terminal:

     ```
     chmod +x script.sh
     ./script.sh
     ```

4. 📁 **Output**:
   - After the script has executed, navigate to the `output` folder to find the resulting annotated images:
     - `top_annotated.png`
     - `bottom_annotated.png`

🤝 Contributions

If you'd like to contribute to the project, please follow these steps:

1. Fork the repository
2. Create a feature branch
3. Commit your changes
4. Push to the branch
5. Open a pull request

We welcome any contributions to improve BoardMapper! 🎉

CubeCourse2D

🕹️ Fast-paced random platformer!

Game Preview

CubeCourse2D is a fast-paced platformer where the player must survive as long as possible by jumping from platform to platform. But be careful—platforms are generated randomly, making each game unique and unpredictable!

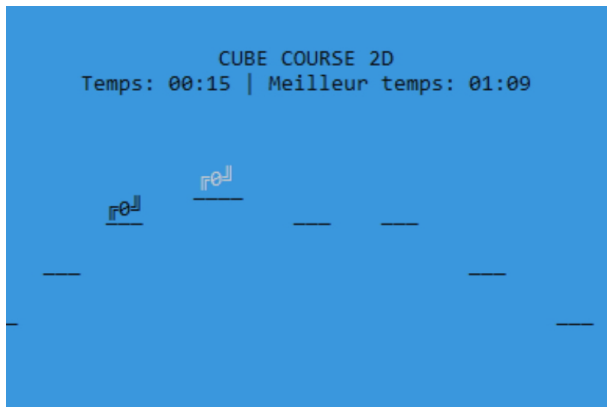
🎯 Objective

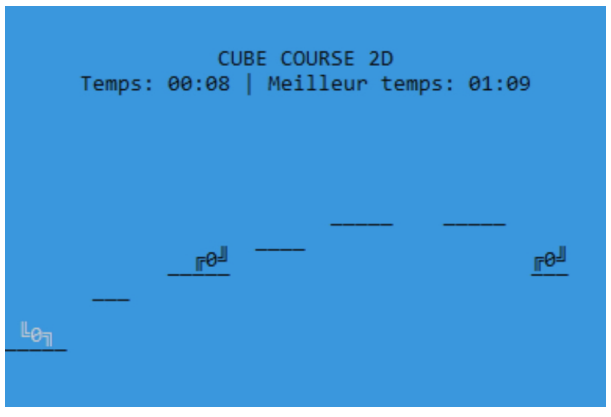
- 🏃 **Survival**: Stay alive by jumping across platforms.
- ⚠️ **Random Challenges**: Platforms and enemies appear randomly, keeping you on your toes.
- 🏆 **Competition**: Test your reflexes and beat your high scores!

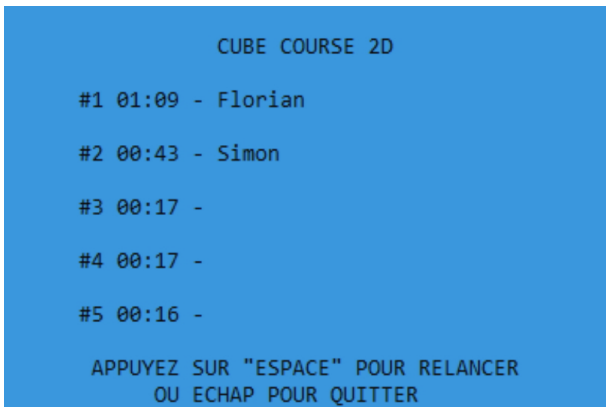
📝 Features

- 🎨 **Minimalist Design**: A simple yet effective aesthetic for full immersion.
- 🎲 **Random Generation**: No two levels are the same—every run is a new challenge!
- ⚔️ **Enemy Elimination**: Jump on enemies to defeat them, but be careful not to miss!
- 🎶 **Inspired by Geometry Dash**: A fast-paced and addictive gameplay experience.

📷 Screenshots







VATN-WaterGame

💧 A strategic game on water management!

VATN (from the Old Norse word for “water”) is a game designed to raise awareness about water management. Players must make critical daily decisions to address major water-related issues in their country, influencing key parameters that determine the nation’s survival.

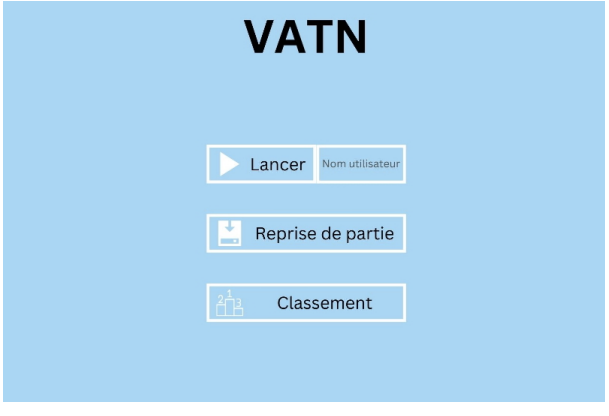
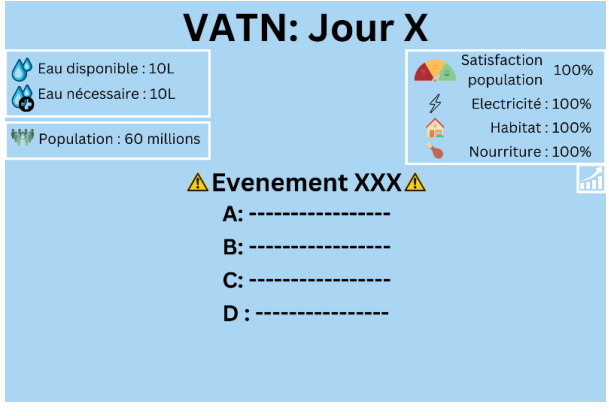
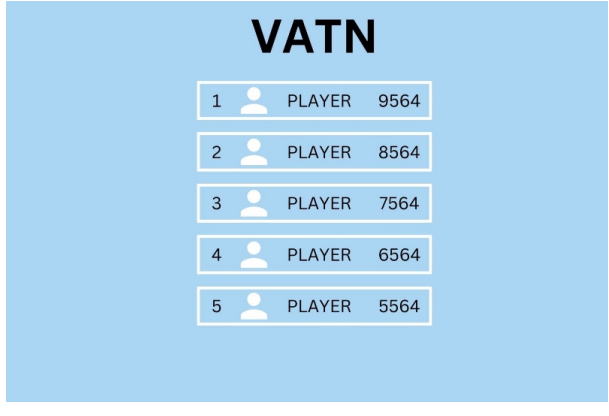
🎯 Purpose

- 💧 **Water Management Awareness**: Educate players on the importance of sustainable water policies.
- 🎮 **Engaging Decision-Making**: Every day presents a new challenge that affects the nation’s status.
- 🏆 **Survival & Strategy**: Keep your country alive as long as possible by maintaining stability.
- 📊 **Progress Tracking**: Visualize the evolution of key parameters through in-game graphs.

📝 Features

🏷️ Feature	🔍 Description
🌎 **Game Type**	Strategic Decision-Making Simulation
🗓️ **Daily Choices**	Players make decisions each day affecting country parameters
📉 **Dynamic Statistics**	Key indicators fluctuate based on player actions
💀 **Game Over**	The country collapses if the population reaches 0
📂 **Save & Load**	Resume previous games using saved files
🏅 **Rankings**	Compare results with previous local games
📊 **Graphical Summary**	Track country status evolution over time

📺 Gameplay Overview

🎮 Main Menu	📊 In-Game Statistics	🏆 Endgame Rankings
		

WorldTourRidersDatabase

🚴 SQLite database for pro cyclists

WorldTourRidersDatabase is an SQLite-based project designed to manage professional cyclists and their teams affiliated with the International Cycling Union (UCI). The database enables structured data storage, querying, and management of riders and their associations with World Tour teams.

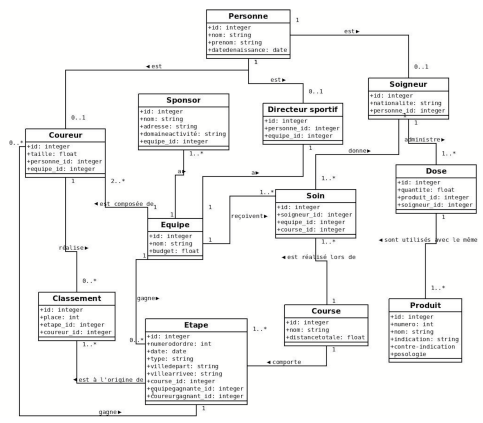
🎯 Purpose

- 📂 **Data Storage**: Efficiently store and manage rider information within an SQLite database.
- 🔍 **Advanced Queries**: Utilize SQL queries to retrieve, filter, and analyze cycling-related data.
- 🔗 **Table Associations**: Implement relational connections between riders, teams, and relevant cycling entities.
- 📊 **UML Modeling**: Design and visualize the database schema using UML modeling techniques.

📝 Features

🏷️ Feature	🔍 Description
📌 **Rider Management**	Store and organize details about professional cyclists
🏆 **Team Associations**	Link riders to their respective World Tour teams
🔍 **SQL Queries**	Perform complex queries to extract meaningful insights
📊 **Database Normalization**	Ensure efficient and scalable data structure
🖥️ **SQLite Integration**	Fully functional within an SQLite environment
📜 **UML Diagrams**	Visualize table relationships for better understanding

📐 Database Structure & Design



ZEthyDex

📱 Android app like a Pokédex

An open-source application designed in the style of a Pokédex, allowing users to explore and catalog various creatures and features. This project aims to provide a fun and interactive way to engage with the Pokémon universe.

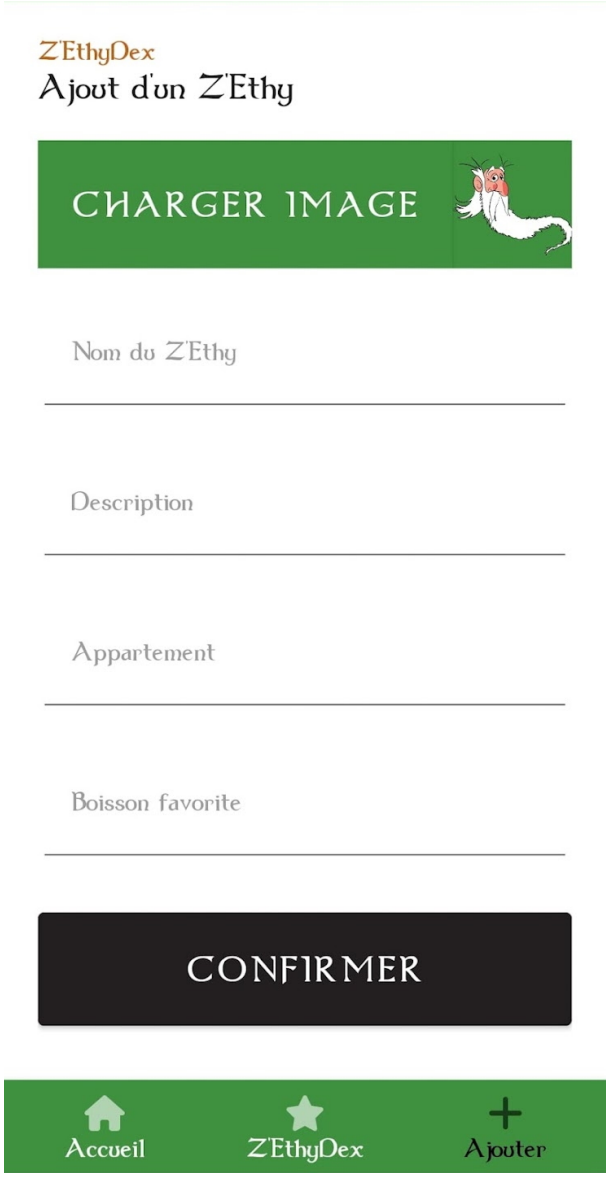
🎯 Purpose

- 📚 **Cataloging**: Create an extensive database of creatures with detailed information.
- 🎮 **Interactive Experience**: Offer a user-friendly interface for exploration.
- 🛠️ **Open Source Development**: Encourage collaboration and contributions from developers.

📝 Features

- 🌟 **User-Friendly Interface**: Easy navigation and search functionalities.
- 🔍 **Search Functionality**: Quickly find information on specific creatures.
- 📸 **Image Gallery**: Visual representation of each creature.

📷 Screenshots



learning

🌐 Jekyll site to archive my knowledge

An open-source website created using Jekyll and the “al-folio” theme, designed to archive my learning journey and share knowledge with anyone interested. This project serves as a personal portfolio and a resource hub.

🎯 Purpose

- 📖 **Knowledge Sharing**: Documenting and sharing what I’ve learned.
- 🌐 **Personal Portfolio**: Showcasing projects and skills.
- 🛠️ **Open Source Development**: Encouraging contributions and feedback from the community.

🌟 Features

- 📄 **Responsive Design**: Optimized for various devices.
- 🔍 **Search Functionality**: Easily find content and resources.
- 🛠️ **Customization Options**: Adapt the site to your preferences.

mpek29-latex-template

📄 Simple LaTeX template for ENIB

A beautiful, simple, and clean LaTeX template designed for ENIB students who want to write project reports. If you like the template, feel free to give it a star!

🎯 Purpose

- ✍️ **Streamlined Reporting**: Facilitate the creation of professional-looking project reports.
- 🎓 **Student Resource**: Tailored specifically for ENIB students' needs.
- 🛠️ **Open Source Development**: Encourage collaboration and customization from users.

📝 Getting Started

To use this template, I recommend following this tutorial.

After that, simply write your LaTeX code in the `body.tex` file.