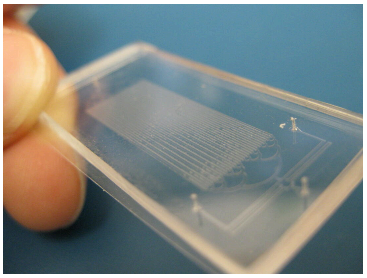
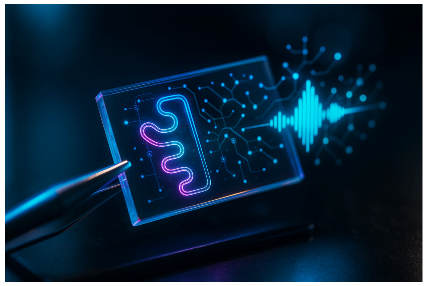
Microfluidic Molecular Communication (MMC) with AI-Assisted Encoding/Decoding

Supervisor: *Pit Hofmann, Juan Cabrera*

Microfluidic channels are compelling communication media for lab-on-chip and constrained environments, such as in-body communication, but they offer extremely low data rates and highly non-Gaussian noise due to diffusion, advection, and cross-talk between chemicals. This project explores **AI-assisted, function/semantics-aware communication** over our microfluidic testbed: Rather than fighting the channel to send raw bits, we co-design the source, channel, and task so that intelligent agents assist both **encoder** and **decoder**.

A concrete toy example system will be built end-to-end: an **"audio-over-fluid"** link where the transmitter listens to speech, converts it to text, communicates that text via chemicals in the microfluidic channel, and the receiver reconstructs intelligible audio using an AI text-to-speech decoder. The baseline will be extended with rate-distortion controls (e.g., summaries, punctuation-aware coding, unequal error protection, grammar-constrained symbols) and language-model-aided error correction.

What you will do

- Implement a baseline pipeline: **Microphone → ASR (speech-to-text) → text encoder → microfluidic modulator → channel → detector/decoder → TTS (text-to-speech)**.
- Design symbol alphabets and release schedules for the microfluidic modulator; implement detection and timing recovery on the receiver.
- Add **AI agents** for compression and error mitigation.
- Evaluate with rigorous metrics: channel BER/SER, **word error rate (WER)**, latency, throughput (symbols/s, effective bit/s), and audio quality proxies (e.g., MOS/PESQ or intelligibility tests).
- Compare strategies: raw character vs. byte-pair tokens; fixed vs. adaptive codebooks; repetition/CRC vs. fountain codes; semantic summarization vs. verbatim.
- Run controlled experiments on the **microfluidic testbed** and provide an engineering analysis (noise models, flow-rate dependency, intersymbol interference, stability/cleaning protocol).

Why this matters

If successful, the system demonstrates **task-oriented, semantics-aware communication** in a domain where classic bit-pipe assumptions break down. Results can generalize to low-rate or high-latency channels (bio-hybrid links, environmental sensing, in-body comms).

Required skills

- **Python** (must), basic command of Linux tooling.
- Familiarity with **digital communications/signal processing** (modulation, detection, synchronization) or a strong willingness to learn fast.
- Experience with at least one **ML framework** (PyTorch/TF) for ASR/TTS or language models.