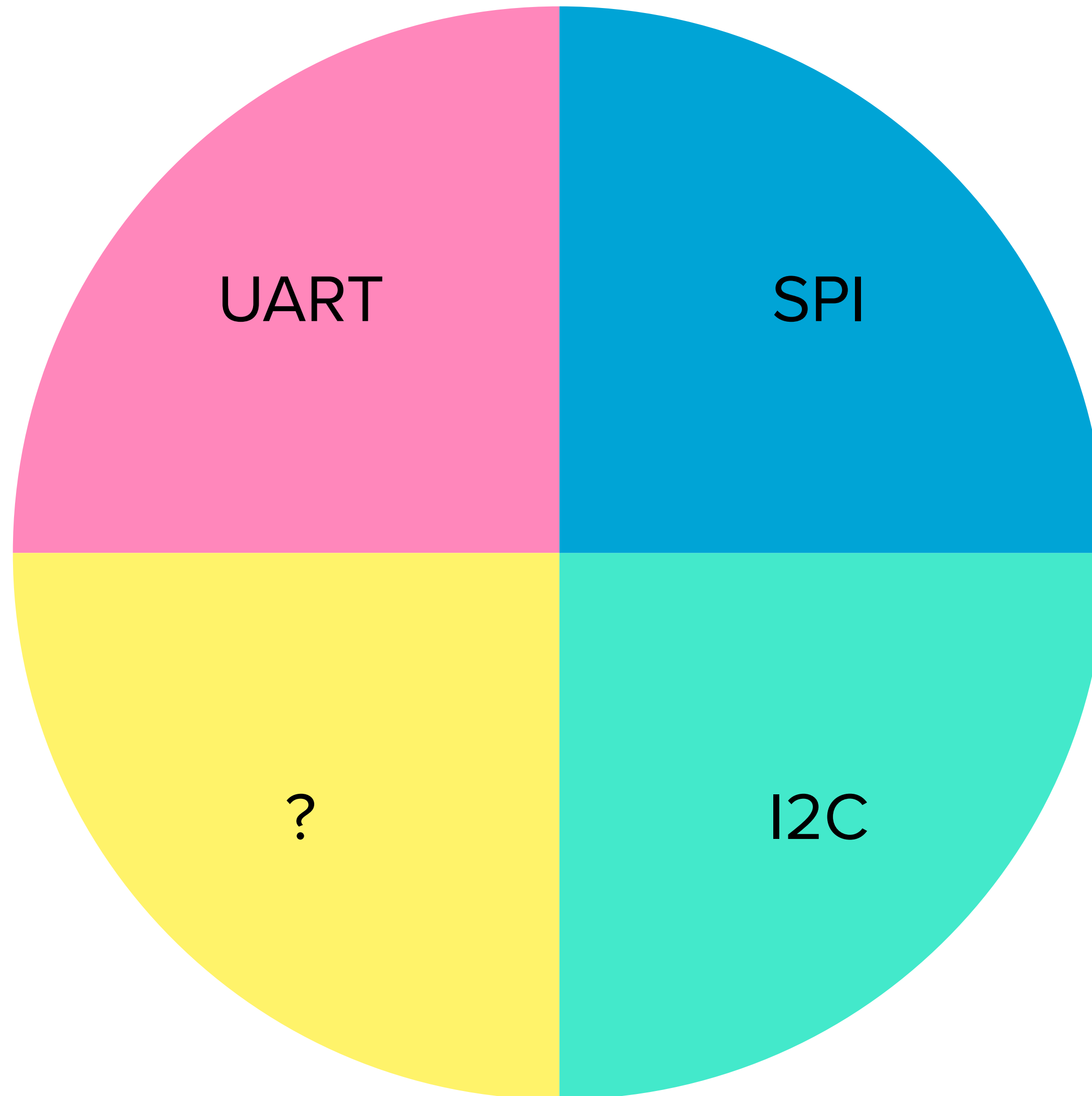


INTERACTION DESIGN

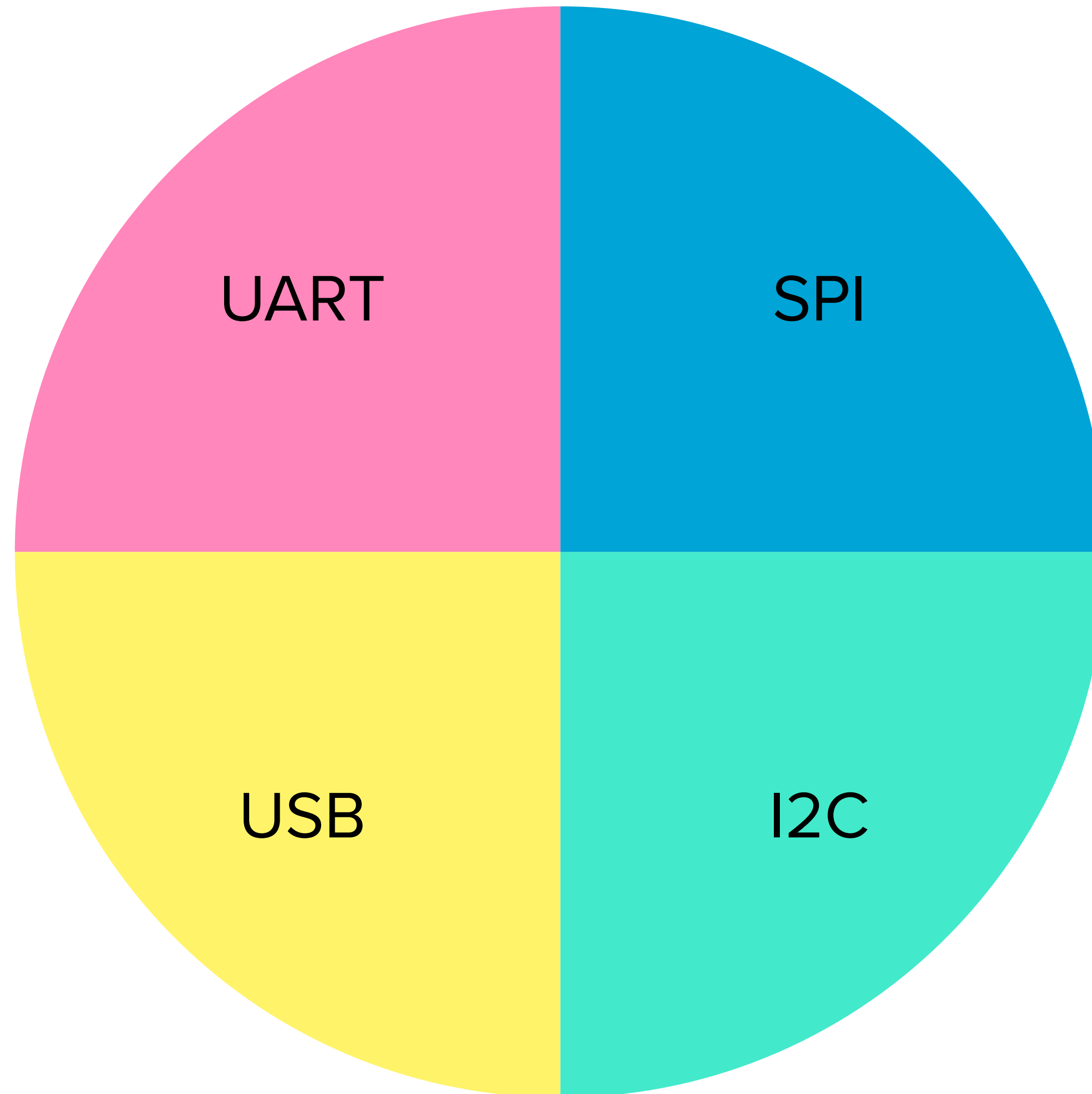
ARDUINO + P5.JS

Physical Computing HS22

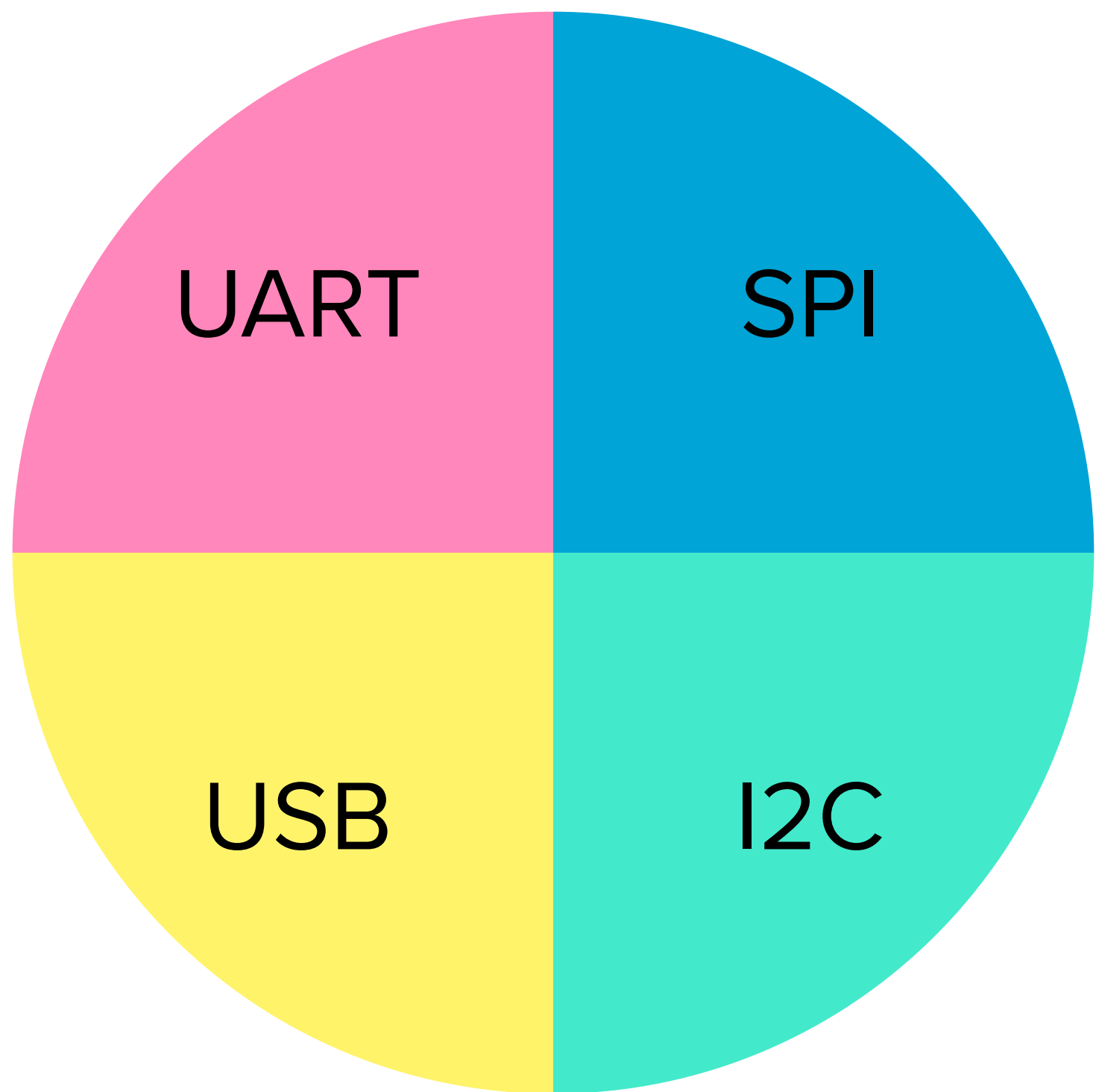
SERIAL COMMUNICATION



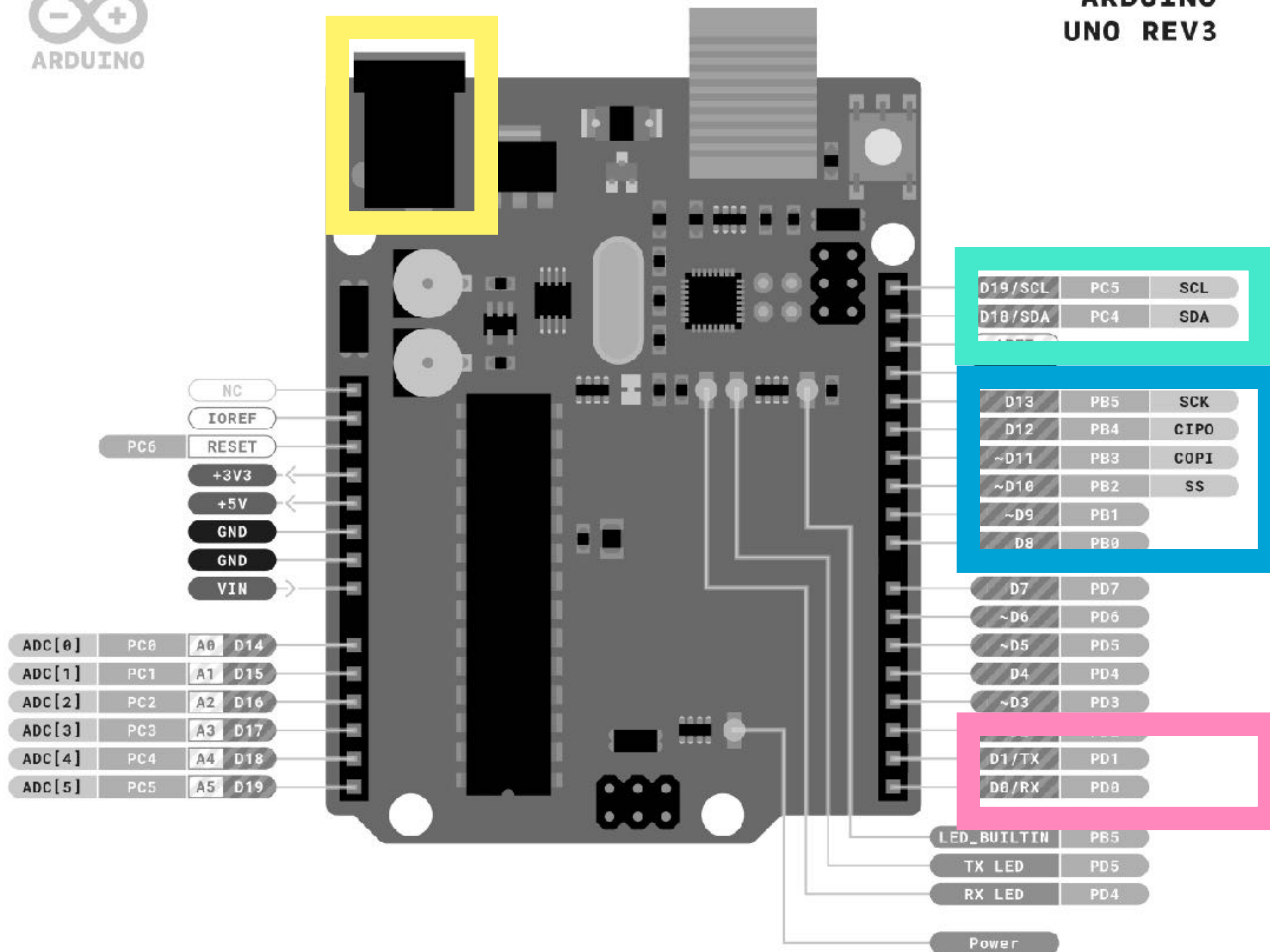
SERIAL COMMUNICATION



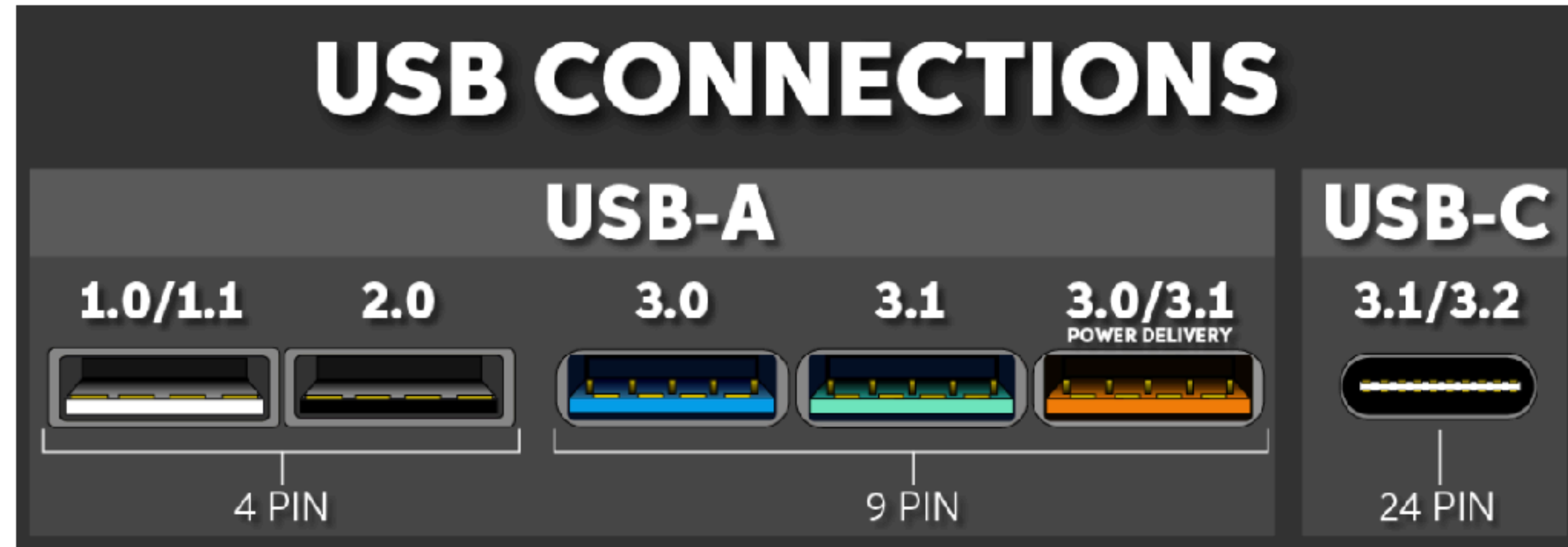
SERIAL COMMUNICATION



ARDUINO
UNO REV3

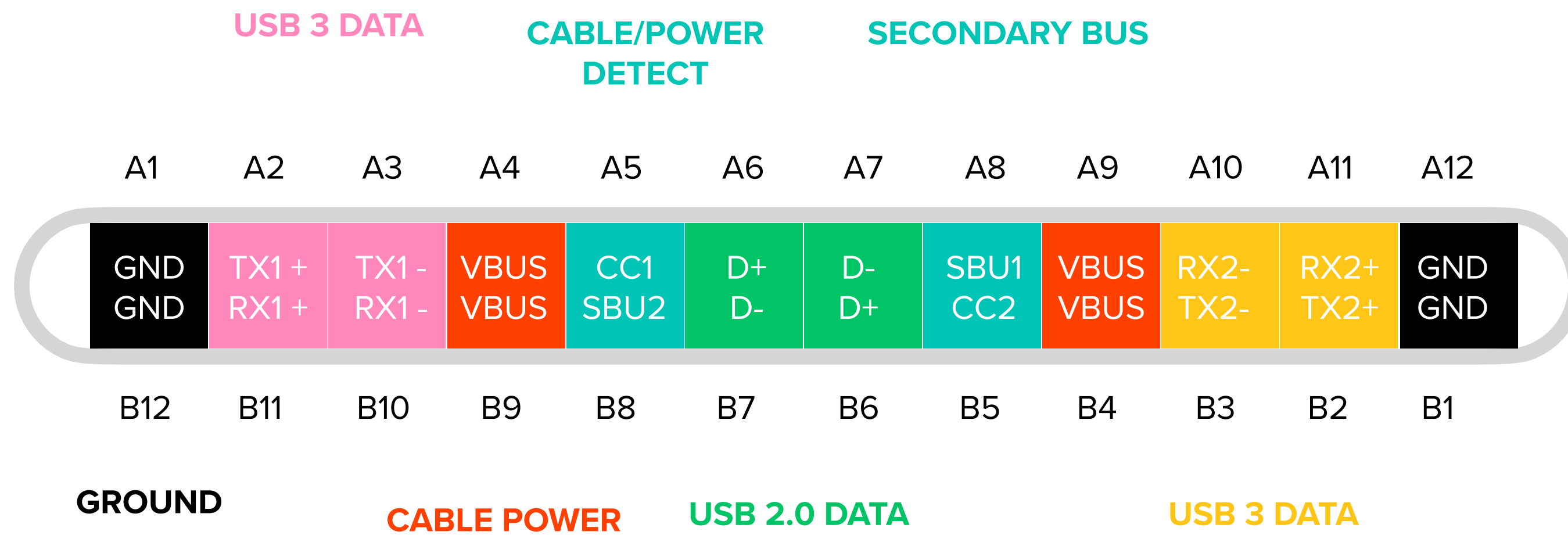


USB SERIAL COMMUNICATION

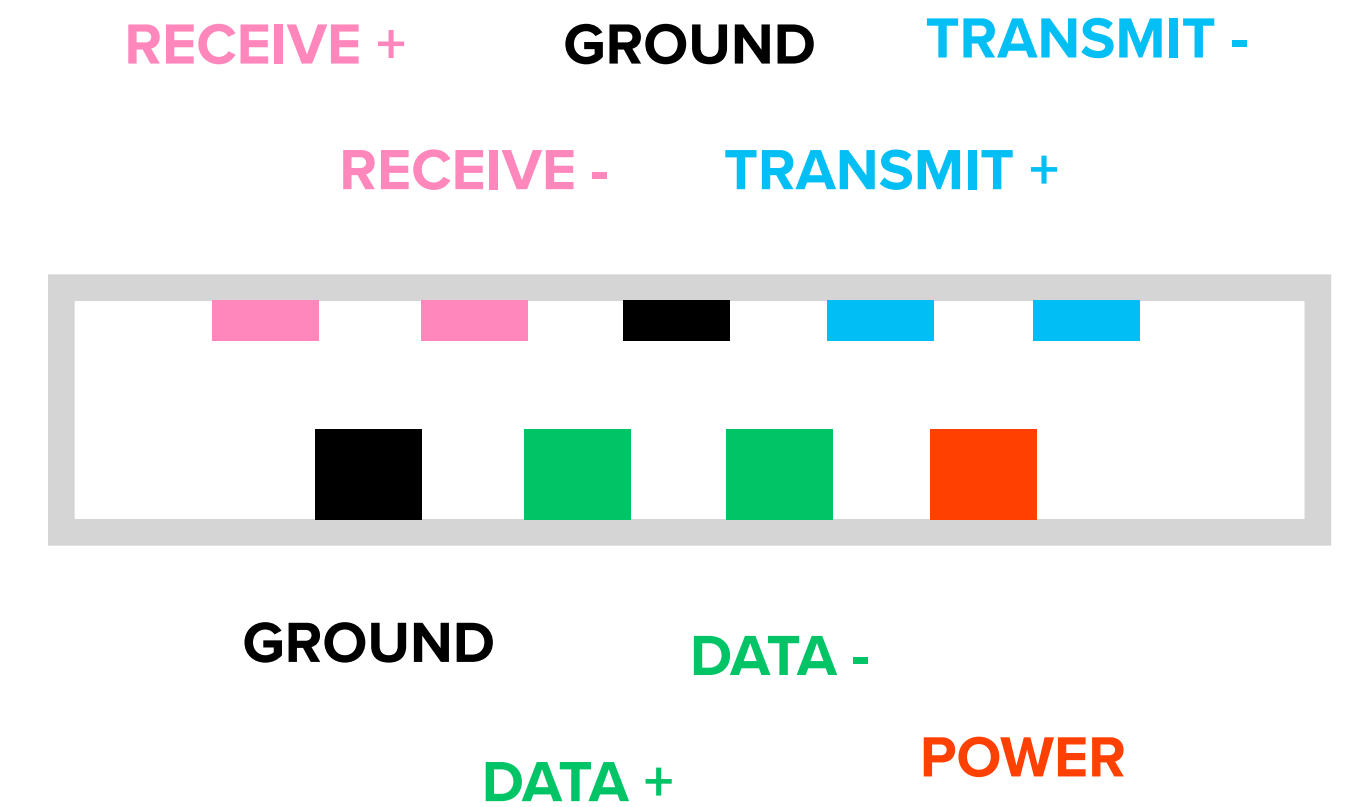


USB 4	Up to 40 Gbps	Up to 100W at 20V
USB 3 (USB 3.1 Gen 2)	Up to 10 Gbps	Up to 15W at 5V
USB 3 (USB 3.1 Gen 1)	Up to 5 Gbps	Up to 900 mA at 5V
USB 2	Up to 480 Mbps	Up to 500 mA at 5V
USB 1.1	Up to 12 Mbps	Up to 500 mA at 5V

USB SERIAL COMMUNICATION

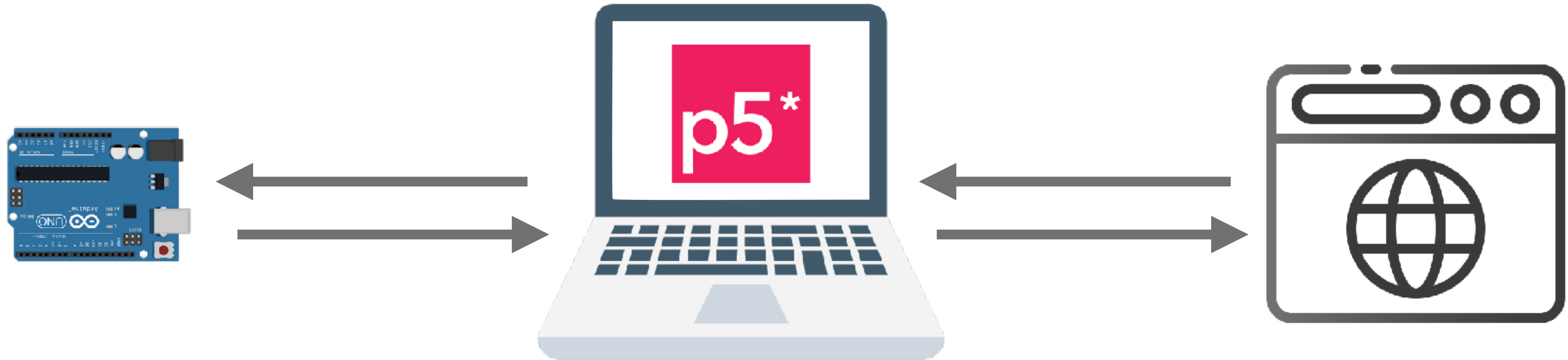


USB-C

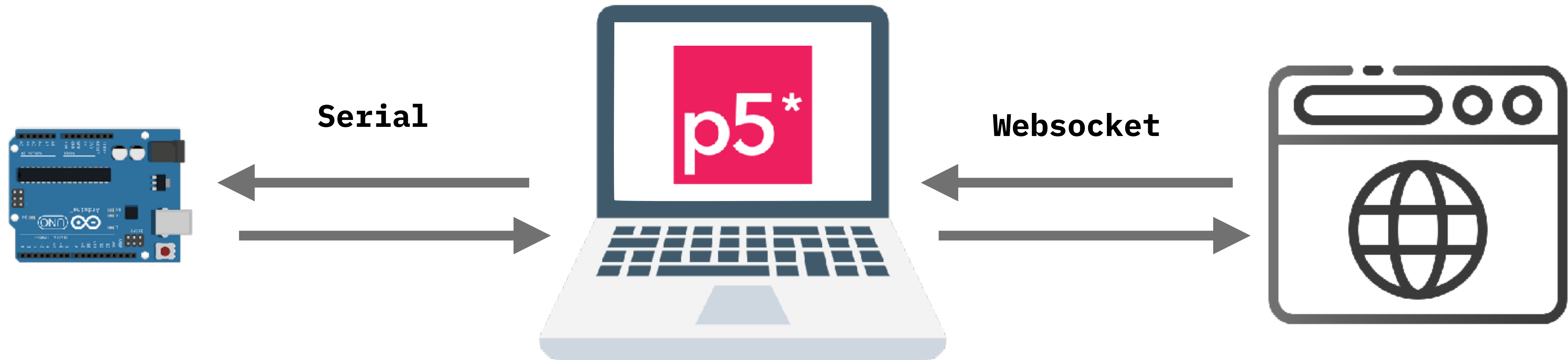


USB-A 3.0

ARDUINO <-> P5.js



ARDUINO <-> P5.js



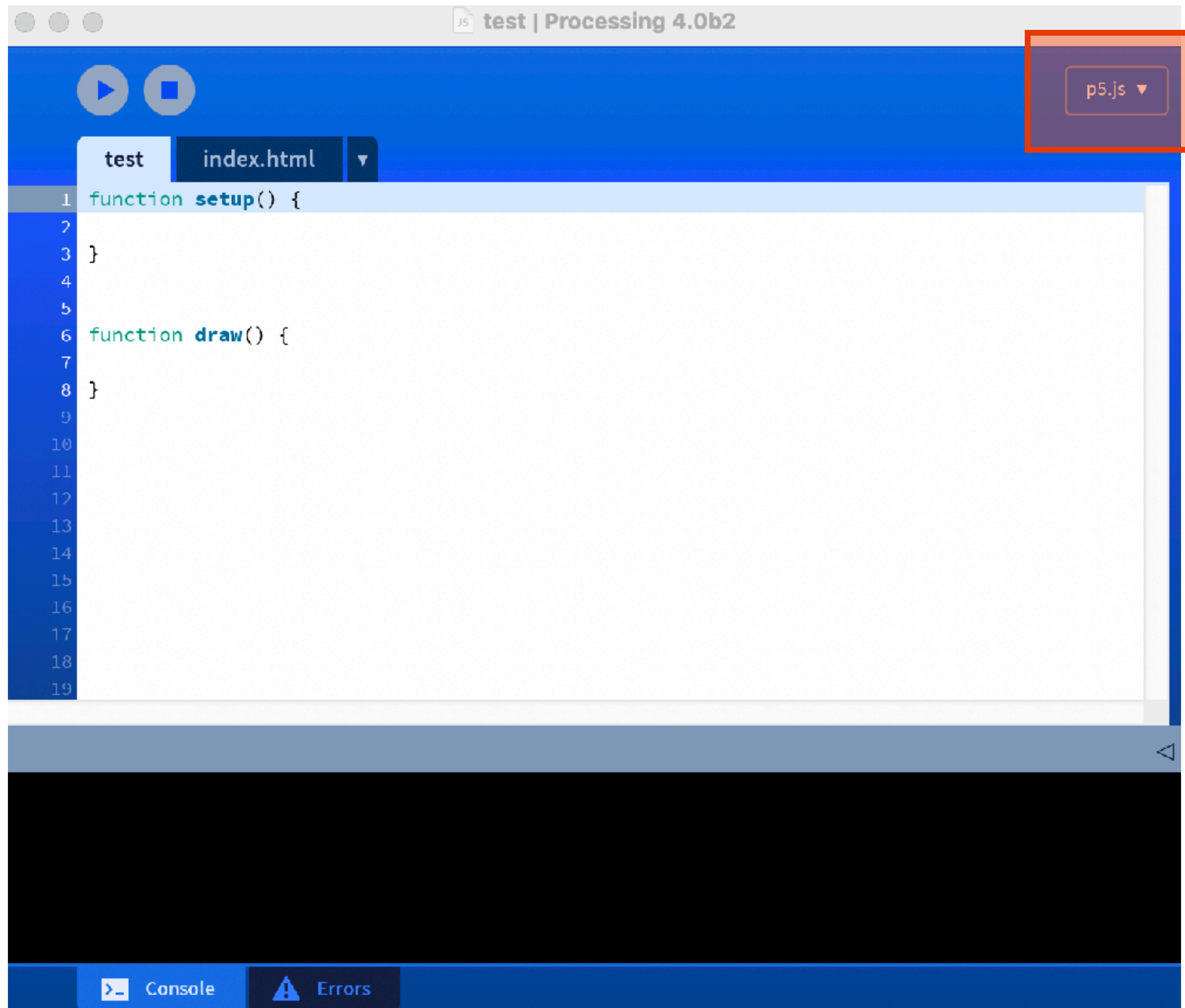
ARDUINO <-> P5.js



WEB SERIAL API

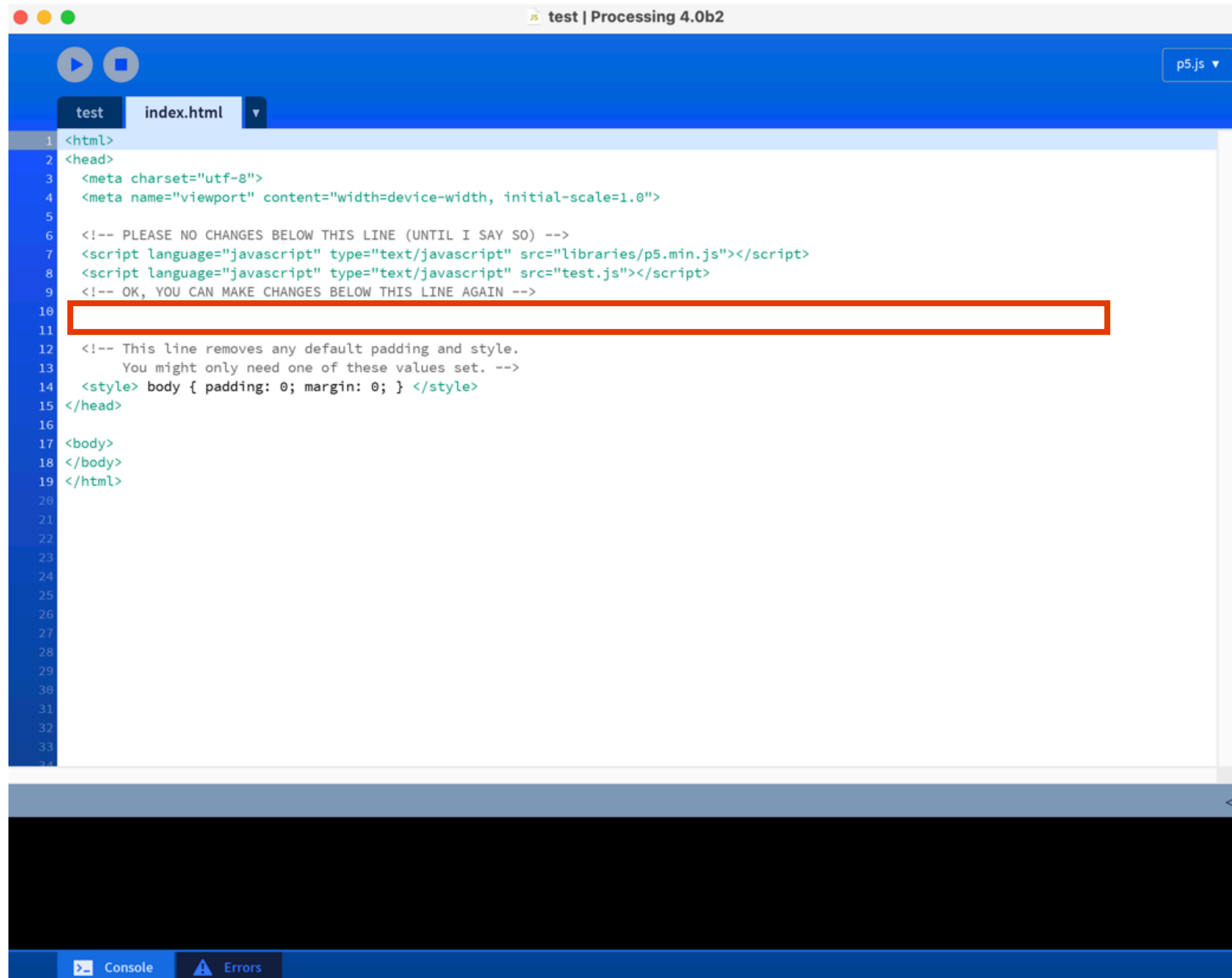
1. Enable flag <chrome://flags/#enable-experimental-web-platform-features>
2. Open dev tool console: `ctrl-shift-i` on Windows, `cmd-alt-i` on Mac
3. Type in console window (yes, the whole sentence): `"serial" in navigator`
4. You should see `"true"` if WebSerial API is running
5. Plug in your Arduino board
6. Type in console: `await navigator.serial.requestPort()`
7. Choose port named `:usbmodemXXXX`

IDE



To work with p5.js and Arduino we're going to use Processing IDE with p5.js mode enabled.

IDE



The screenshot shows the Processing IDE interface. The title bar reads "test | Processing 4.0b2". The main editor window displays the content of the "index.html" file. The code is as follows:

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5
6   <!-- PLEASE NO CHANGES BELOW THIS LINE (UNTIL I SAY SO) -->
7   <script language="javascript" type="text/javascript" src="libraries/p5.min.js"></script>
8   <script language="javascript" type="text/javascript" src="test.js"></script>
9   <!-- OK, YOU CAN MAKE CHANGES BELOW THIS LINE AGAIN -->
10
11
12   <!-- This line removes any default padding and style.
13       You might only need one of these values set. -->
14   <style> body { padding: 0; margin: 0; } </style>
15 </head>
16
17 <body>
18 </body>
19 </html>
```

The IDE interface includes a toolbar with play and stop buttons, a file browser showing "test" and "index.html", and a status bar at the bottom with "Console" and "Errors" tabs.

In this mode Processing automatically creates index.html file, where you can add external libraries:

- **Using CDN (like jsDelivr)**
- **Importing the library directly in libraries/ folder inside your Processing sketch.**

serial.js

To allow communication between an Arduino board and p5.js we'll use `serial.js` library written by Jon E. Froehlich.

Use the `jsDelivr` service and add the line in `index.html` file below the line:

//OK, YOU CAN MAKE CHANGES BELOW THIS LINE AGAIN:

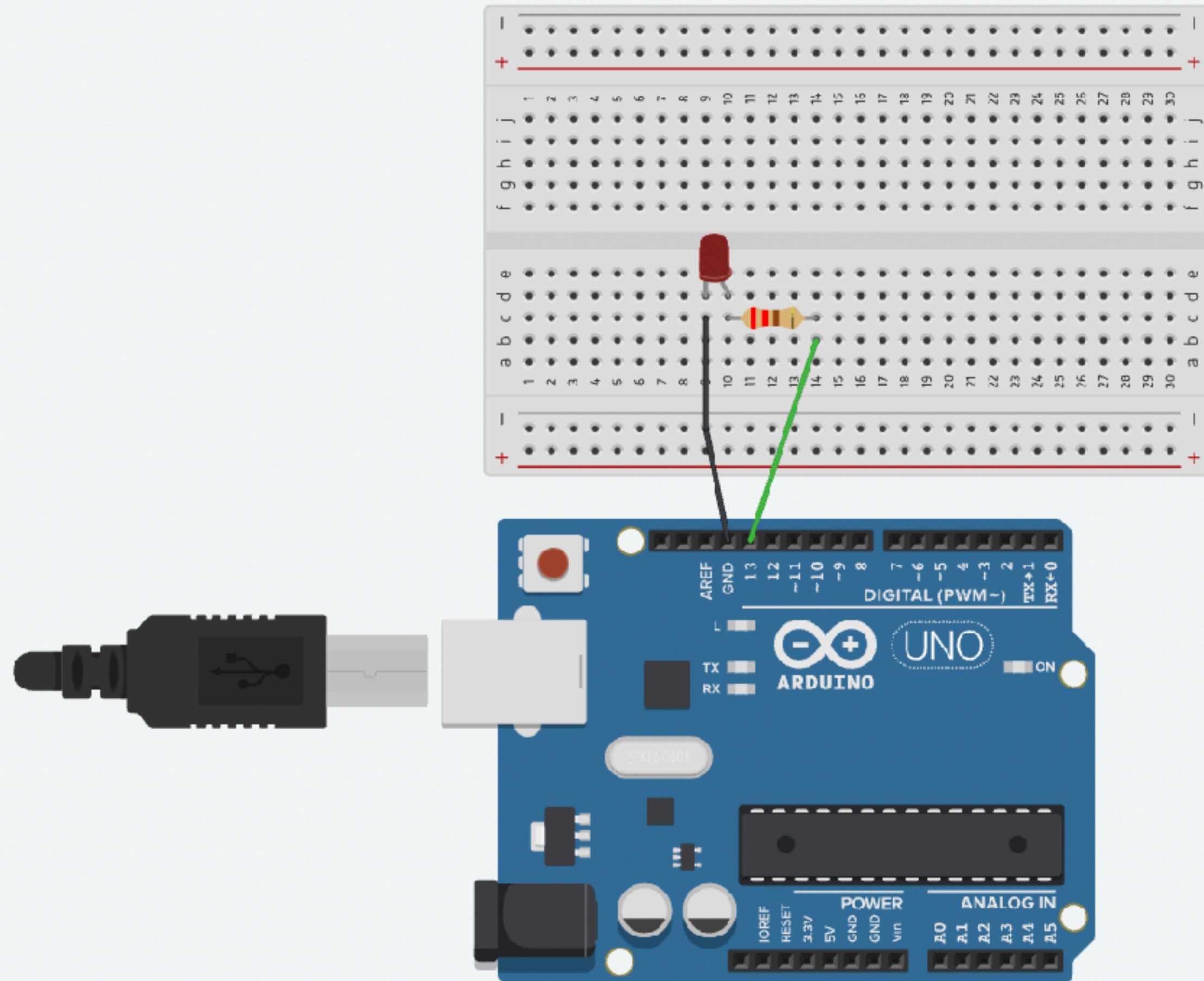
```
<script src="https://cdn.jsdelivr.net/gh/makeabilitylab/p5js/_libraries/serial.js"></script>
```

P5.JS → ARDUINO

Physical Computing HS22

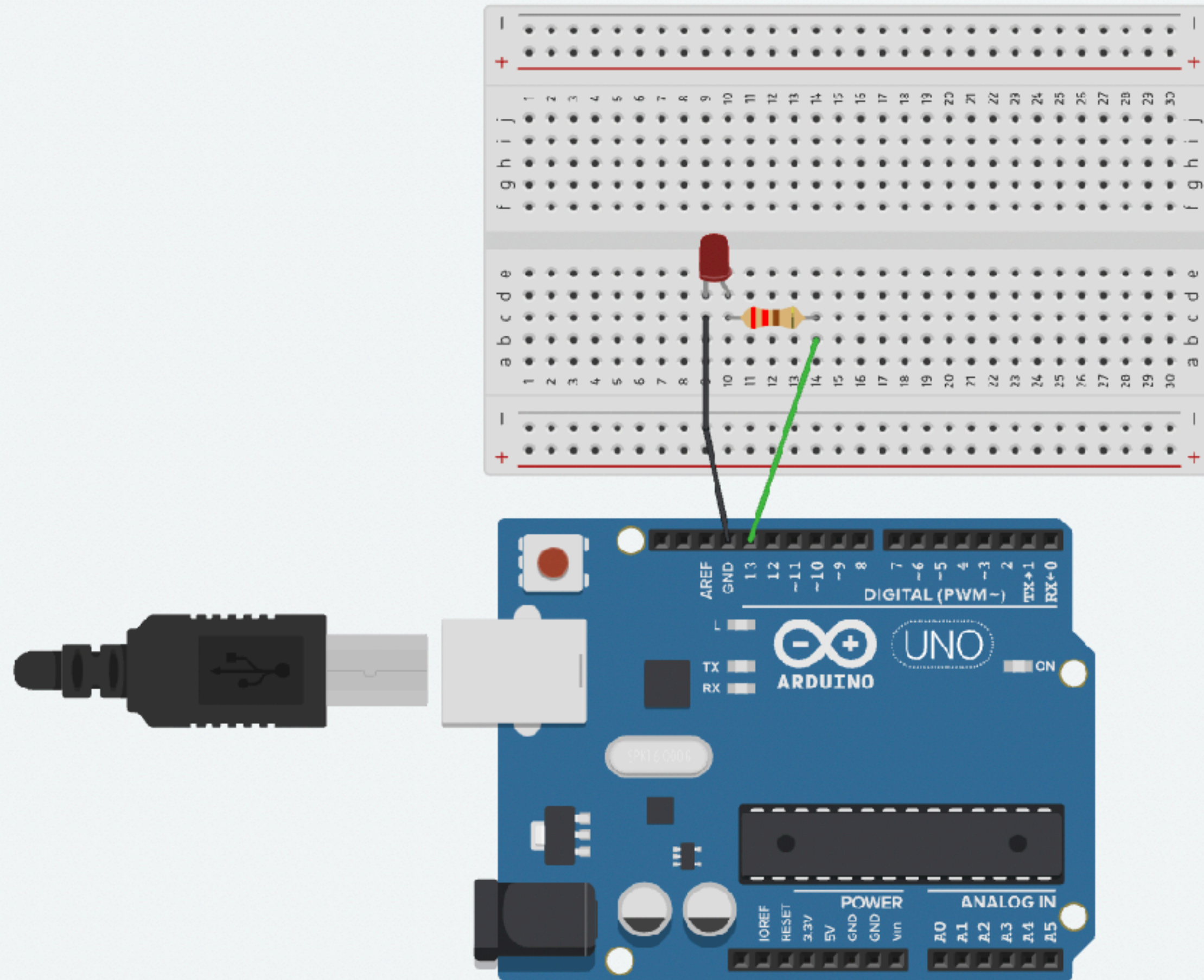
Example 1

Control the LED with an array of data.
Control the blinking with on/off button.



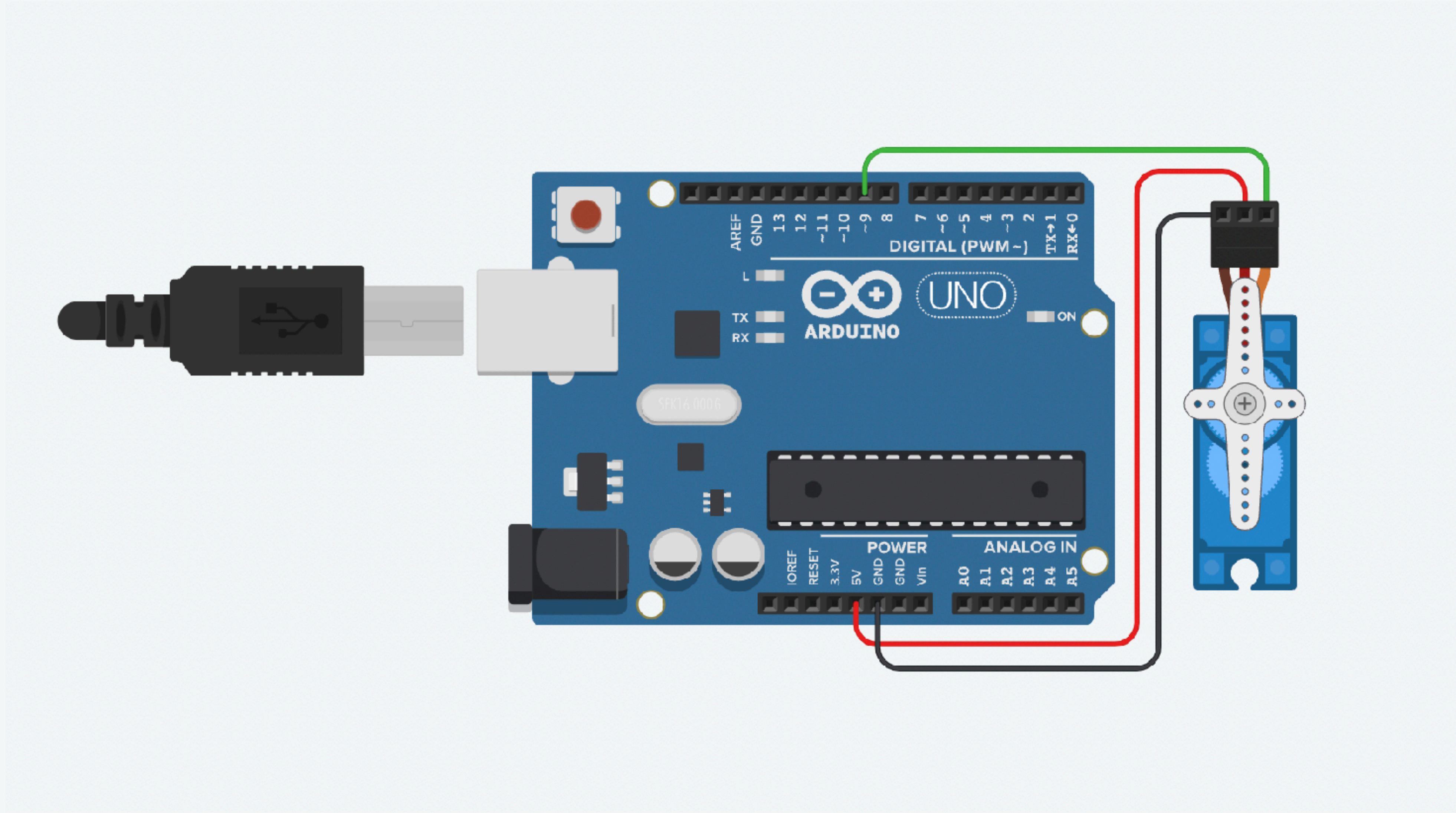
Example 2

Control the LED brightness using your voice.



Exercise 1

Control the servo speed using your voice.

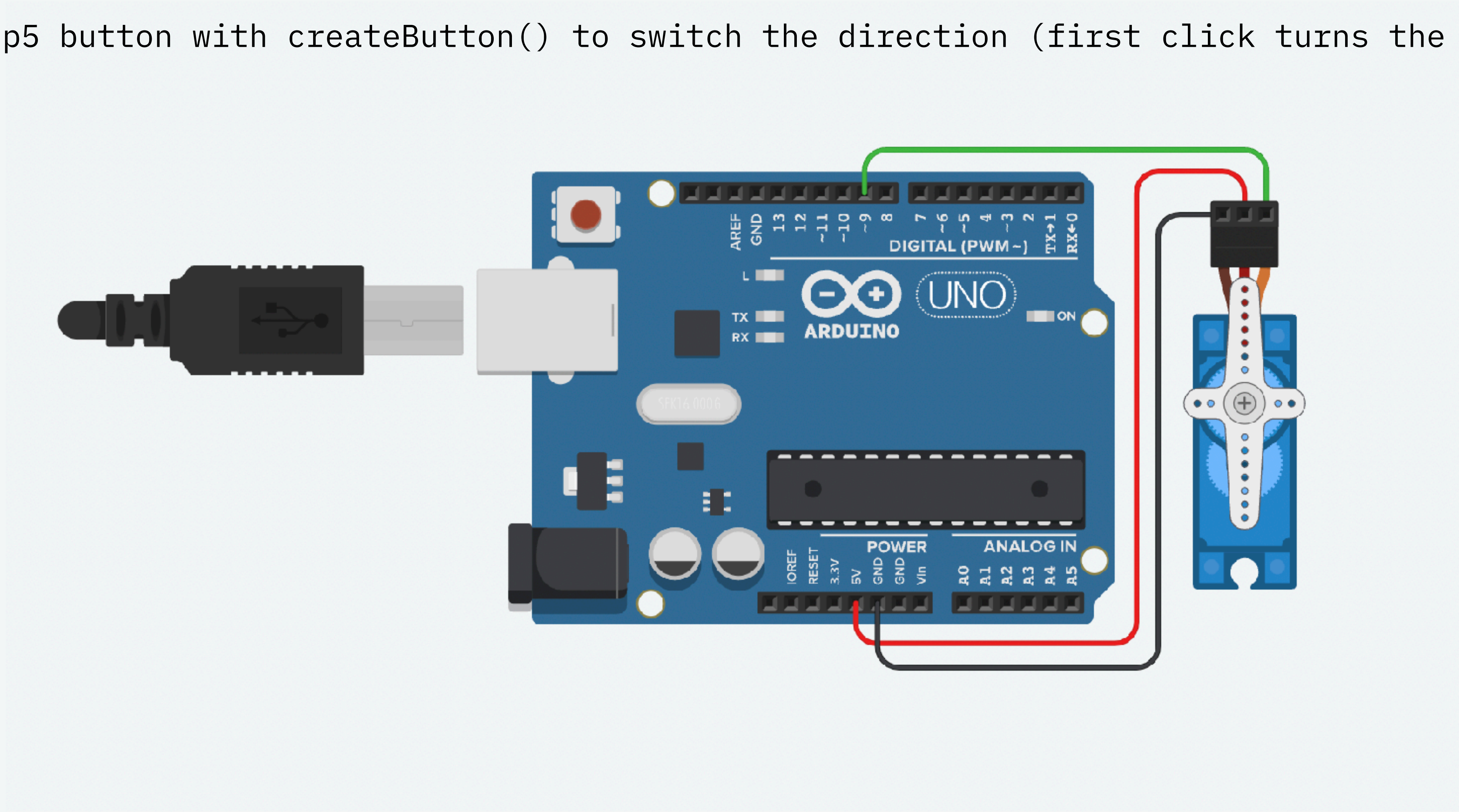


Exercise 2

Create a p5 slider `createSlider()` the speed of the servo motor

Advanced:

Create a p5 button with `createButton()` to switch the direction (first click turns the servo on)



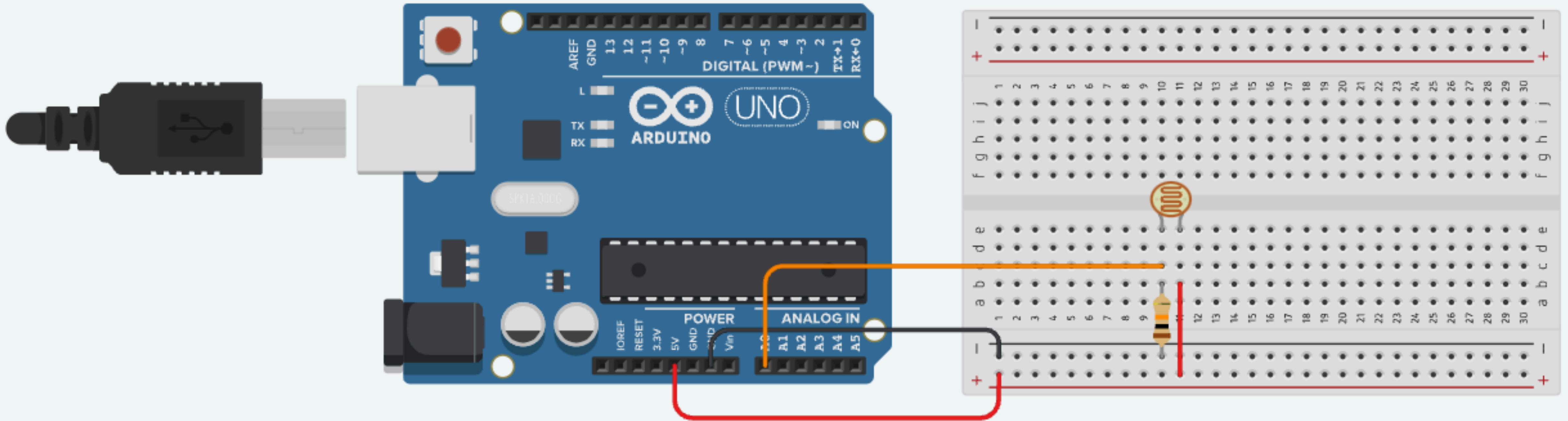
INTERACTION DESIGN

ARDUINO → P5.JS

Physical Computing HS22

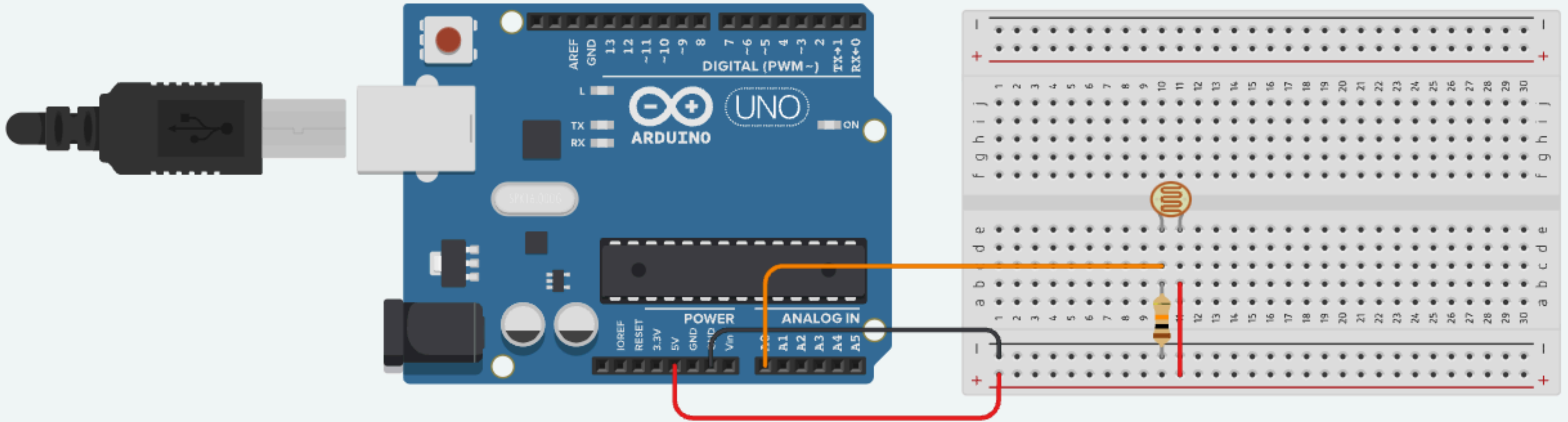
Example 3

Add a photoresistor on pin A0 to change the text from "DAY" to "NIGHT"



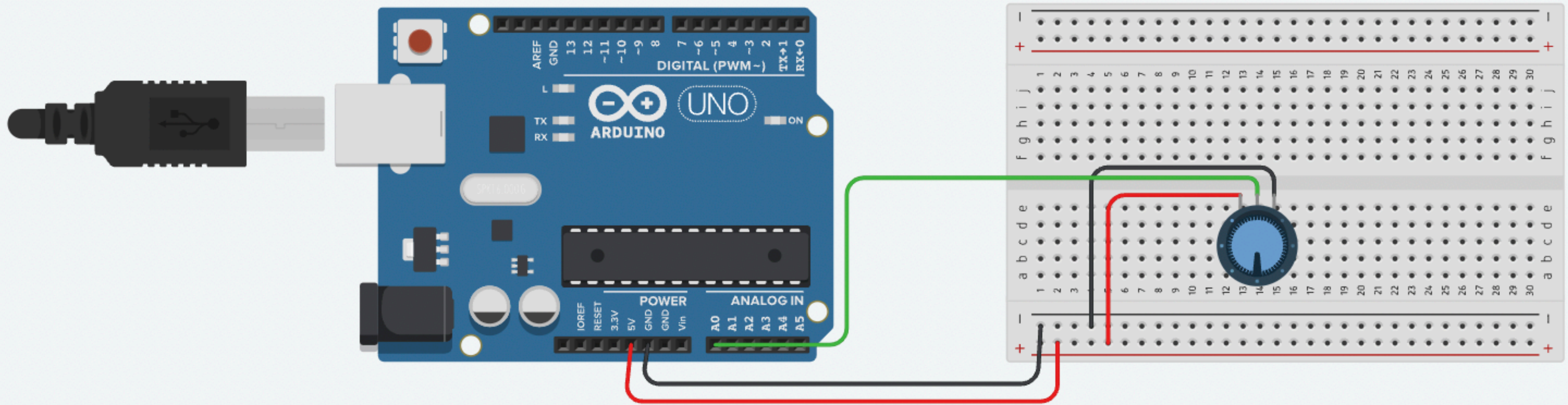
Exercise 4

Control the position of a circle with photoresistor.
Make sure to stay within the size of your canvas.



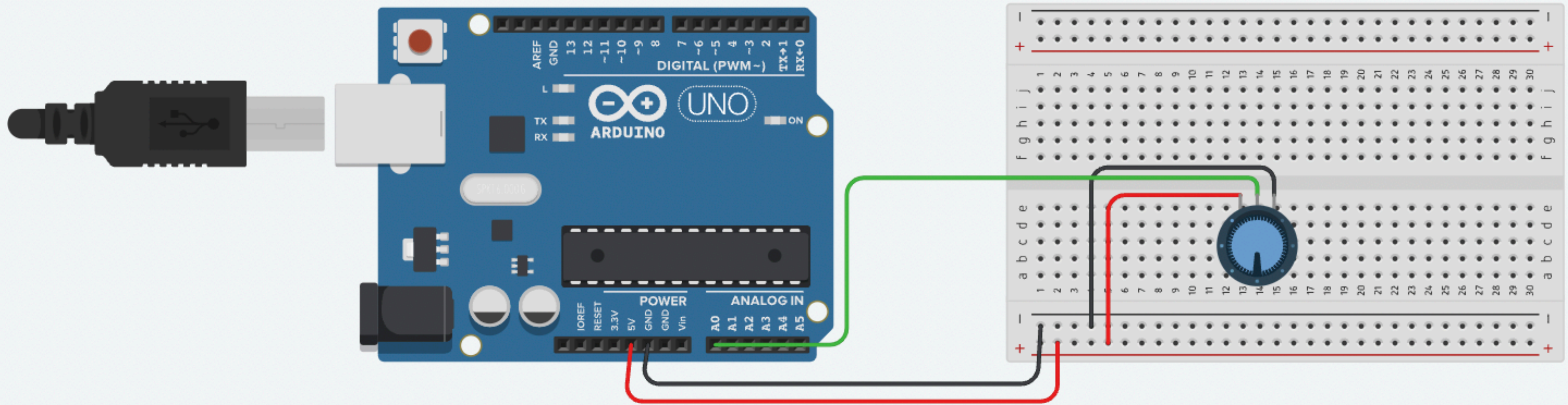
Example 4

Add a potentiometer on pin A0 to manipulate circle size in p5.js



Exercise 4

Use the potentiometer to change the colours of the background.
Use `lerpColor(color1, color2, amount)` to smoothly change from `color1` to `color2`
The `amount` is controlled by the `shapeFraction` value.



Exercise 5 (optional)

Add a button on PIN 2 to invert the **color1** and **color2** from the exercise 4.

