Yens & Yens – Internship Technical Test Assignment

What’s the goal?
At Yens & Yens, we love working with curious minds who enjoy figuring things out and can independently explore new tools and technologies. This test assignment is designed to show us how you approach learning something new, how you structure a project, and how creative and self-sufficient you are. You're welcome to use Google, documentation, forums, or tools like ChatGPT — your process and problem-solving mindset are what we care most about.

We want to see how comfortable you are with exploring unfamiliar areas, combining hardware and software, and coming up with clever solutions. The goal is <u>not perfection</u>, but demonstrating resourcefulness and communication.

🧪 The Challenge: Motion-Driven Light with Pi Pico & PCB Hat

You’ll be designing a fun, interactive microcontroller project that combines:

1. **A Custom PCB Hat/Shield/Adapter for the Raspberry Pi Pico (This step is not mandatory, in case you are not familiar with designing hardware)**

    - Use **KiCad** to design a PCB that connects a **MPU-6050 motion sensor** to the **Pi Pico**.

    - The motion sensor should be placed on the PCB hat itself. However, if you prefer using a separate **MPU-6050 module**, your PCB can function as an **adapter** to connect the module properly to the Pi Pico, basically connecting all the pins of the sensor module with the pins of the Pi Pico.

    - The PCB design should be ready for upload and quoting on **JLCPCB** to check correctness — but you don’t need to actually order it.

2. **Understanding the MPU-6050 Motion Sensor**
   This sensor contains a 3-axis accelerometer and 3-axis gyroscope — it can detect tilt, motion, and rotation. You'll use this to influence light behaviour. If you think of another creative use for the sensor that you’d like to build instead, that’s fine too — as long as the setup and idea are clear and interesting.

3. **Interaction Design: Water-Inspired LED Animation**

    - Design a behavior where a **WS2812B LED strip** reacts to motion.

    - The LED strip starts with a **blue “water glow” in the middle**.

    - Tilting the sensor left/right makes the “water” flow that way — like gravity moving liquid.

    - Add fun touches like shimmer, flow delay, bounce, or edge reflections.
    - **If the LED strip isn’t connected**, the “water movement” should still be visualized in the **Serial Monitor** — using **ASCII characters** only (no indexes), for example: ____~~~~_____.

4. **Dual-Core Code on the Raspberry Pi Pico**
    - Use **C++ with the Arduino IDE**.
    - Upload your code via the **Arduino IDE**, and use its **Serial Monitor** for output.
    - **Core 0**: Reads motion data from the MPU-6050 (or simulated data if not found).
    - **Core 1**: Controls the LED strip and also prints the visual water movement via ASCII characters to the Serial Monitor.
5. **Robustness**
    - The code should run even **without the PCB or sensor connected** by falling back to simulated/dummy sensor values.
    - This ensures we can test your code using just our Pi Pico and Arduino IDE Serial Monitor.
6. **Serial Monitoring**
    - Show clear **motion sensor values** and the corresponding **ASCII-based visual of the water strip**.
    - Ensure everything prints on **one line per frame**, so it's easy to follow in the Serial Monitor without flickering or jumping lines.
7. **Hardware Notes**
    - The designing of the PCB on KiCad is not mandatory, only for interns who want to challenge themselves.
    - You don’t need to physically build the PCB — we’ll just check the design (if made).
    - You can use **JLCPCB** to quote your PCB to verify that your layout is correct (if made).
    - A **Pi Pico** costs only a few euros. You're not required to have one, but it might be fun to order one to test your code. Even without the sensor, it can run with dummy input.
8. **Timeline Flexibility**

- Since ordering and testing can take time, we **let you propose your own deadline**.
- Just let us know what works for your schedule.
- In the week of the 1st of December, we will start reviewing the submitted tests.

☑ What to send us

1. Your **KiCad project folder** (schematic + PCB files, if you choose to make it)
2. Your **C++ .ino code for the Raspberry Pi Pico, ready to be uploaded with Arduino IDE**
3. A short **README** with:
    - What the project does
    - How to test it (with or without hardware)
    - How each core is used
    - Sample Serial Monitor output
    - (Optional) Screenshot of PCB layout or 3D view

We’re excited to see how you approach this — and we hope you enjoy building something creative and interactive. Good luck!

— Yens & Yens