Program, code, use of data structures, design, and use of the API

In this project we are sending data from a p5.js to a microcontroller. The purpose is to click and drag the mouse inside of a p5 sketch and in that way, you can change the brightness of a LED on the board. The LED is attached to pin 27 and after board-setup we started the Arduino-code by having a variable to hold the LED. The first variable contains the serial communication at 9600 and turn the LED as an output. Afterwards we have the local variable of the brightness in a loop that's going to store the data of information that we get from p5.js into our sketch. The second loop

```
void loop() {
  int brightness;
  // If statement til at checke hvorvidt der er sendt noget fra p5js:
  if (Serial.available() > 0) {
    // Læs data sendt via p5js scriptet:
    brightness = Serial.read();
    // Juster lysstyrke:
    digitalWrite(LED_PIN, brightness);
  }
}
```

is for checking to see if there is something inside the serial buffer on the Arduino and if it is necessary, it reads the information and shows on the pin. Therefore the "serial.available" checks to see if there is something inside the serial buffer. If you send something from processing, then there will be more then 0 bite in there, which means there is something, and if there is something inside there, then the other serial will read the oldest bite out the serial buffer and then it will store inside this variable called brightness which we use to set the value of the LED.

In the p5 sketch the first variable contains the serial object which contains holding the serial library the second variable contains the brightness and how bright the LED is going to be and the data that we will send back to the Arduino sketch. The last two variables hold the color because we draw a gradient.

```
let serial; // variabel der skal indeholde Serial objektet
let data = 0; // variabel der indeholder vores data der er sendt
let black
let white;
```

In the function setup

```
function setup() {
  createCanvas(512, 512);
  black = color(0);
  white = color(255);
  serial = new p5.SerialPort()
  serial.open(serialPortName);
```

we define the colors, so the dark is black, and the light is white/yellow-ish. We also have the serial port we can open without reading the data from the microcontroller. The draw gradient function is to draw the gradient, so we are interpolating between the two colors: dark and light so we can draw every time we come to the draw loop.

```
function drawGradient(c1, c2) {
  noFill();
  for (let y = 0; y < height; y++) {
    let interp = map(y, 0, height, 0, 1);
    let c = lerpColor(c1, c2, interp);
    stroke(c);
    line(0, y, width, y);
  }
}
```

We also did a circle where the mouse is so we can see where we are on the sketch. The only time we send data to the Arduino is when we are clicking and dragging the mouse so therefor, we have the mouse drag function which gets called every time we click and drag the mouse. The way we send the data is by saying “serial.write” and the value we want to send is a bite which is between 0 to 255.

```
function mouseDragged(){

  data = floor(map(mouseY, 0, 512, 0, 255));
  data = constrain(data, 0, 255);
  serial.write(data);
  console.log(data);
  serial.read();
}
```

The result of this project is that when you drag the mouse down on the sketch you will see that the LED on the board will get lighter and lighter and if you do the opposite the LED will get darker and darker and then turn off.

Clarification of concepts

Serial – The serial monitor is used to communicate with the Arduino board

LerpColor – is used to blend two colors to find the third color, it depends on the monitor/parameters

Data – It's a variable with the data we send

CreateCanvas – it creates an element and the dimensions of the pixels

Map – Re map a number from one range to another

Console – It is used to print a messenger to the browser's web console