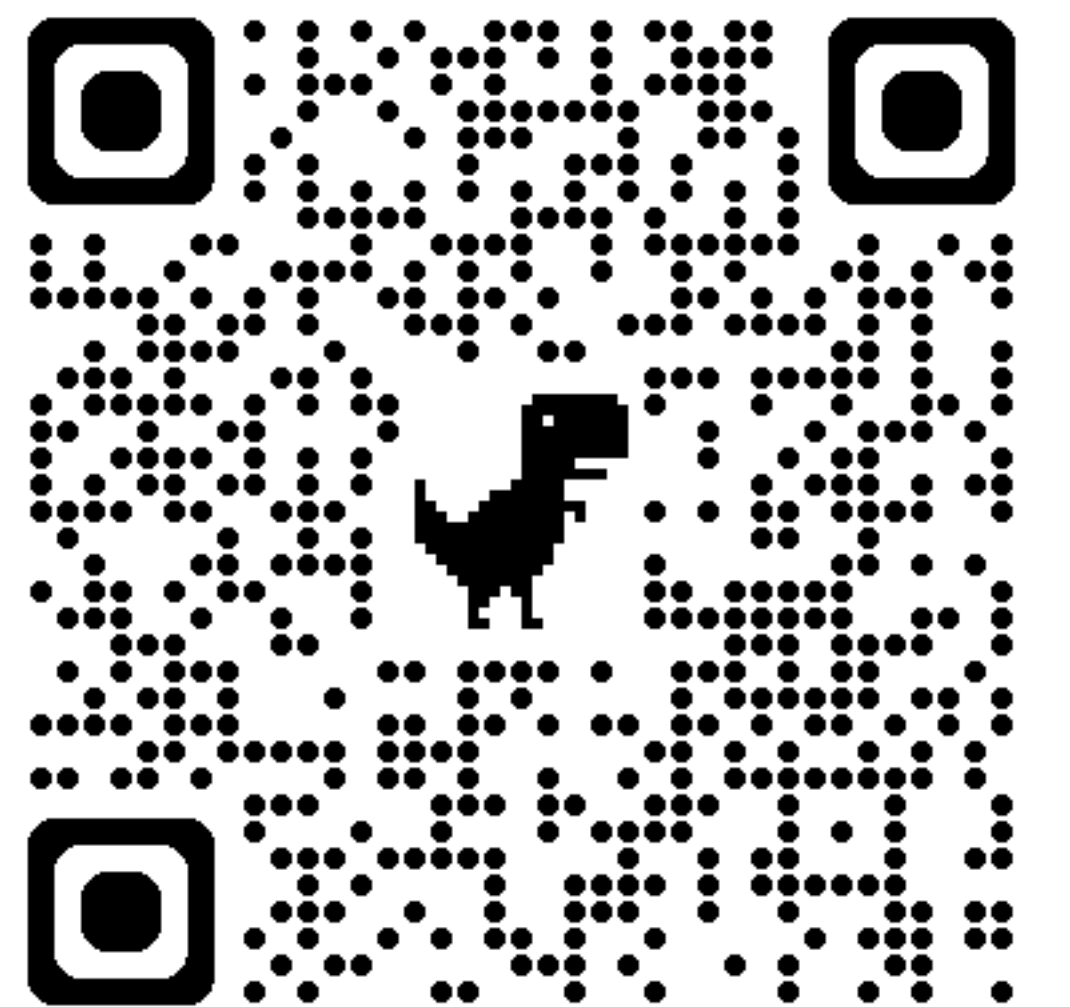


Intro to Physical Computing

Jeff Feddersen, ITP/NYU



Class 1 Agenda

- Welcome!
- Class overview
 - Big picture
 - Course logistics
- Your intros
- Intro fantasy device project
- Quick workshop tour
- [Break]
- Fantasy device group project
- Distribute parts
- Connect Arduino (if we have time)

2024 Pcomp with Jeff!



2023 Pcomp with Jeff!



2022 Pcomp with Jeff!



Why?

(required class... but why else?)

Purpose

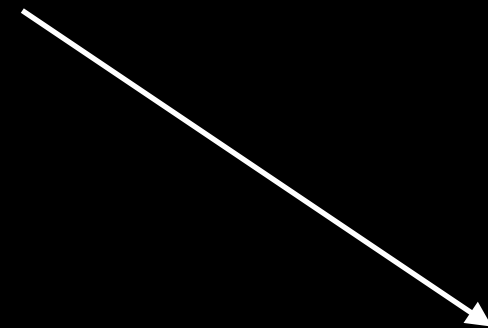
Support/Expand/Enable creative expression...

...with technology...

...focused on physical, embodied interaction.

Method

“...with technology”



Programming Microcontrollers and Circuits

to be Cool*

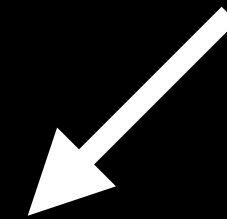
“Enable creative expression...”



“...focused on physical, embodied interaction.”

Circuits

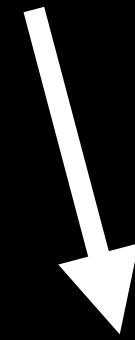
Programming Microcontrollers and **Circuits** to be Cool*



- Electrical properties (voltage, resistance)
- Basic components and simple circuits
- Breadboard prototyping
- Testing and measuring
- Handling higher power

Microcontrollers

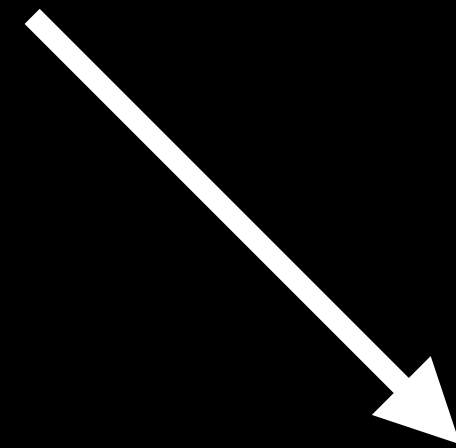
Programming **Microcontrollers** and Circuits to be Cool*



- Small cheap single-task computers
- Can sense and output voltages
- Fast (compared to us)

Programming

Programming Microcontrollers and Circuits to be Cool*



- Simple Development Tools
- Programming basics
 - Pseudocode
 - Variables, memory
 - Structure, control flow

to be Cool*

Programming Microcontrollers and Circuits **to be Cool***

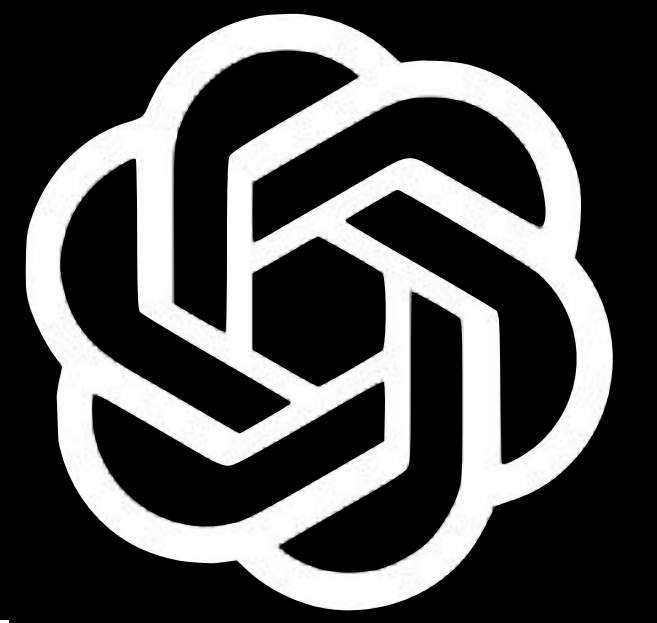
to be Cool*

Programming Microcontrollers and Circuits **to be Cool***

*Up to you



Voice



AI



Metaverse

Bret Victor's
“Pictures under glass”

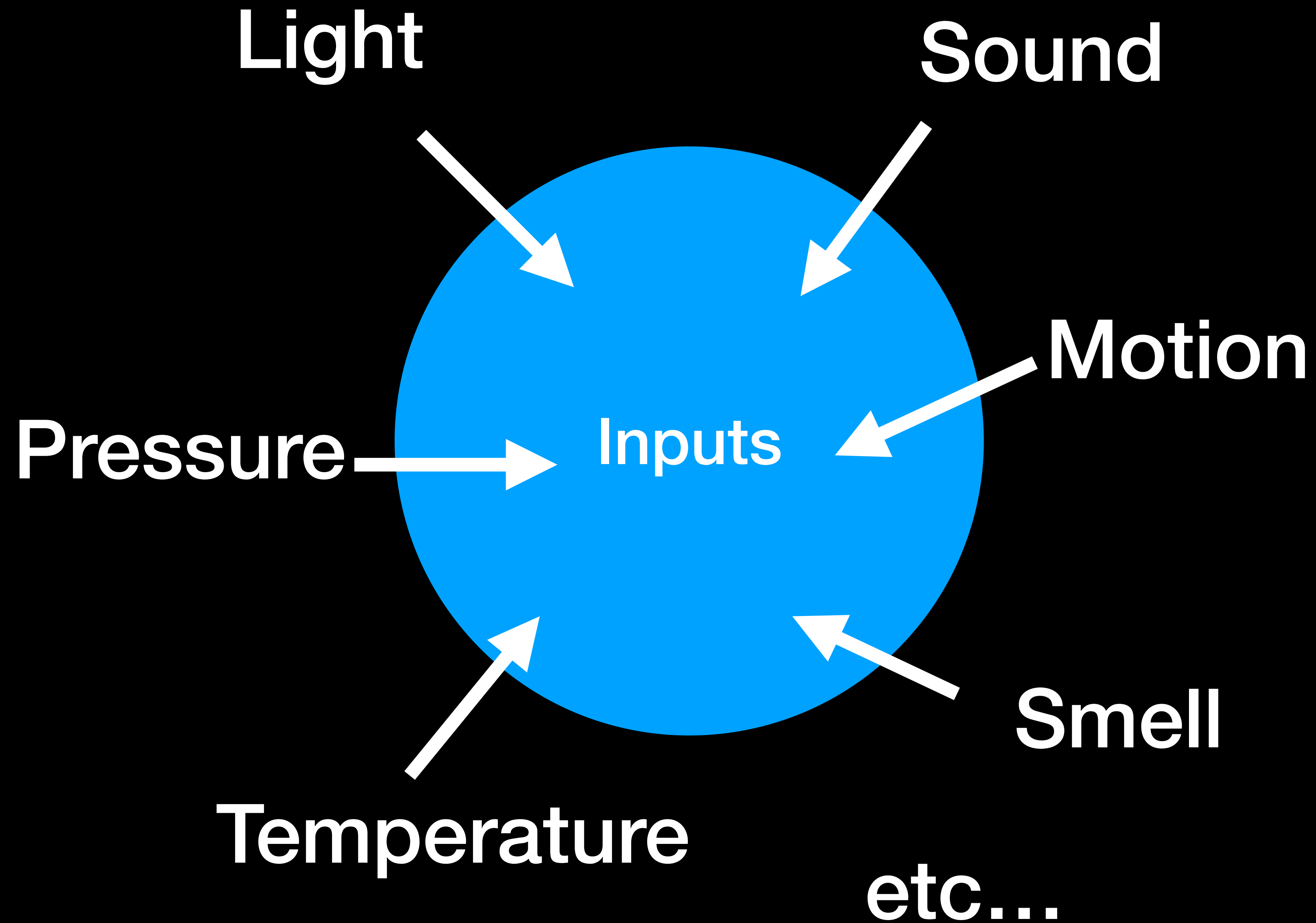


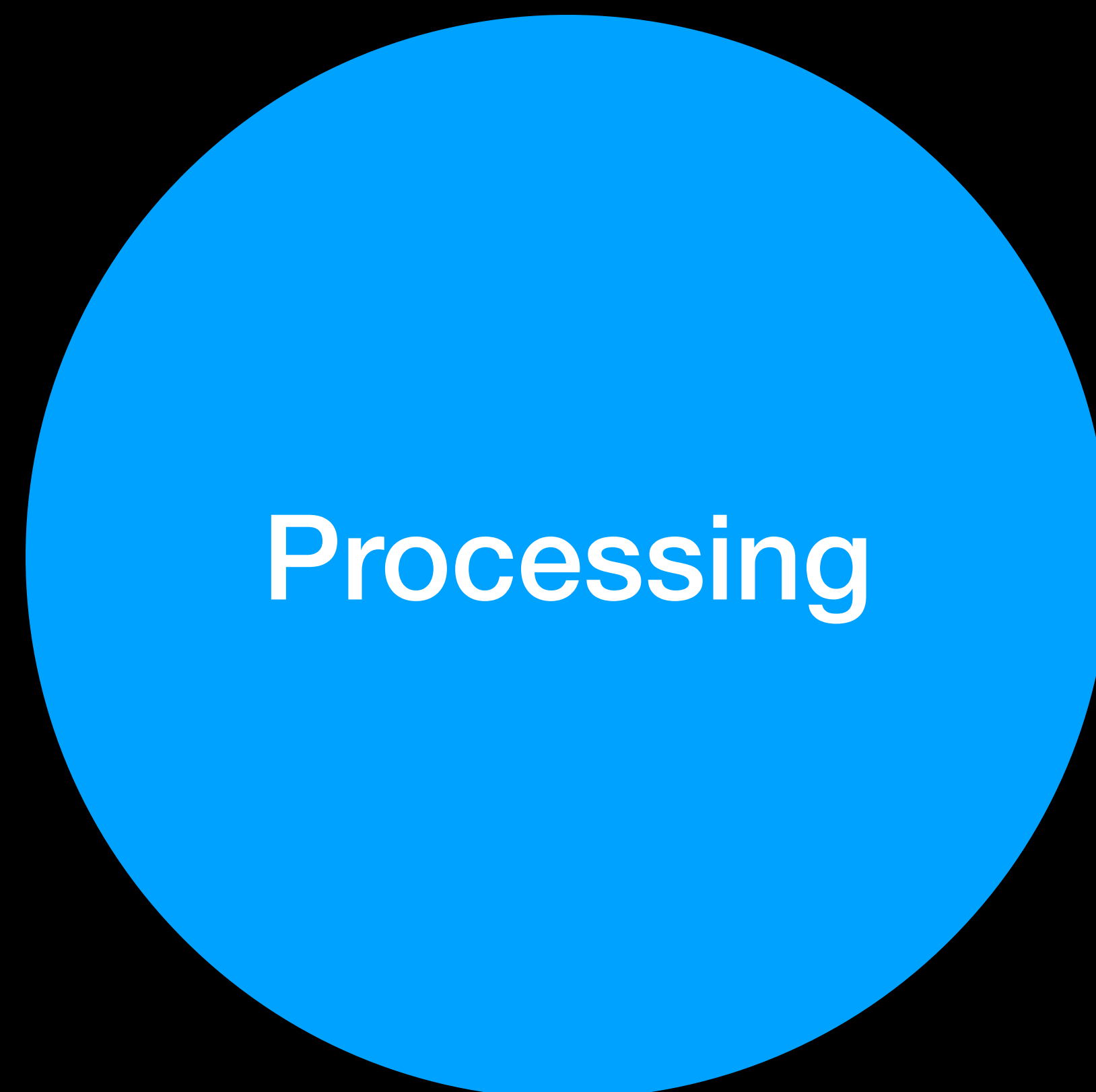


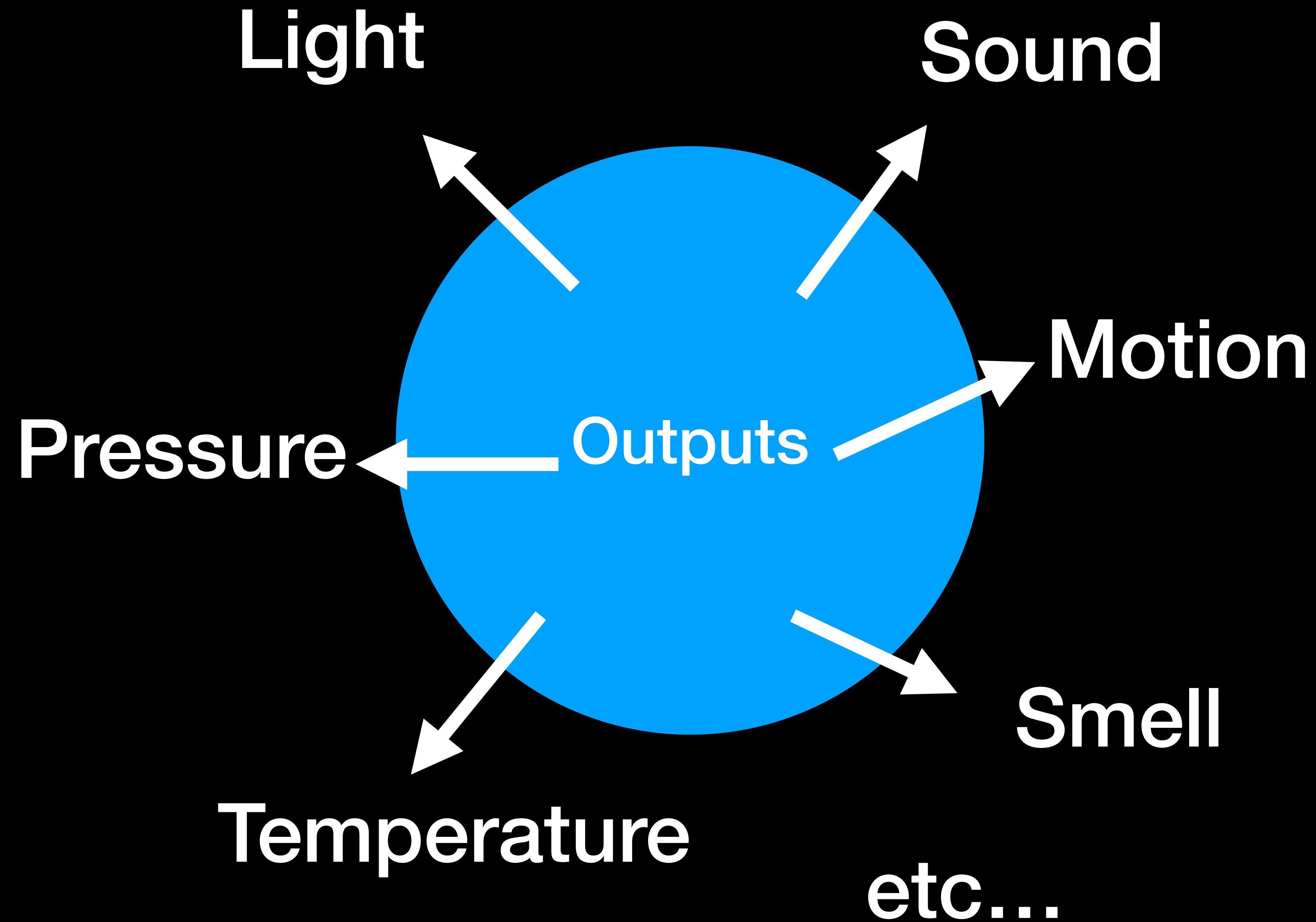


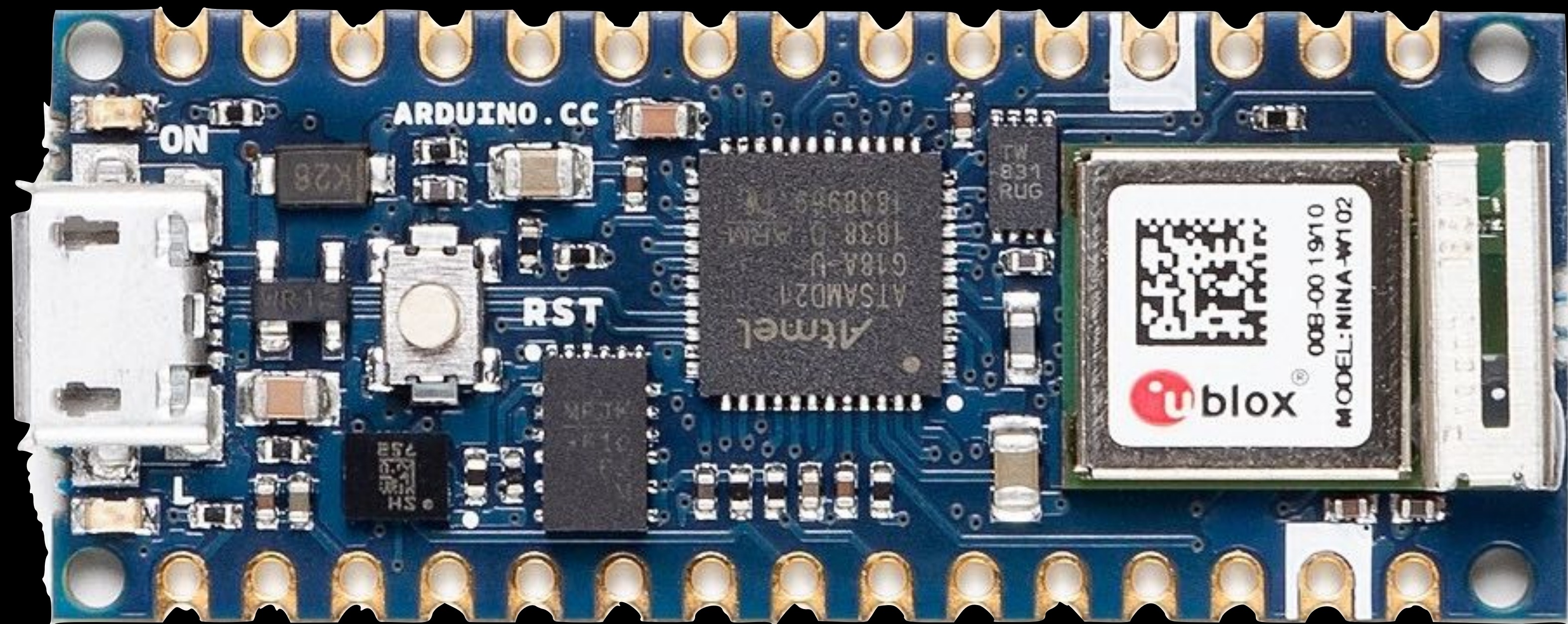


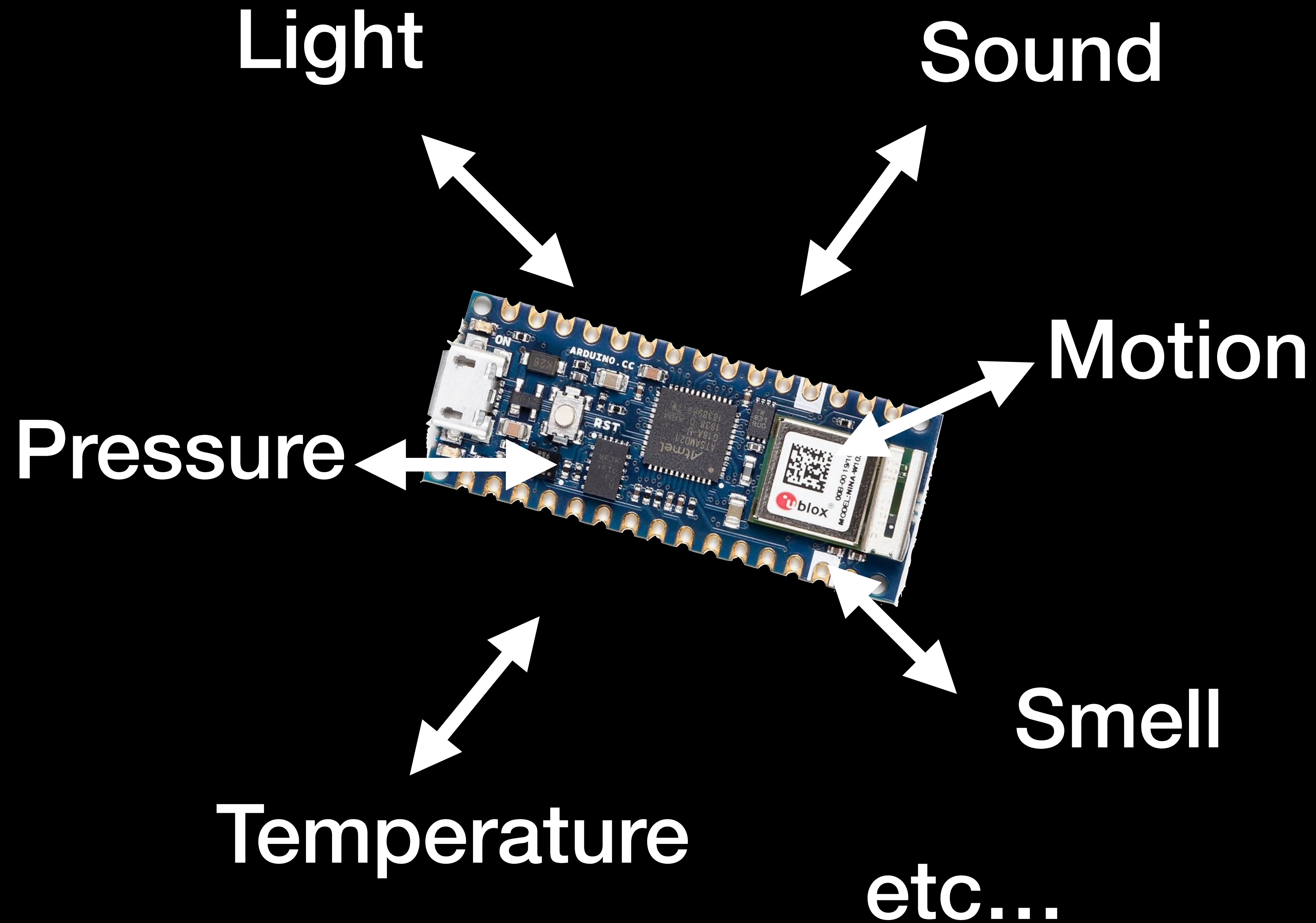


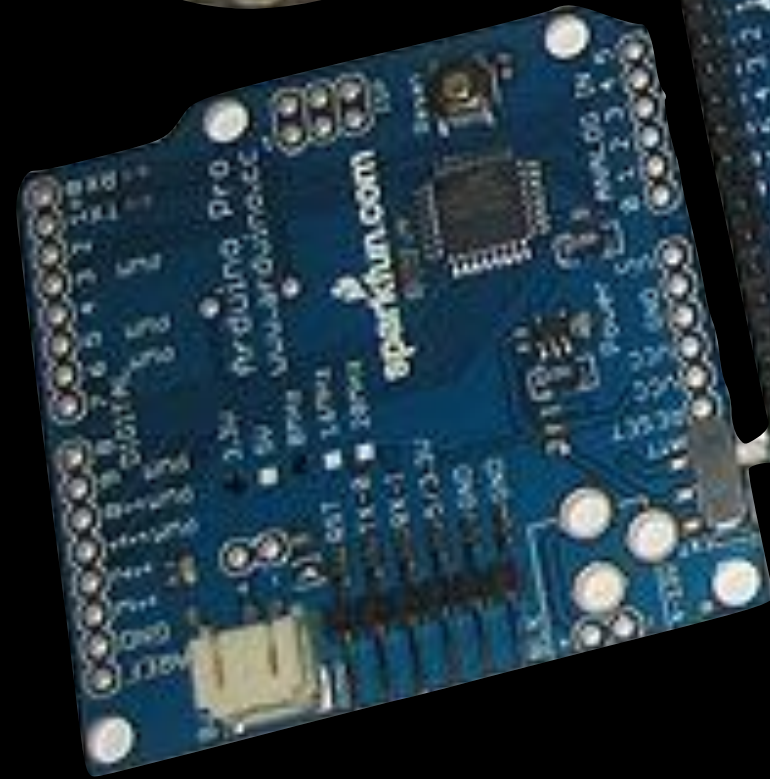
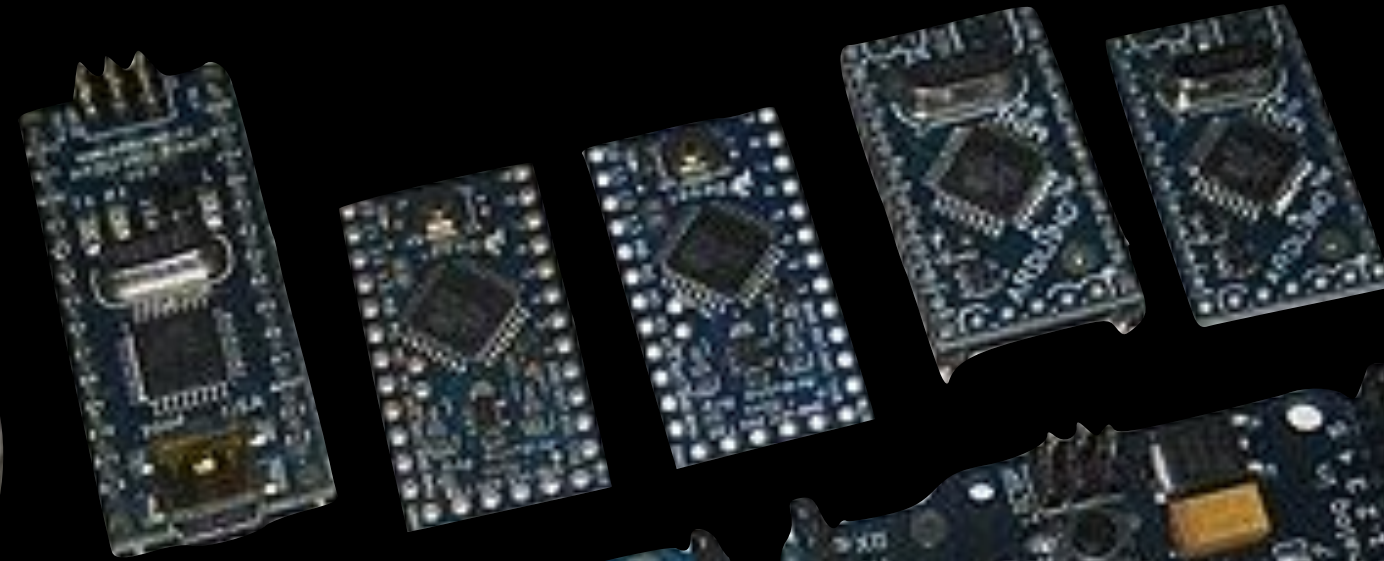
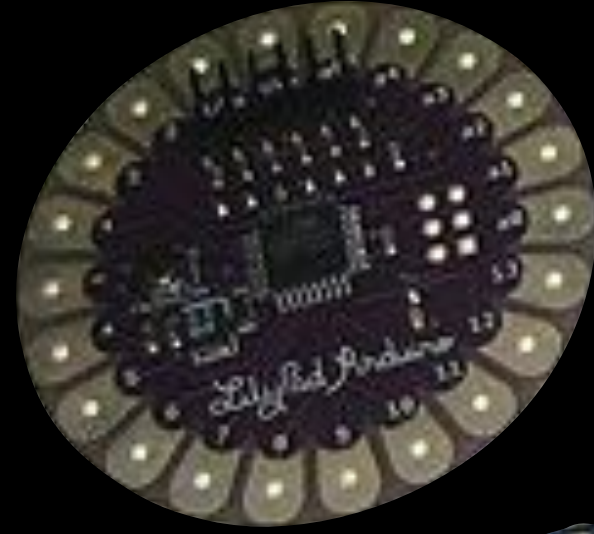
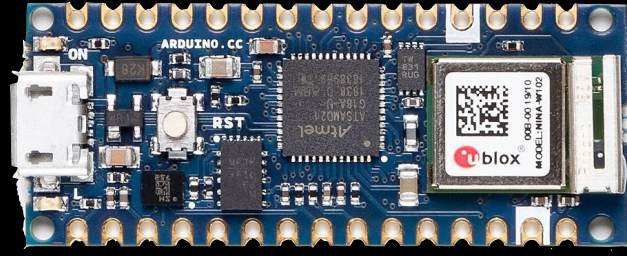


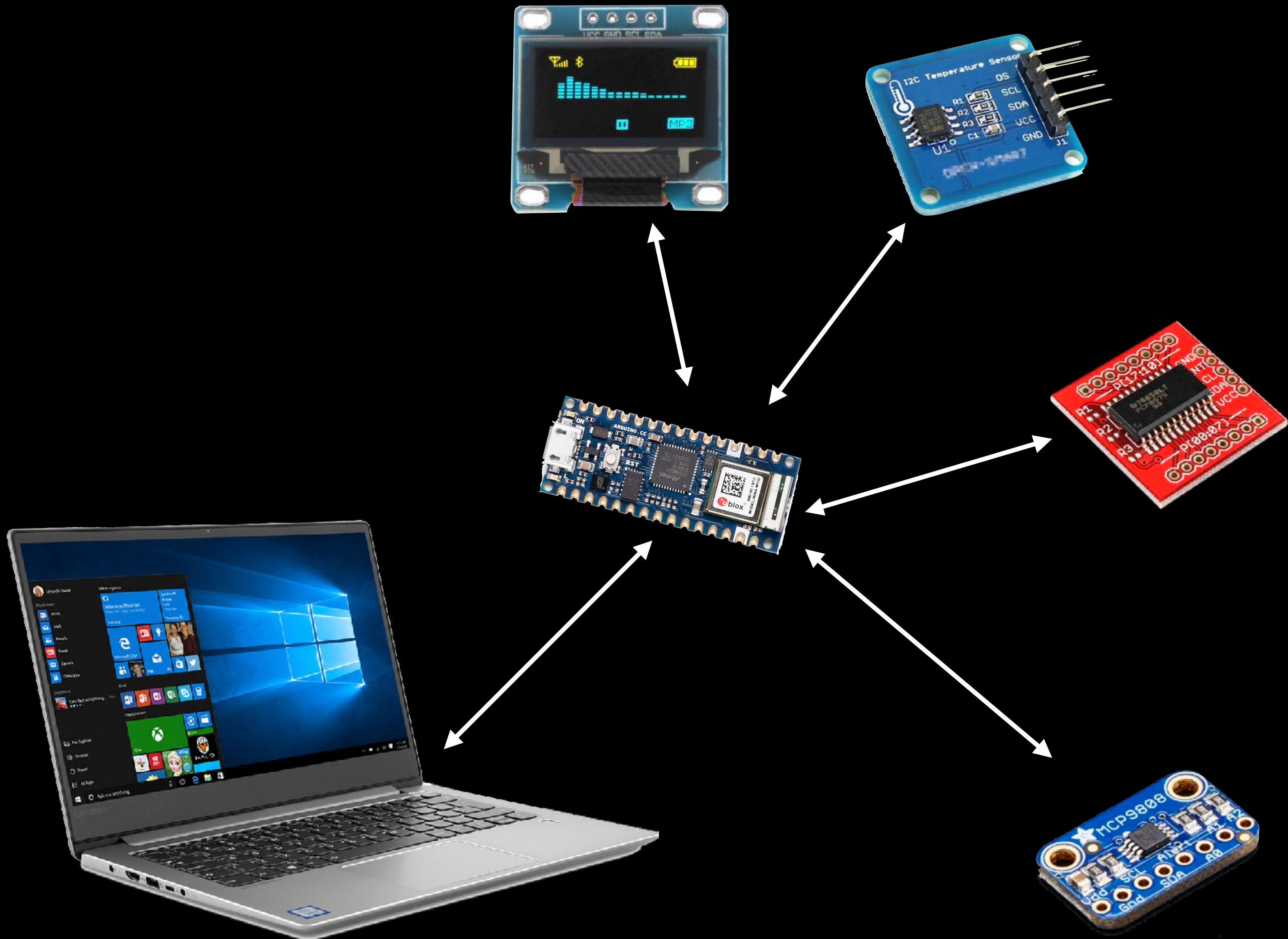




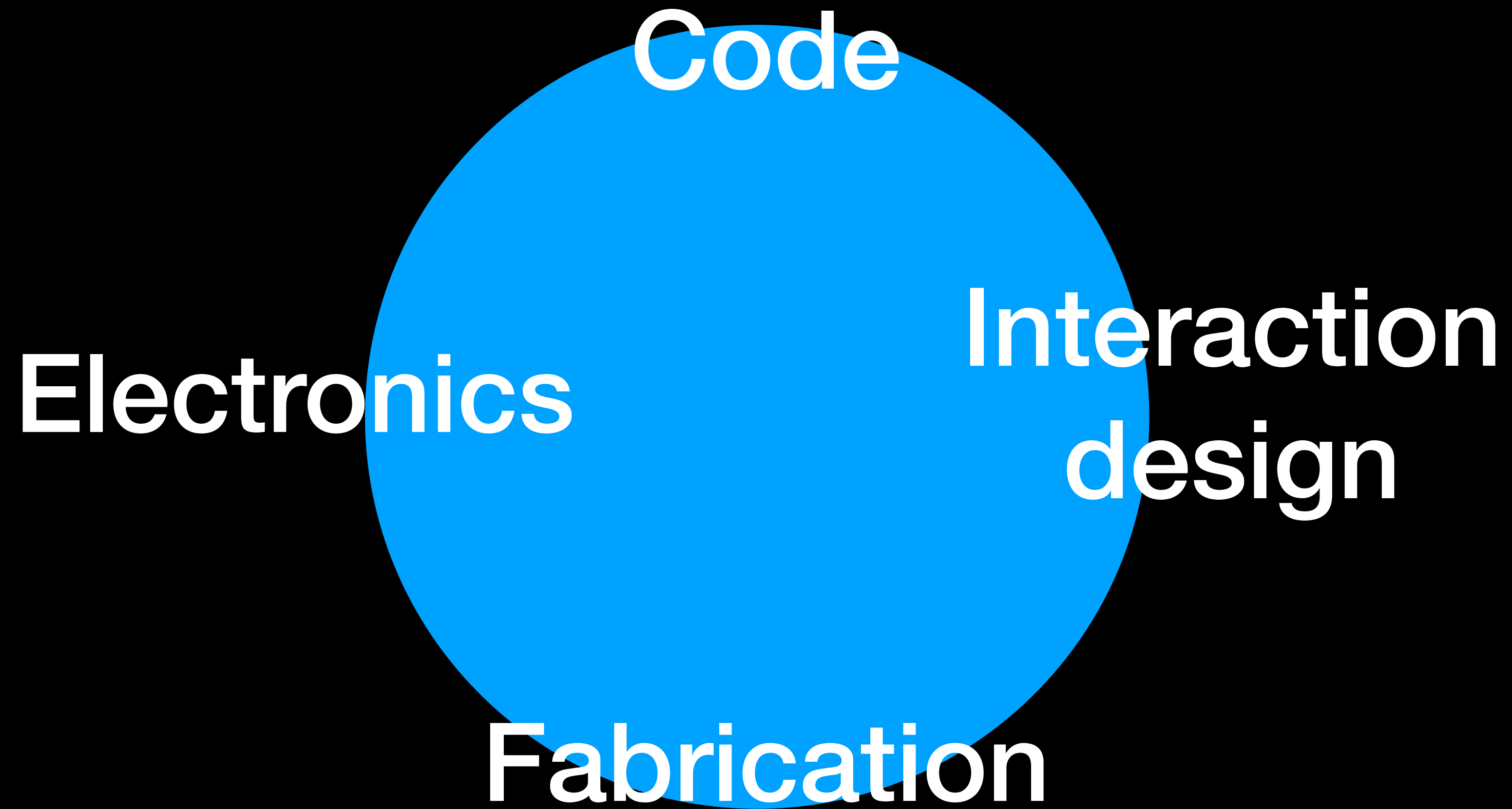




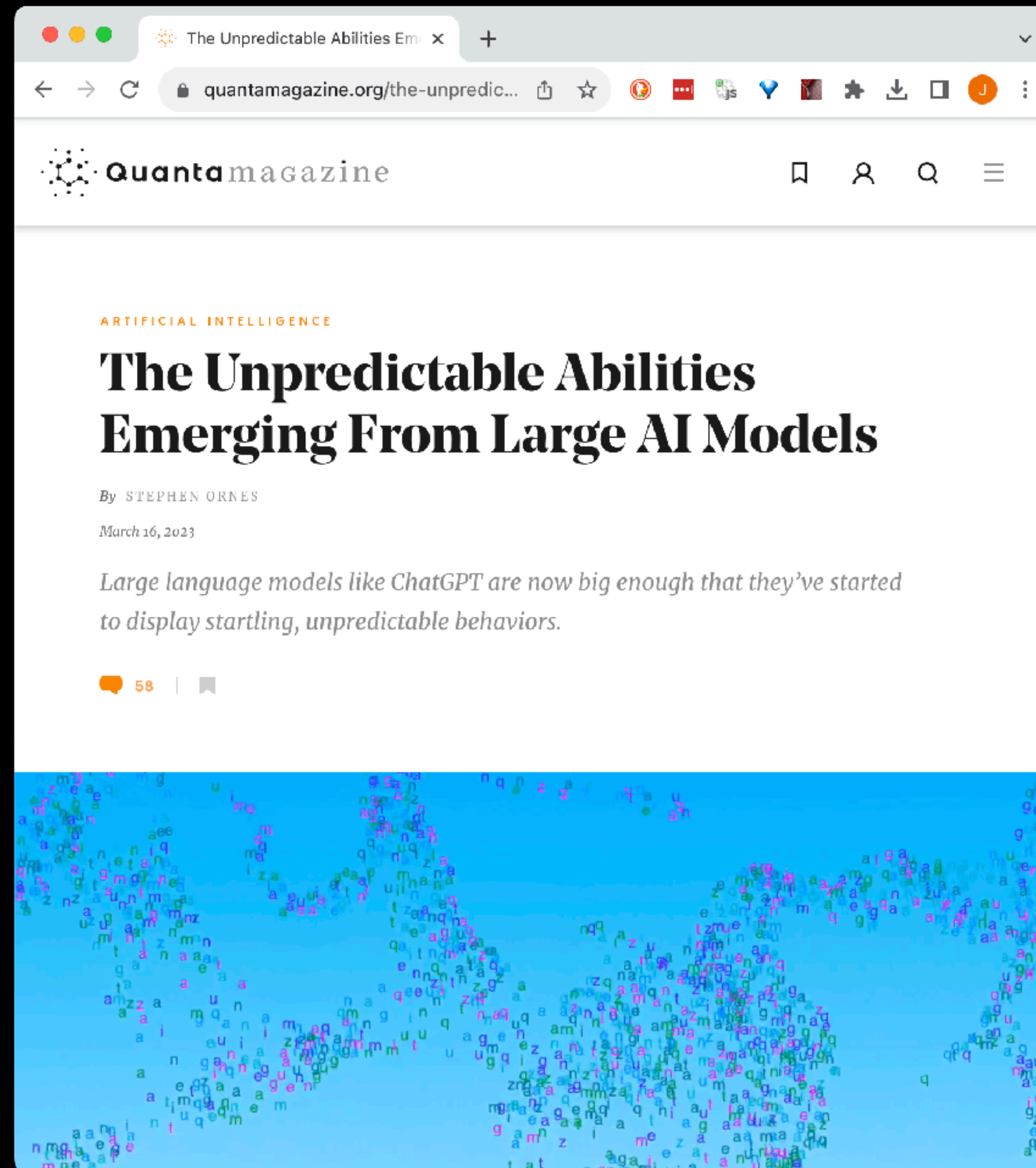




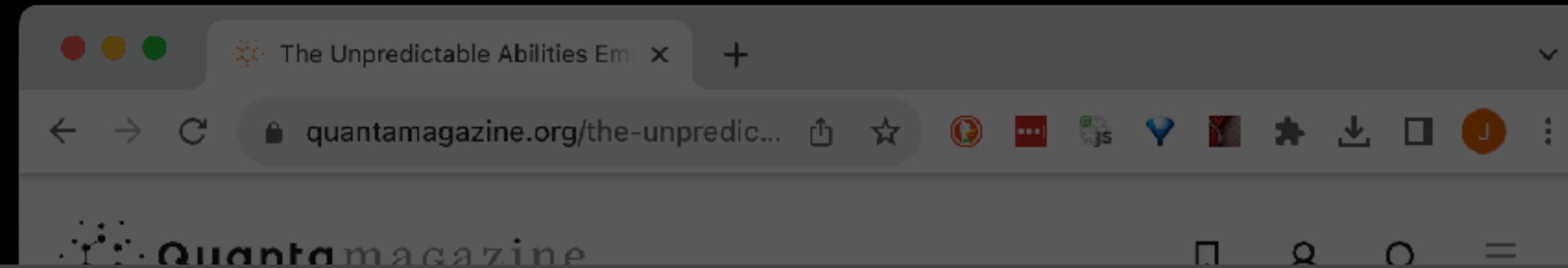
- Methods for handling larger projects
- Fabrication
- Play testing and User Testing
- Presenting and documenting



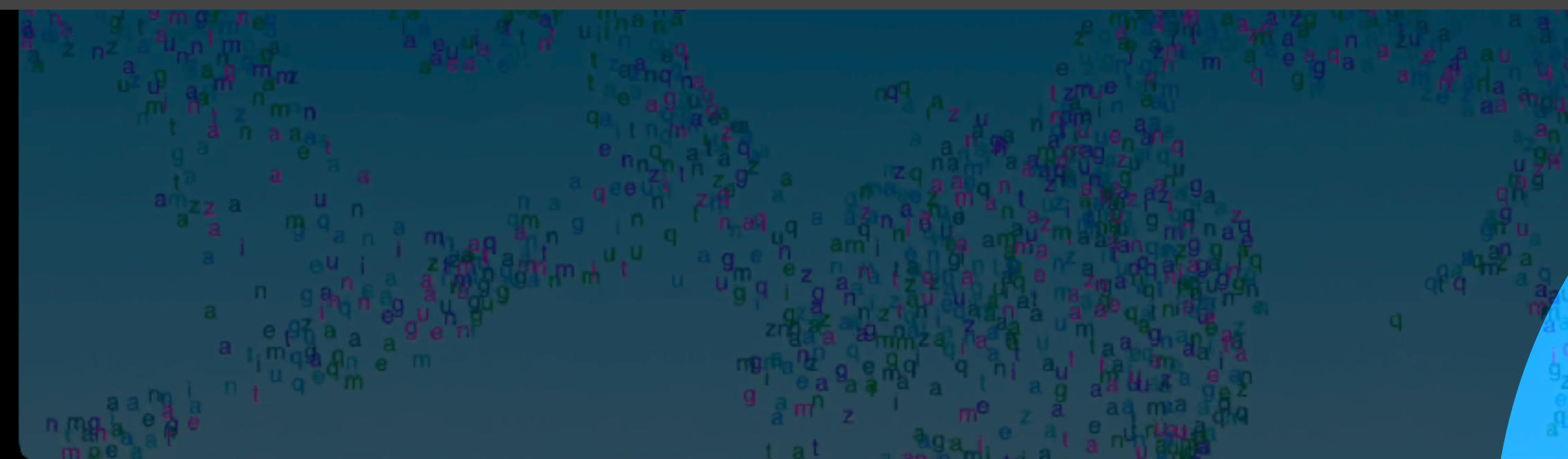
Code

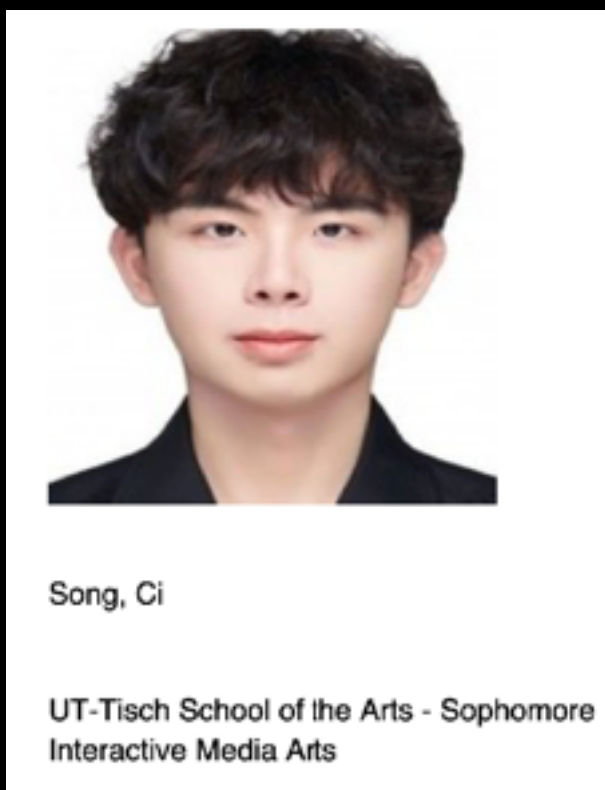


Code



Recent investigations like the one Dyer worked on have revealed that LLMs can produce hundreds of “emergent” abilities — tasks that big models can complete that smaller models can’t, many of which seem to have little to do with analyzing text. They range from multiplication to **generating executable computer code** to, apparently, decoding movies based on emojis. New analyses suggest that for some tasks and some models, there’s a threshold of complexity beyond which the functionality of the model skyrockets. (They also suggest a dark flip side: As they increase in complexity, some models reveal new biases and inaccuracies in their responses.)





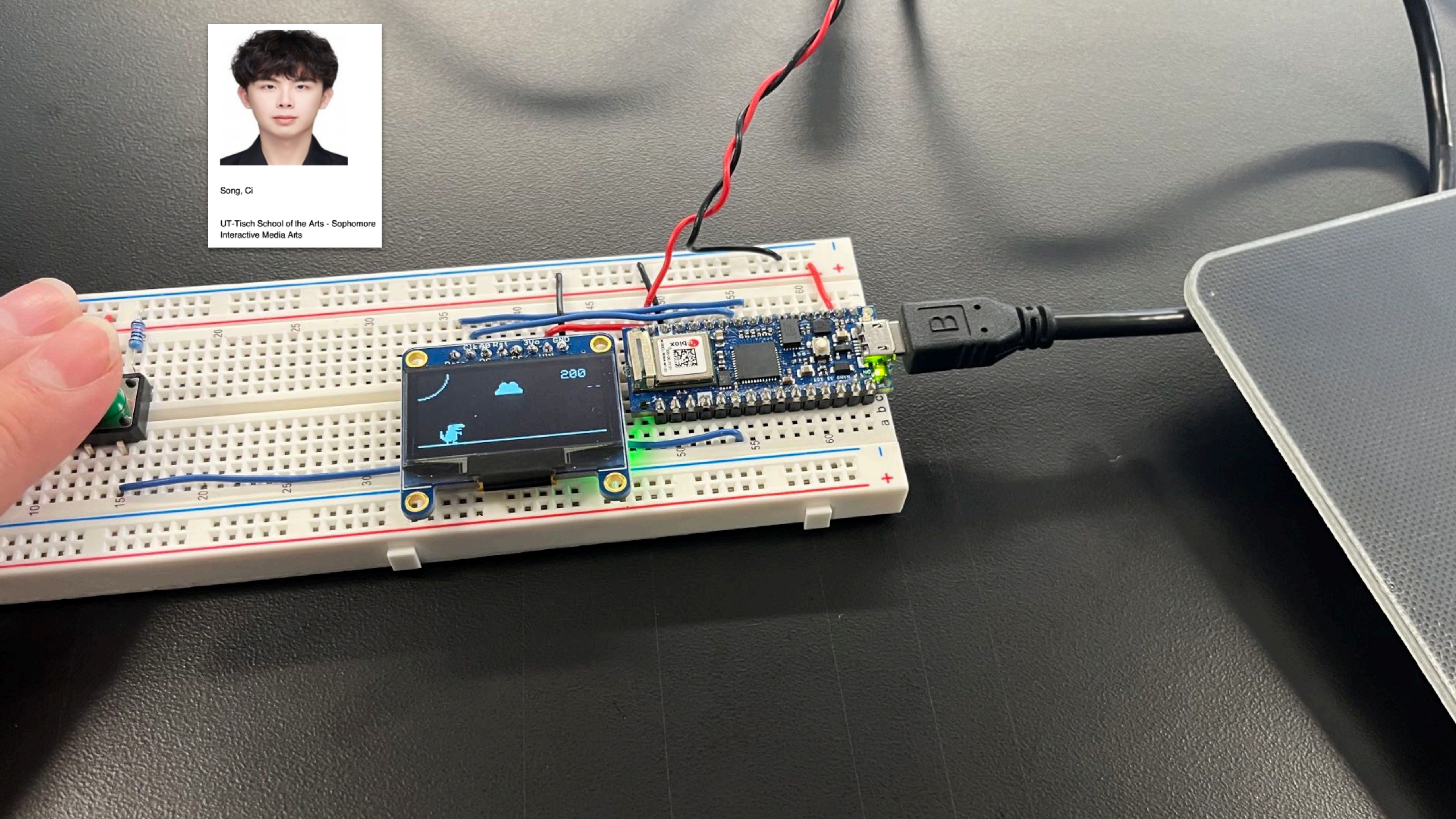
Ci was an undergraduate student in my 2022 IMA Energy course.

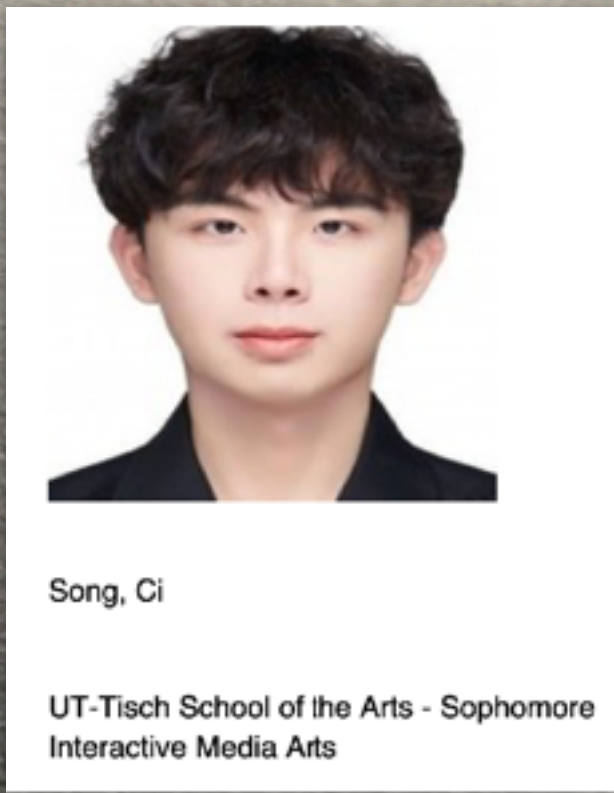
He had a little pomp experience, and wanted to make a solar-powered video game for the class final.



Song, Ci

UT-Tisch School of the Arts - Sophomore
Interactive Media Arts





Here is my chat history with ChatGPT4 re: Arduino Dinosaur Game

User: how to make a Dinosaur Game in Arduino and display it on Monochrome 1.3" 128x64 OLED Graphic Display with a bottom to play the game

ChatGPT: *To create a simple dinosaur game on an Arduino and display it on a 1.3" 128x64 OLED Graphic Display, you'll need the following components:*

1. *Arduino board (e.g., Arduino Uno or Nano)*
2. *Monochrome 1.3" 128x64 OLED Graphic Display (e.g., SSD1306-based)*
3. *A pushbutton*
4. *Breadboard and jumper wires*

...

Feel free to modify the code to improve the game or add extra features like different obstacles, variable jump heights, or a high score feature.

User: could you make the cactus more randomly

ChatGPT: *Sure! You can use the `random()` function to generate random distances between the cacti. Here's an updated version of the code that introduces random spacing between cacti:*

```
```cpp
#include <Wire.h>
...
```

**User:** why is the shape of the cloud look like a heart

**ChatGPT:** *You're right, the cloud shapes in the code might resemble a heart due to the position of the circles used to create the cloud. We can adjust the cloud shape to make it look more like a traditional cloud by adding more circles and changing their positions. Here's the updated `displayClouds()` function with a more cloud-like shape:*

```
```cpp
...
```

Excerpt of about 50
pages of interaction Ci
reported

processing of the IMU's data. I would like to be able to detect when the arduino is tapped, probably by looking at a sudden spike (positive or negative) in the z axis accel data. The Arduino should add a seventh parameter, "tap" that is either 0 (no tap) or 1 (tap occurring). It should be debounced, so once it is triggered, it does not

should be able to detect taps as close as 250ms apart. After that is implemented, we'll work on integrating the tap gesture into the game.

Thought for 54s

Awesome—let's add tap detection on the Arduino side and append it as a 7th CSV field (`tap = 0/1`). This version:

- Samples IMU at ~100 Hz
- Uses a simple high-pass (LPF baseline) on Z accel to spot spikes (positive or negative)
- Debounces with a short **hold** (so you don't get flicker)
- Enforces a **refractory** period so you can still detect taps as close as **250 ms** apart
- Prints: `ax,ay,az,gx,gy,gz,tap`

If you want to keep your current p5 sketch running (it expects 6 fields), temporarily set `OUTPUT_TAP` to `0` below; we'll switch it back to `1` when we update p5.



+ Ask anything



JF

ChatGPT can make mistakes. Check important info.

Tilt Marble P5 (script)

```
1 // Marble-on-a-tilt-table using @gohai/p5.webserial v1
2 // Arduino sends CSV lines: ax,ay,az,gx,gy,gz[,tap] @ 115200
3 // Keys: C = calibrate neutral tilt, R = reset marble
4
5 // ---- Serial setup (v1 API) ----
6 let port;
7 const BAUD = 115200;
8 let connectBtn;
9 const NL = String.fromCharCode(10); // copy/paste-safe newline for readUntil()
10
11 // ---- IMU state (smoothed) ----
12 let ax = 0, ay = 0, az = 9.81; // m/s^2 (accelerometer)
13 let gx = 0, gy = 0, gz = 0; // deg/s (gyro)
14 const accLPF = { x: 0, y: 0, z: 9.81 }; // low-pass accelerometer
15 const accAlpha = 0.15; // smoothing factor
```

Connect

Tilt-Table Marble — IMU → p5

Status: Disconnected

Last data: —s ago

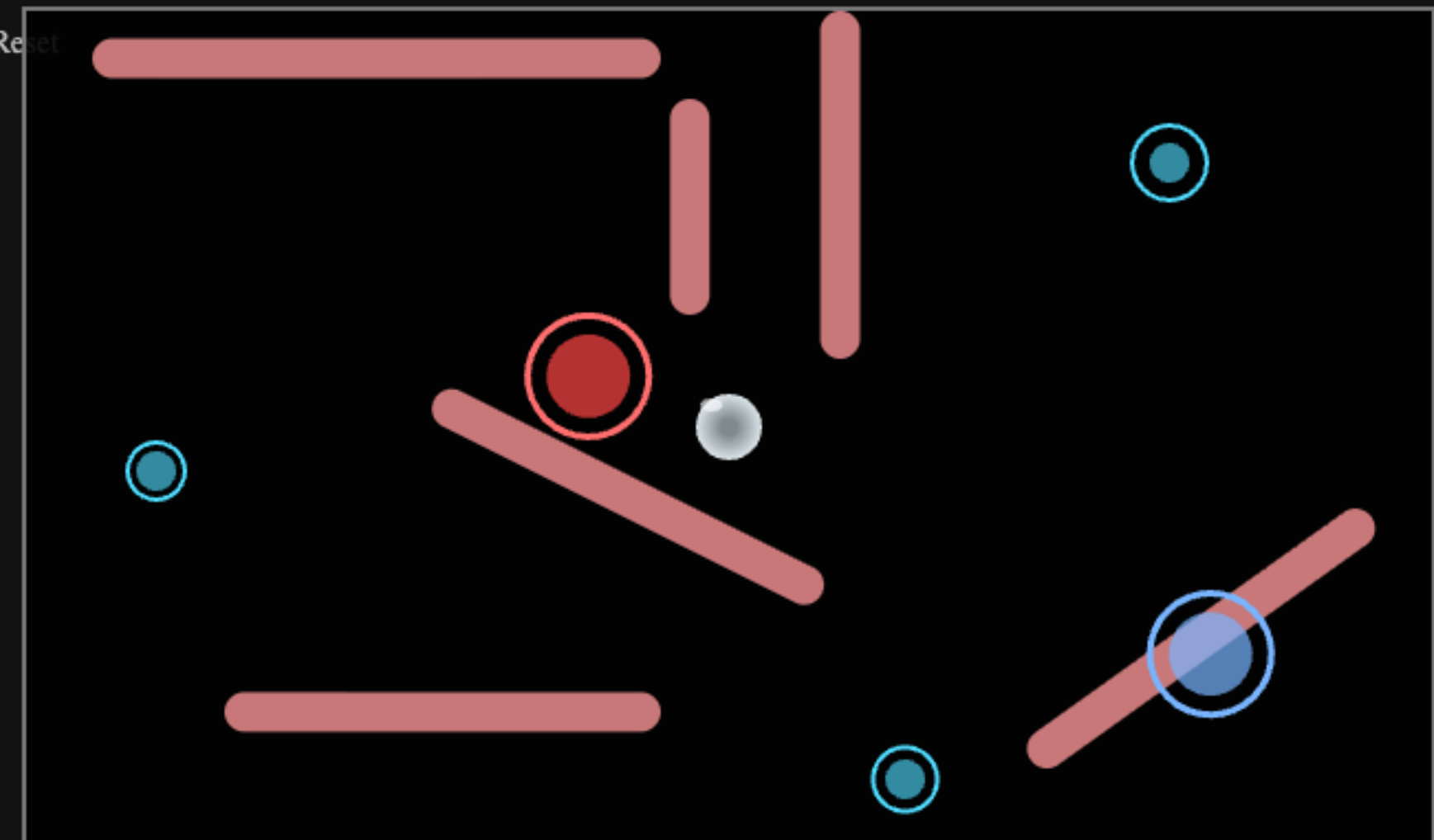
Roll (deg): 0.0

Pitch (deg): 0.0

C: Calibrate R: Reset

Score: 0 / 1000

Shields: 2



This sounds great. Now add a B section that alternates with the current pattern every four measures

Got it—let's add a **B section** and make the sequencer alternate:

- Play your current (A section) patterns for **4 measures**
- Switch to **B section** patterns for 4 measures
- Then back to A, and so on.

We'll just add an alternate set of arrays (`melodyPatternB`, `bassPatternB`, `percPatternB`) and track which section we're in. The sequencer already advances by steps, so we just need a **measure counter**. Since each measure = 16 steps / 4 beats (at 4 ticks per beat), we can count measures easily.

Here's the modified sketch excerpt (only the new/changed parts shown for clarity):

cpp

Copy code

```
// ===== Sequencer =====
#define STEPS 16

// Section A patterns
const int8_t melodyPatternA[STEPS] = {
  72, 76, 79, 76, 74, 77, 81, 77,
  79, 76, 74, 72, 71, 72, 74, 76
};
const int8_t bassPatternA[STEPS] = {
  48, 48, 36, 36, 43, 43, 41, 41
  48, 48, 36, 36, 43, 43, 41, 41
}
```

+ Ask anything



ChatGPT can make mistakes. Check important info.

```
1  /*
2   Arduino Nano 33 IoT (SAM21G18) – 3-voice chiptune with A/B sections,
3   tonal kick + noise snare, and LED light show on D10..D13.
4
5   Upgrades vs Uno:
6   • Uses the on-chip DAC on A0 for cleaner audio (10-bit) instead of PWM.
7   • Higher sample rate: 22.05 kHz via TC5 interrupt for better highs/less aliasing.
8   • Optional DC-blocking high-pass filter (HPF) in mixer, toggle with USE_HPF.
9   • No external libraries – direct register setup for TC5 + DAC.
10
11  Wiring
12   • Audio: A0 (DAC) → 1 kΩ series → 10 μF to GND (RC low-pass) → amplifier/speaker.
13     (Tip: add 100 Ω in series if driving small piezo directly.)
14   • LEDs: D10 kick, D11 snare, D12 bass note, D13 beat pulse (with 220 Ω series resistor)
15  */
16
17  #include <Arduino.h>
18
19  // ===== Platform: SAMD21 (Nano 33 IoT) =====
20  // Sample rate and tempo
21  static const uint32_t SAMPLE_RATE = 22050; // Hz (TC5 interrupt)
22  static const uint8_t TICKS_PER_BEAT = 4; // 16th notes
23  static uint16_t BPM = 120; // tempo (feel free to change)
24  volatile uint32_t samplesPerTick; // computed from BPM
25  volatile uint32_t tickSampleCounter = 0;
26
27  // Toggle high-pass filter
28  #define USE_HPF 1
29
30  // Mix levels (0..255). Keep headroom to avoid clipping.
31  static const uint8_t VOL_MELODY = 96;
32  static const uint8_t VOL_BASS = 72;
33  static const uint8_t VOL_NOISE = 64;
34  static const uint8_t VOL_KICK = 96;
35
36  // ===== Tone osc state =====
37  typedef struct {
38    uint32_t phase;
39    uint32_t incr;
40    uint8_t vol;
41    uint8_t targetVol;
42    uint16_t env;
43    uint16_t envMax;
44    bool gate;
45  } ToneOsc;
```

[>]

How about the pitch?

Show thinking

That's another great question about the fundamentals of how this works.

Similar to the speed, the **pitch is also "baked into" the LPC data for each word**. The library doesn't have a simple function like `voice.setPitch()` to change it.

The LPC data for a word like "ALPHA" contains information that essentially says, "for the next few milliseconds, create a buzzing sound at *this specific frequency* and pass it through a filter shaped like *this*." The library's job is to follow that recipe exactly.

To get a different pitch, you would need to re-encode the original `.wav` files using the LPC tools and change the pitch of the source audio before you create the data. This would give you a completely new set of voice data (e.g., a high-pitched vocabulary and a low-pitched one).

What do you want to build?

Canvas

Your New York University chats aren't used to improve our models. Gemini can make mistakes, so double-check it. [Your privacy & Gemini](#)

Arduino LPC Speech Synthesis

```
1  /*
2   * NATO Phonetic Alphabet & Number Announcer using real-time LPC Speech Synthesis
3   * for Arduino Nano 33 IoT.
4   *
5   * This program uses the "Talkie" library to say a random word from
6   * the NATO phonetic alphabet or a number from 0-9.
7   *
8   * --- LIBRARY ---
9   * You MUST install the "Talkie" library by Peter Knight.
10  * Go to Sketch > Include Library > Manage Libraries... and search for "Talkie".
11  *
12  * --- HARDWARE ---
13  * 1. An audio amplifier (like an LM386 module) and a speaker.
14  * 2. Connect the amplifier's input to Arduino pin 6.
15  * 3.
16  * 4.
17  */
18
19 #include <Arduino.h>
20 #include <Talkie.h>
21
22 // Constants
23 Talkie talkie;
24
25 // Pin definitions
26 // Arduino pin 6
27 // Talkie module pin 1
28 const int PIN_6 = 6;
29 const int PIN_1 = 1;
30
31 // Variables
32 sp
33 sp
34 sp
35 //
36 sp
37 sp
```

Filter by

- <> Code 6
- Issues 2
- Pull requests 0
- Discussions 0
- Commits 1
- Packages 0
- Wikis 0

Languages

- Markdown
- C++
- C
- More languages...

Advanced

Owner

6 files (108 ms) in ArminJo/Talkie

Save

README.md

139 | `ENABLE_PITCH` | disabled | Talkie.h | If defined we interpret second parameter
`aSampleRateForPitch` of `say()` and...

140 ..._DEFAULT` | 8000 | Talkie.h | f you want to globally set `pitch` for Talkie, you can
change this value, this saves the o...

160 ...from 2017, based on Peter Knights version [extended with `pitch`, speed and bending]
(<https://github.com/technologiescoll...>

164 - Adding parameter `aSampleRateForPitch` and macro `ENABLE_PITCH`.

src/Talkie.h

42 // #define ENABLE_PITCH // requires around 160 bytes of program space

45 ... ORIGINAL_SAMPLE_RATE // If you want to globally set `pitch` for Talkie, you can
change this value, this saves the o...

82 ...nst uint8_t *aWordDataAddress, unsigned int aSampleRateForPitch =
ORIGINAL_SAMPLE_RATE); // Blocking version with `pitch`

83 ...st uint8_t *aWordDataAddress, unsigned int aSampleRateForPitch = ORIGINAL_SAMPLE_RATE);
// Queuing version. Returns fr...

125 * - Adding parameter `aSampleRateForPitch` and macro `ENABLE_PITCH`.

Introduce yourself!

**Bonus Question: What's something you've
learned in the last year that you're excited
about?**



Introduce yourself!

**Bonus Question: What's something you've
learned in the last year that you're excited
about?**

Fantasy Device

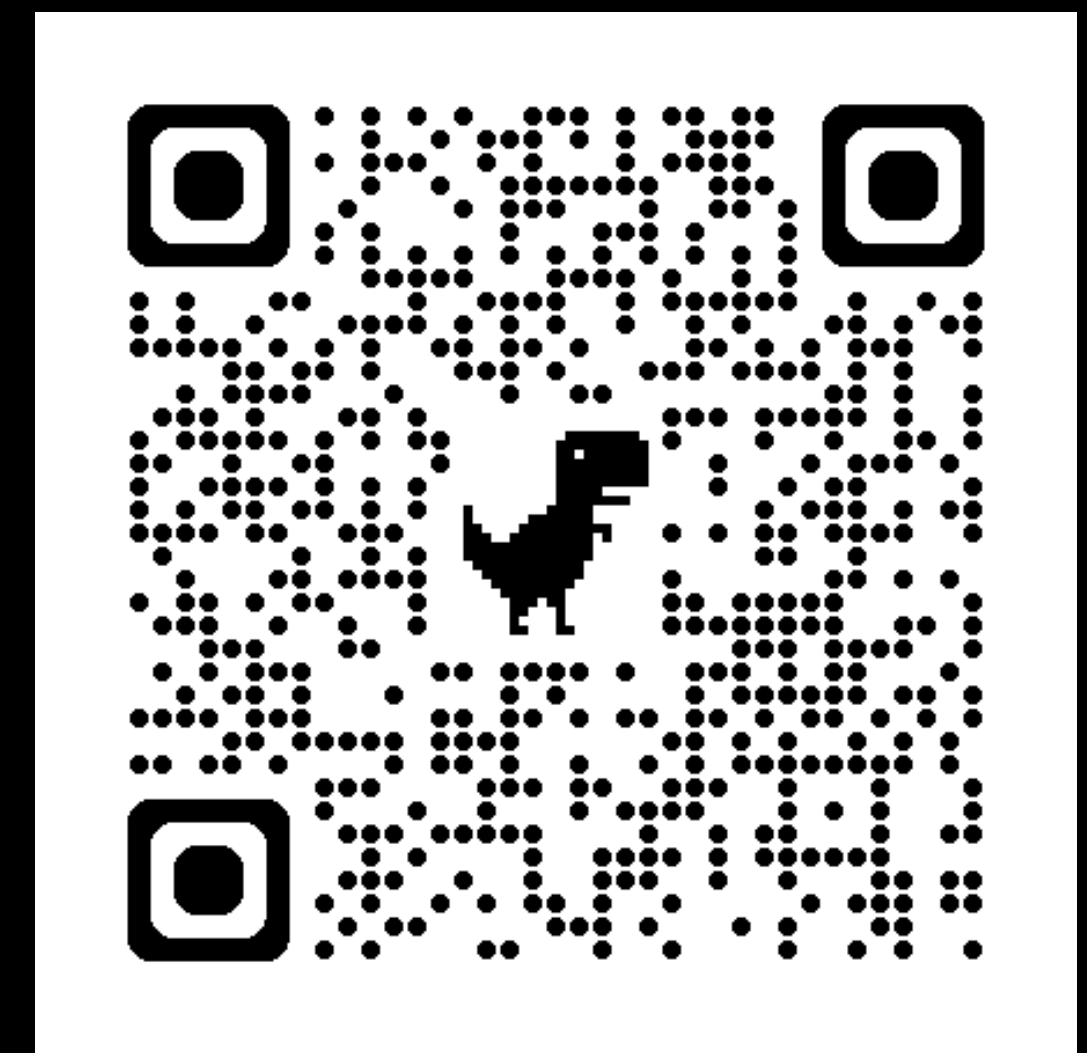
Physical Interaction
Goal/purpose

Resources

Syllabus

<https://itp.nyu.edu/physcomp/>

Schedule, assignments, labs, videos



2025 Residents

Christina Tang, Octavio Figueroa Moya,
Andres (Tobi) Franco Villamizar, Andre Lira

Drop-in Office Hours

Mine, other professors, residents.

Other students

If you can, do your homework at ITP and learn by osmosis!

Weekly Help Sessions

Mondays 12:30-2pm and Thursdays 1:30-3pm