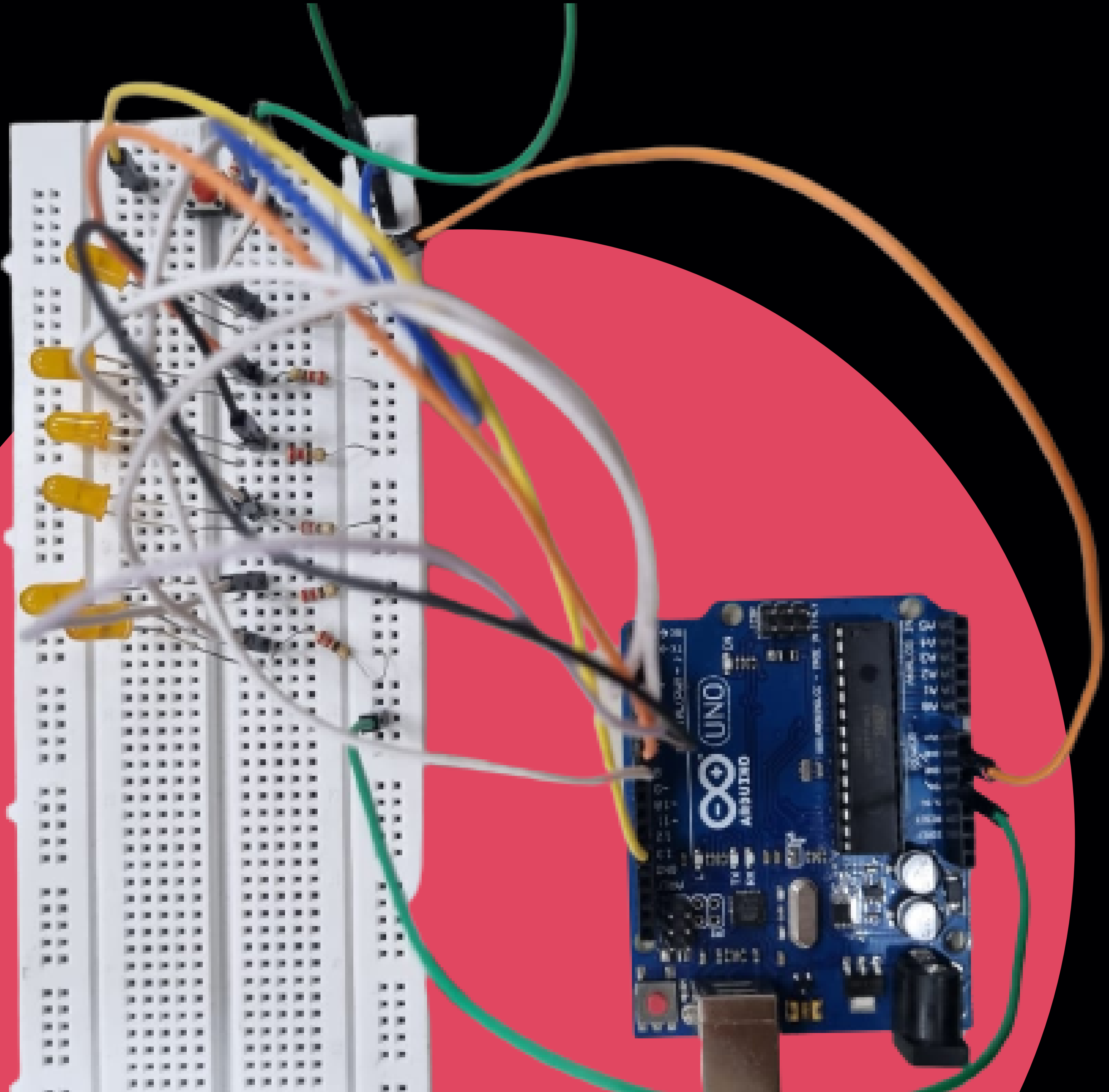
DSL 124 ~ PROF. JAY DHARIWAL

ASSIGNMENT 4

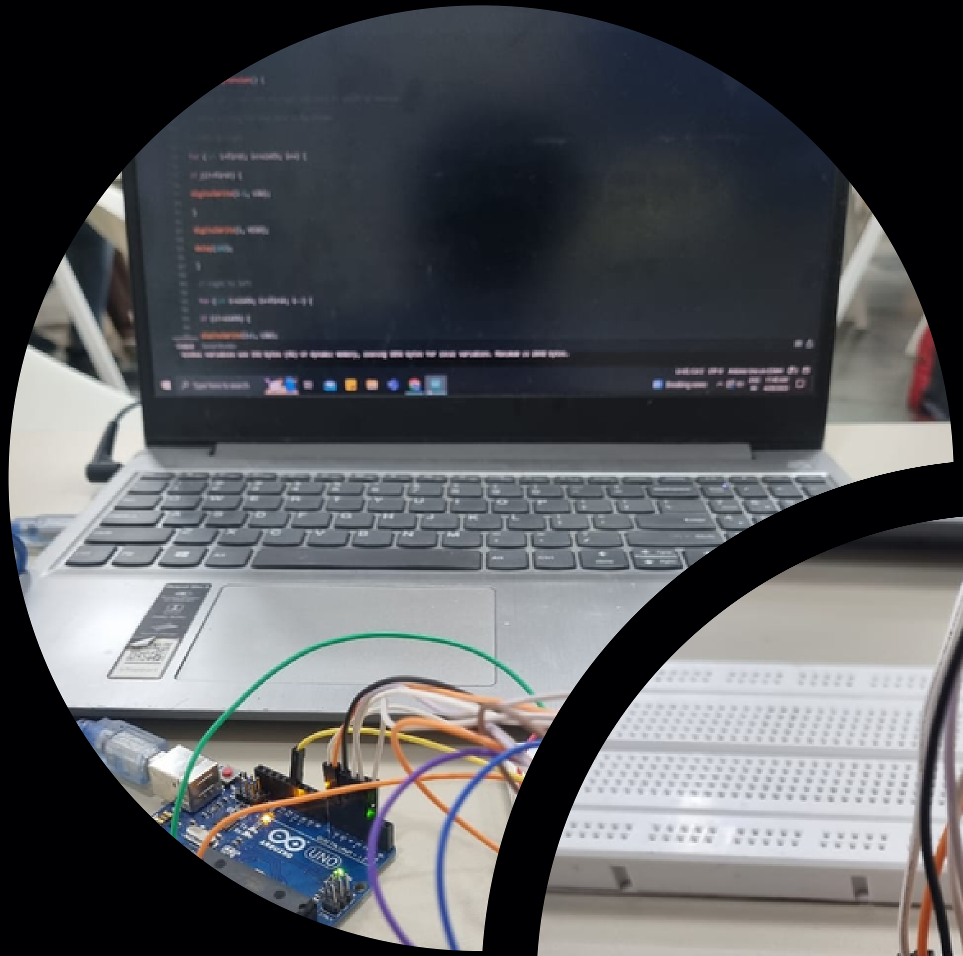


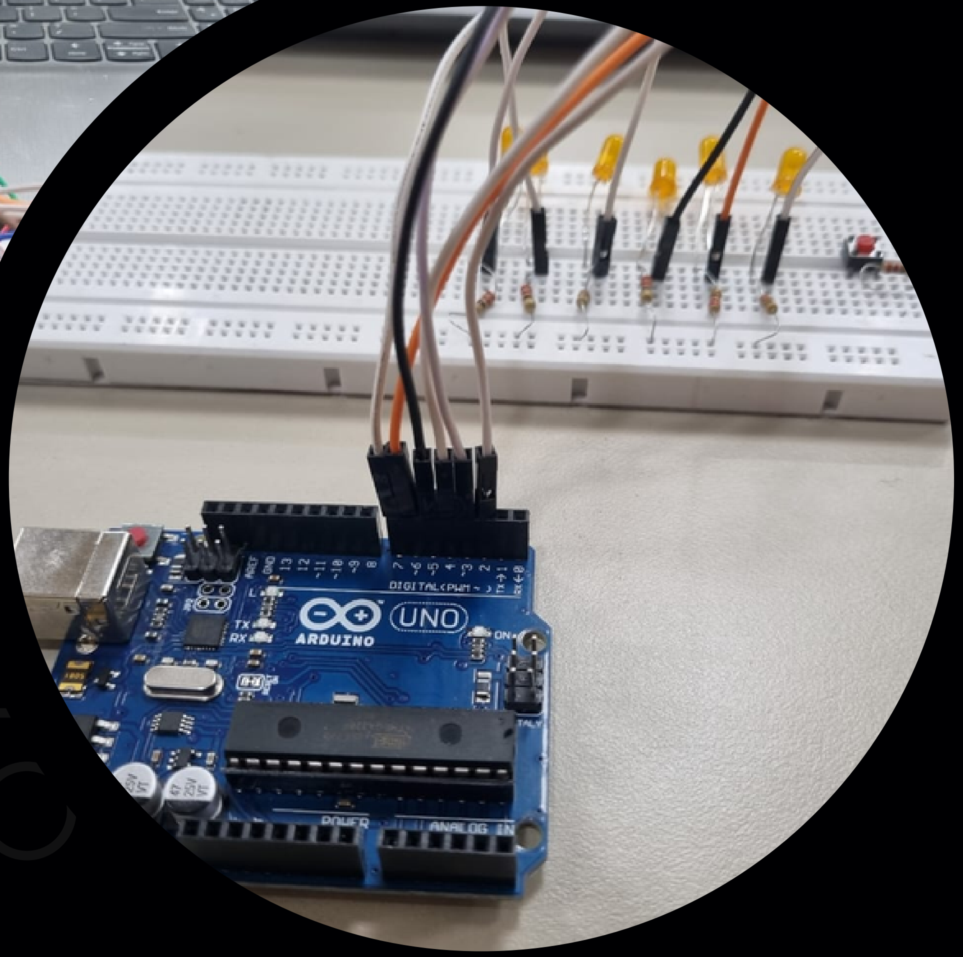
"Roll the dice with a push - a fun way to learn electronics and probability!"

-An arduino project

MORE ABOUT THE PROJECT

This project is a digital dice that can be used as a fun game accessory or a teaching tool for probability and electronics. With just a push of a button, the program generates a random number between 1 and 6 and lights up the corresponding number of LEDs. This project can be used in situations where dice are not available, such as during travel, outdoor activities or when you want to play a quick game with friends. It's also a great way to learn the basics of electronics and programming with the Arduino platform.





WHAT I LEARNT ...

Through building this project, I was able to develop my skills in circuit design and programming, gaining valuable hands-on experience in combining sensors and devices according to my design requirements. I also learned how to modify existing code from online resources to suit my needs, demonstrating my adaptability and resourcefulness as an engineer. These newfound skills are essential for success in a variety of engineering projects, making this project a valuable learning experience for me.

MATERIALS REQUIRED

01 **Arduino UNO Board**

The microcontroller used

02 **Push Button**

Used to roll the dice

03 **Breadboard**

To do the connections

04 **6 LED'S**

Number of LED'S that light up = Number on dice

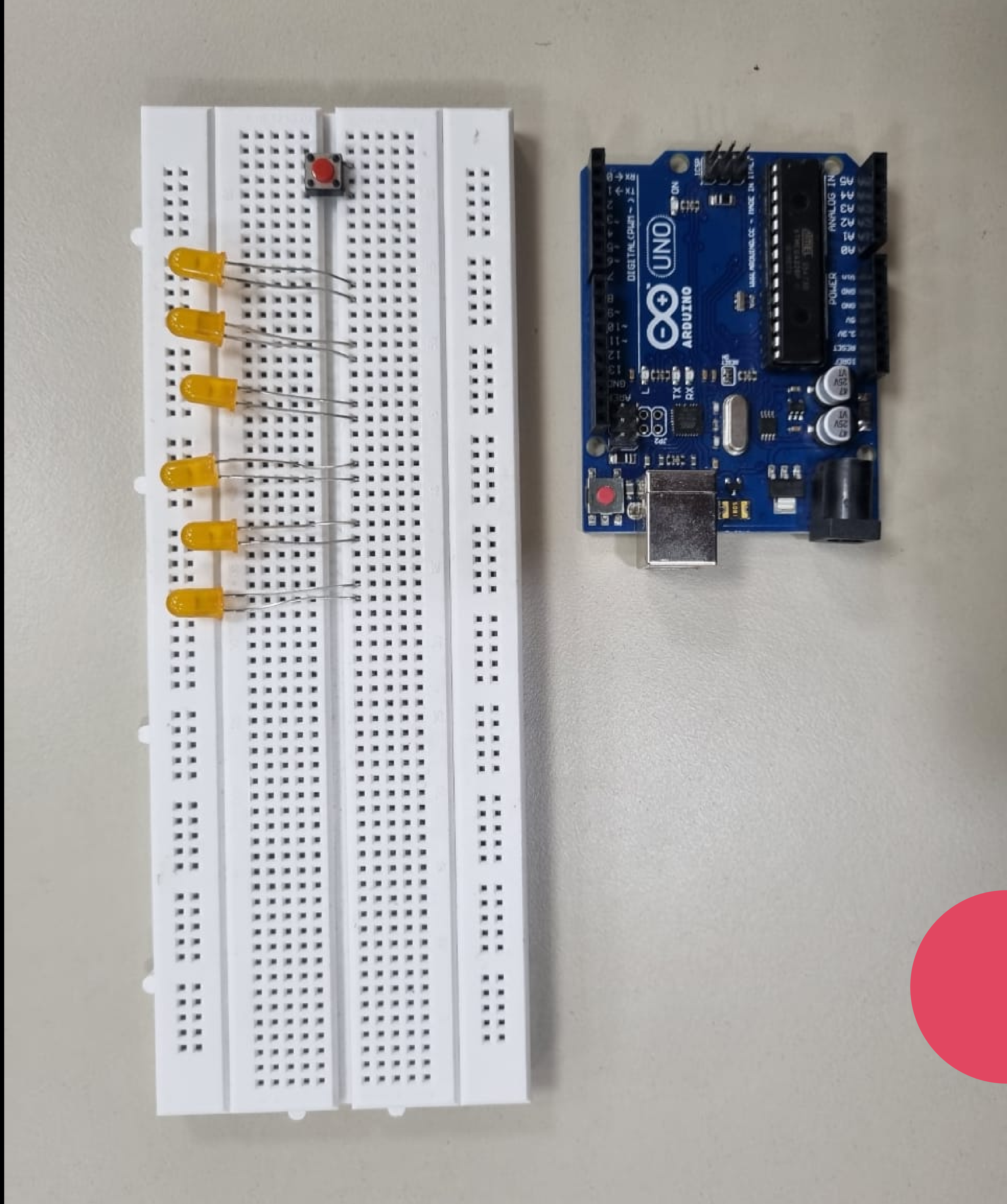
05 **Jumper Wires**

06 **Resistor**

SET UP THE LEDS AND BUTTON

STEP - 1

1. Place the 6 LEDs in the breadboard as shown. They should go in rail E with two spaces in between each LED. You should place the Anode end (long end) of the LED in the first of the two spaces and the Cathode end (short end) in the second space. Place your LED Anode ends in rows 3, 7, 11, 15, 19, and 23.
2. Place your button in the breadboard as shown. The left side should go in rail E, space 59 and 61, and the right side should go in rail F, space 59 and 61.

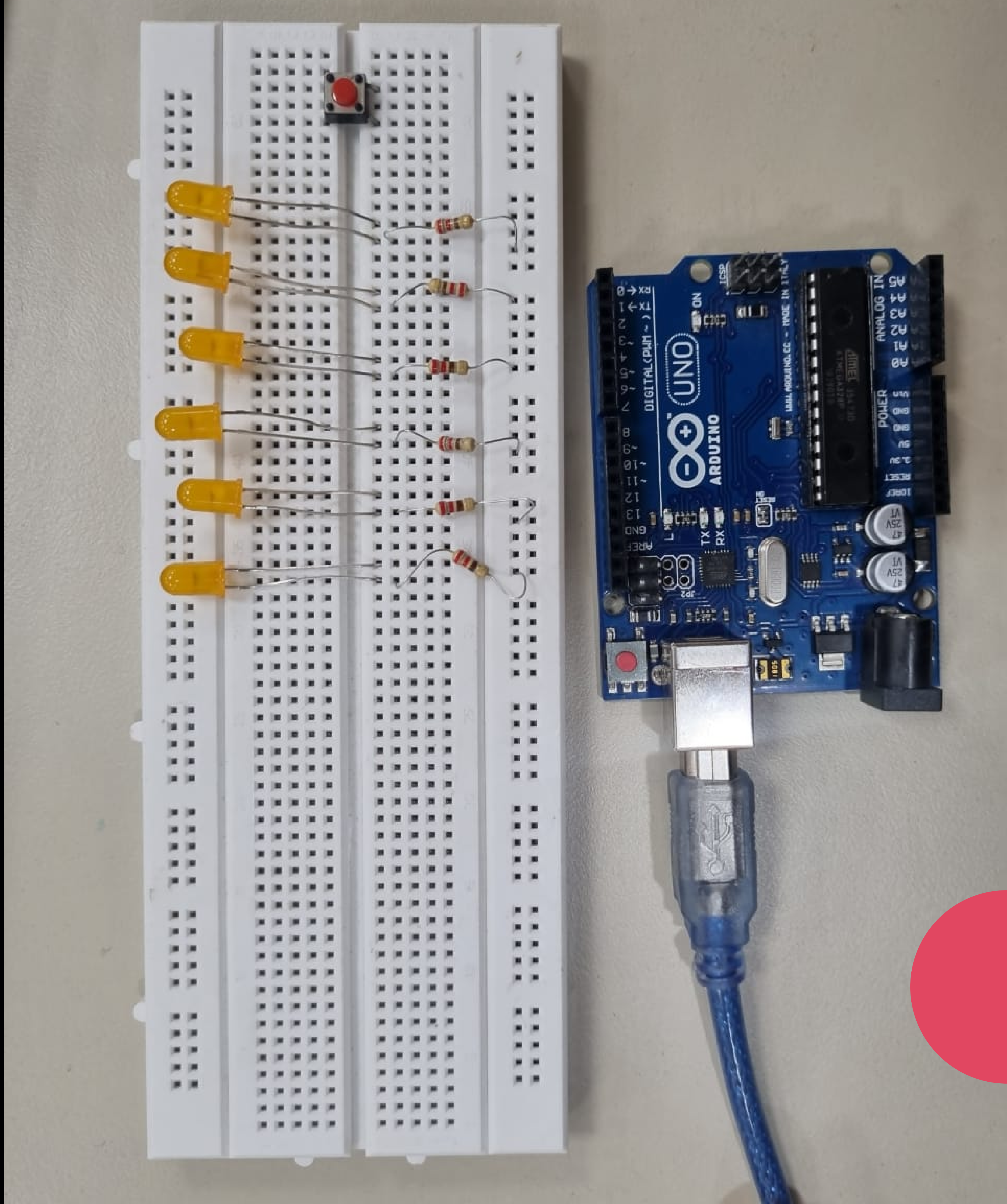


ATTACH THE RESISTORS

STEP - 2

1. Place a 1 k-ohm resistor near your button. It should go from the bottom left of the button (row 61) to row 64.

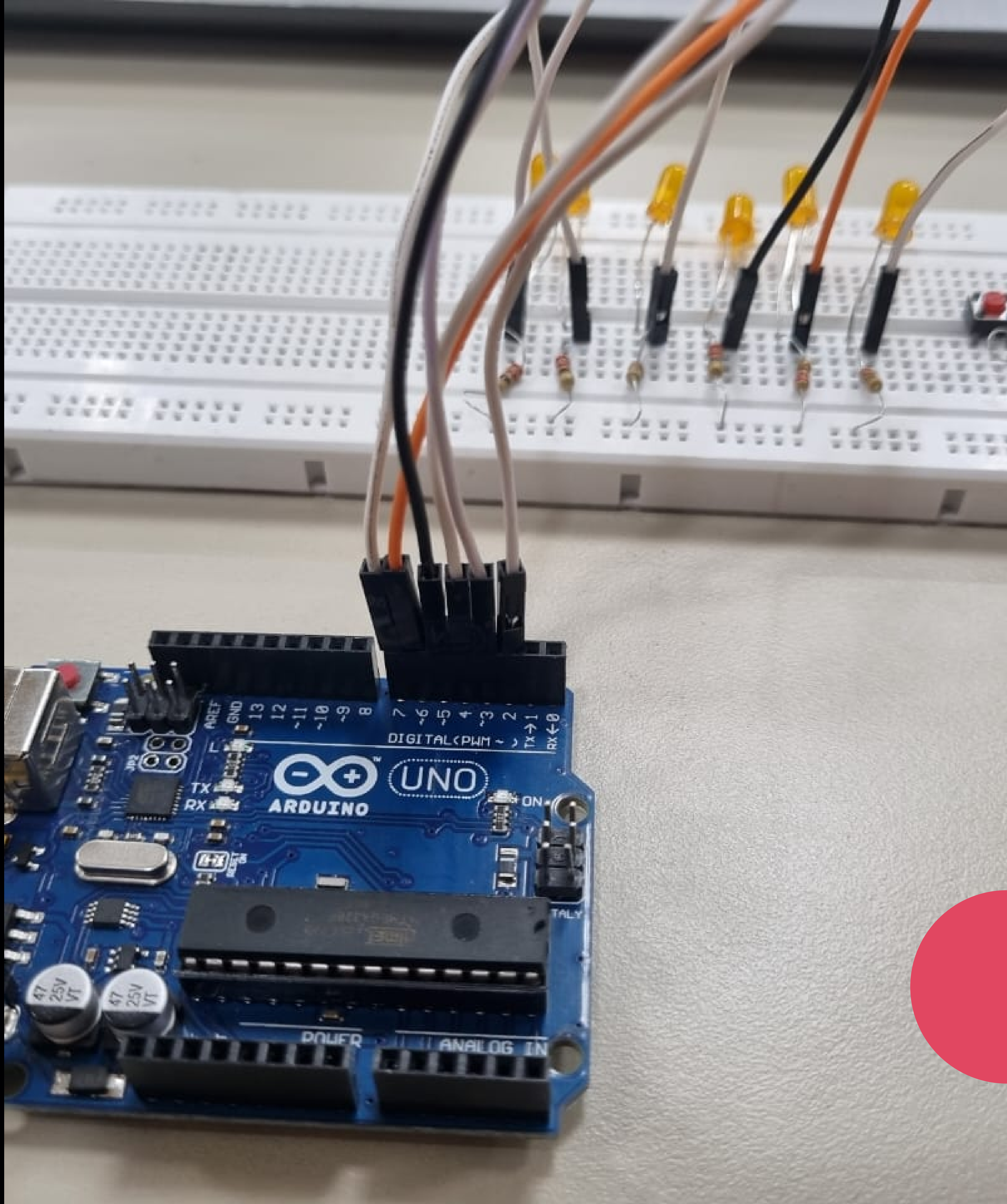
Initially, I had added a resistor for each LED while building the apparatus. However, I noticed that this caused a reduction in the brightness of the LED during testing. To resolve this issue, I removed the resistors and found that the LED brightness improved.



WIRE THE LEDS

STEP - 3

1. Take one jumper wire and attach it from the Anode end of the first LED (row 3) and attach it to the digital output on the Arduino at space 2.
2. Take the next jumper wire and attach it from the Anode end of the second LED (row 7) and attach it to the digital output on the Arduino at 3.
3. Take the next jumper wire and attach it from the Anode end of the third LED (row 11) and attach it to the digital output on the Arduino at space 4.
4. Take the next jumper wire and attach it from the Anode end of the fourth LED (row 15) and attach it to the digital output on the Arduino at space 5.
5. Take the next jumper wire and attach it from the Anode end of the fifth LED (row 19) and attach it to the digital output on the Arduino at space 6.
6. Take the next jumper wire and attach it from the Anode end of the sixth LED (row 23) and attach it to the digital output on the Arduino at space 7.

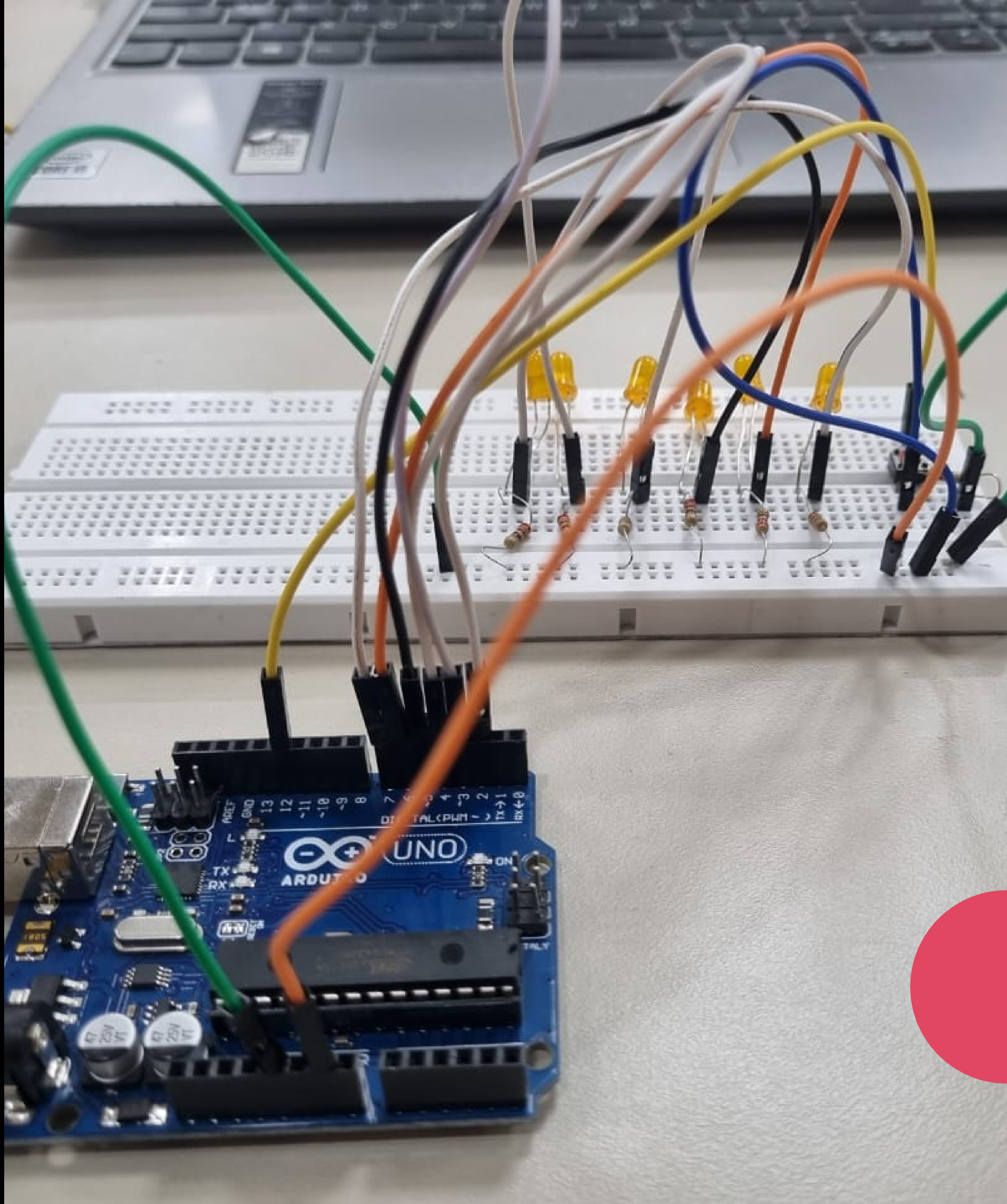


WIRE THE BUTTON AND GROUND

STEP - 4

1. Take one jumper wire and attach it on the bottom right (row 61, rail G) by the button. Attach the other end of the wire to the digital output on the Arduino at space 12.
2. Take another jumper wire and attach it next to the resistor in the space at row 64, rail C. Attach the other end to the ground rail (-, blue).
3. Take another jumper wire and attach it to the top left side of the button (row 59, rail D). Attach the other end to the positive rail.

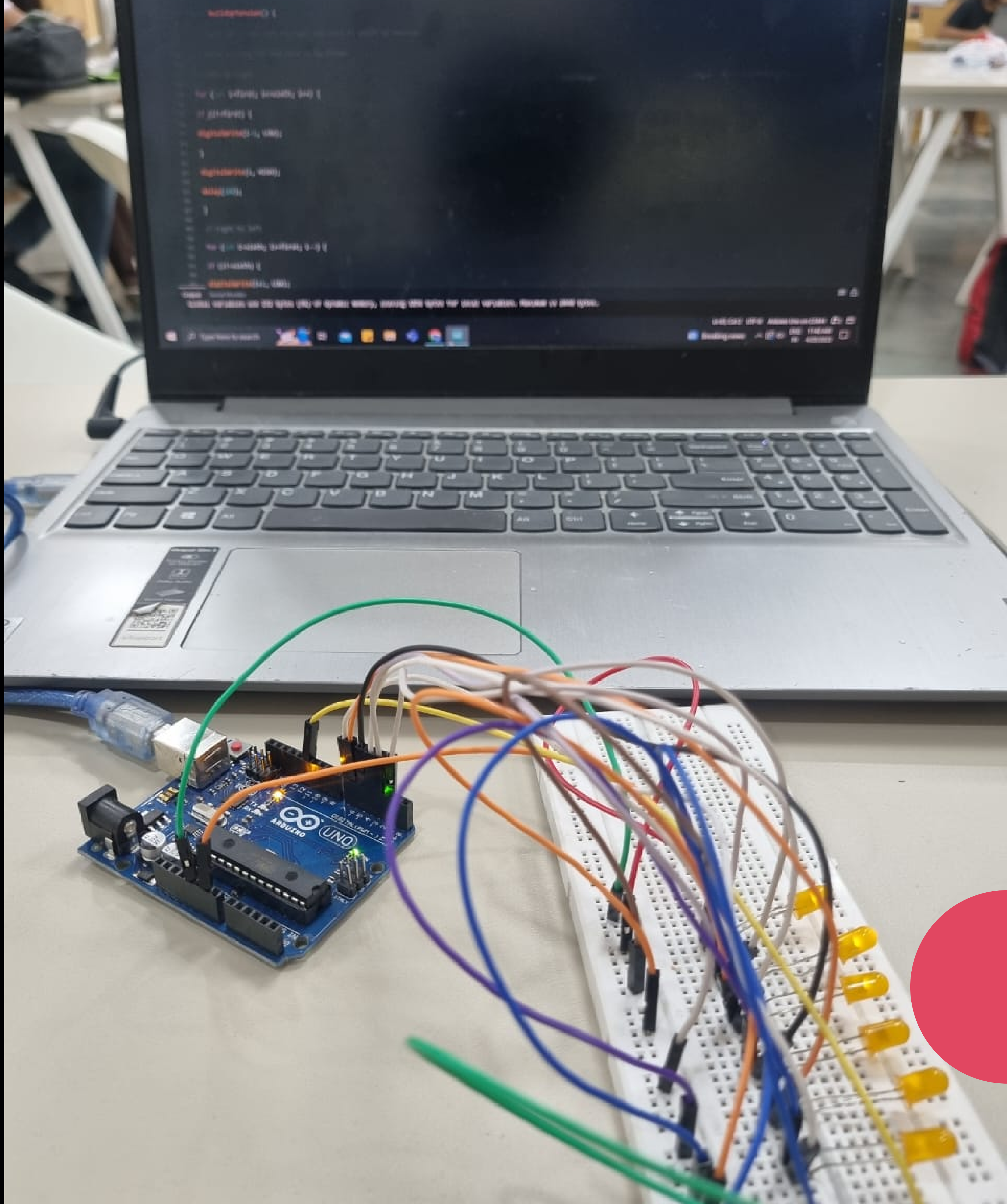
1. Take a jumper wire and attach it to the Ground port on the Arduino. Attach the other end to the ground rail on the breadboard.
2. Take another jumper wire and attach it to the 5V port on the Arduino. Attach the other end to the positive rail.



CONNECT TO LAPTOP AND UPLOAD CODE

STEP – 5

The fifth step is to write the code for the digital dice project and upload it to the Arduino board using the Arduino IDE software. This can be done by selecting the correct board type and serial port, and then clicking on the "Upload" button.



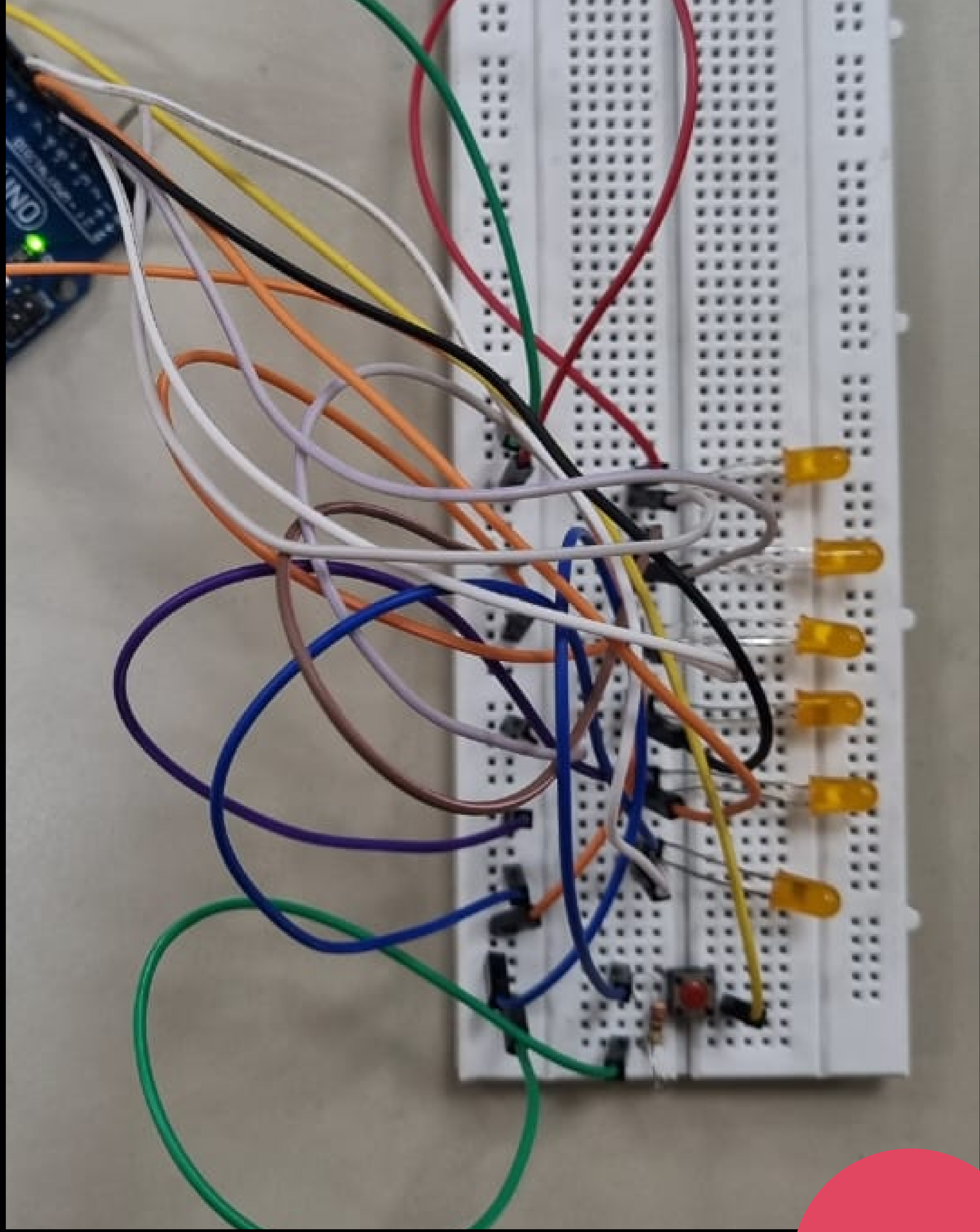
CODE FOR THE PROJECT

```
// set to 1 if we're debugging
#define DEBUG 0

// 6 consecutive digital pins for the LEDs
int first = 2;
int second = 3;
int third = 4;
int fourth = 5;
int fifth = 6;
int sixth = 7;

// pin for the button switch
int button = 12;
// value to check state of button switch
int pressed = 0;

void setup() {
// set all LED pins to OUTPUT
for (int i=first; i<=sixth; i++) {
pinMode(i, OUTPUT);
}
// set buttin pin to INPUT
pinMode(button, INPUT);
// initialize random seed by noise from analog pin 0 (should be unconnected)
randomSeed(analogRead(0));

// if we're debugging, connect to serial
#ifdef DEBUG
Serial.begin(9600);
#endif

}

void buildUpTension() {
// light LEDs from left to right and back to build up tension
// while waiting for the dice to be thrown
// left to right
for (int i=first; i<=sixth; i++) {
if (i!=first) {
digitalWrite(i-1, LOW);
}
digitalWrite(i, HIGH);
delay(100);
}
// right to left
for (int i=sixth; i>=first; i--) {
if (i!=sixth) {
digitalWrite(i+1, LOW);
}
digitalWrite(i, HIGH);
delay(100);
}
}

void showNumber(int number) {
digitalWrite(first, HIGH);
if (number >= 2) {
digitalWrite(second, HIGH);
}
if (number >= 3) {
digitalWrite(third, HIGH);
}
if (number >= 4) {
digitalWrite(fourth, HIGH);
}
if (number >= 5) {
digitalWrite(fifth, HIGH);
}
if (number == 6) {
digitalWrite(sixth, HIGH);
}
}

int throwDice() {
// get a random number in the range [1,6]
int randNumber = random(1,7);
#ifdef DEBUG
Serial.println(randNumber);
#endif
return randNumber;
}

void setAllLEDs(int value) {
for (int i=first; i<=sixth; i++) {
digitalWrite(i, value);
}
}

void loop() {
// if button is pressed - throw the dice
pressed = digitalRead(button);

if (pressed == HIGH) {
// remove previous number
setAllLEDs(LOW);
buildUpTension();
int thrownNumber = throwDice();
showNumber(thrownNumber);
}

}
```

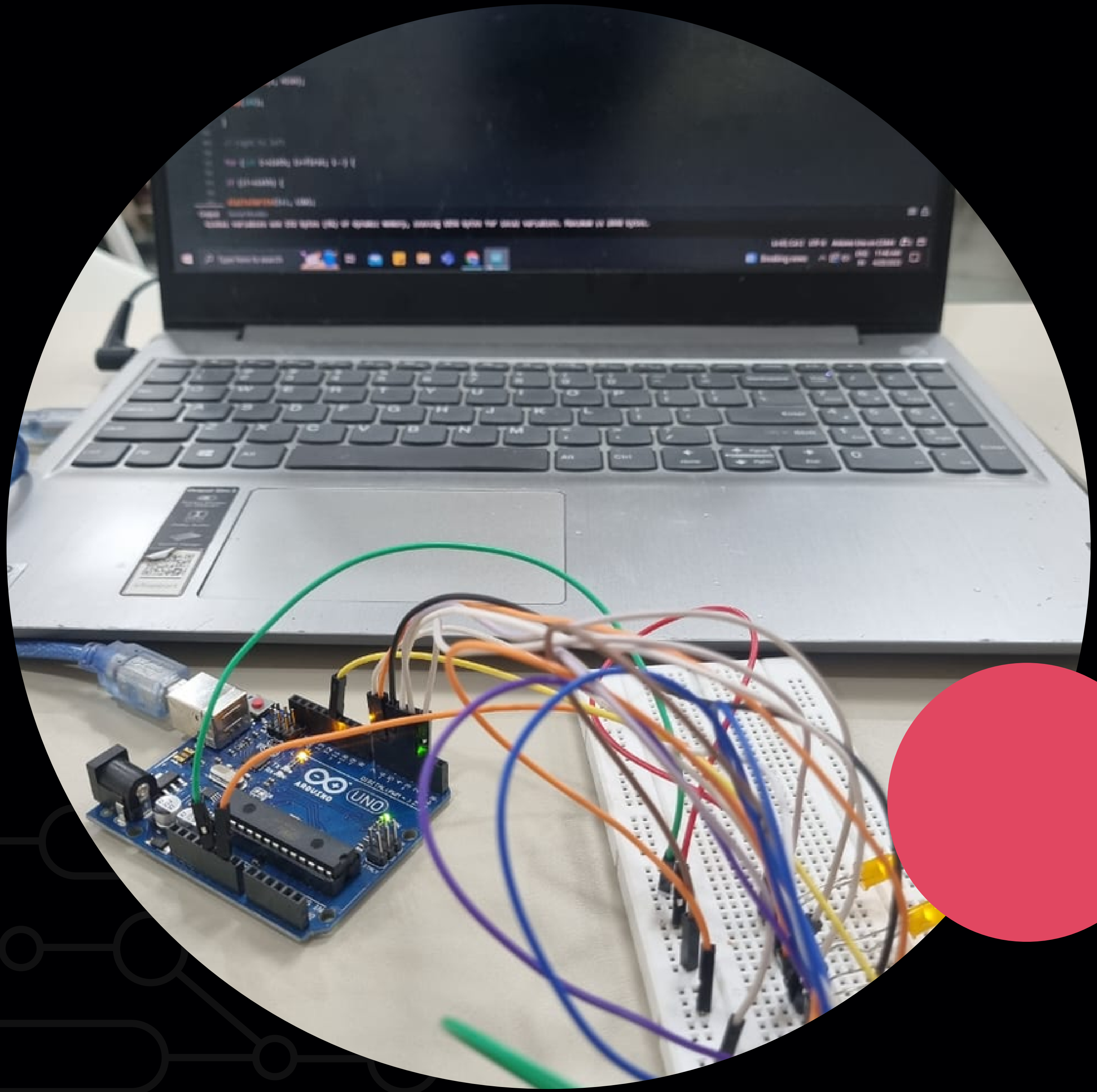
PROBLEMS FACED

Resistor causing the LED to dim

The issue was that when I added a resistor for each LED in the circuit, it caused a voltage drop across the resistor which reduced the amount of current flowing through the LED, thus dimming its brightness. This happened because resistors are designed to resist the flow of current and the amount of resistance they offer can affect the flow of current in the circuit. So, to resolve this issue, I removed the resistors and found that the LED brightness improved as the current flowing through them increased.



MICROCONTROLLER PROGRAMMING



THANK YOU

Daksh Kumbhat
Bdes 1st Year
2022DD14769
Department of Design