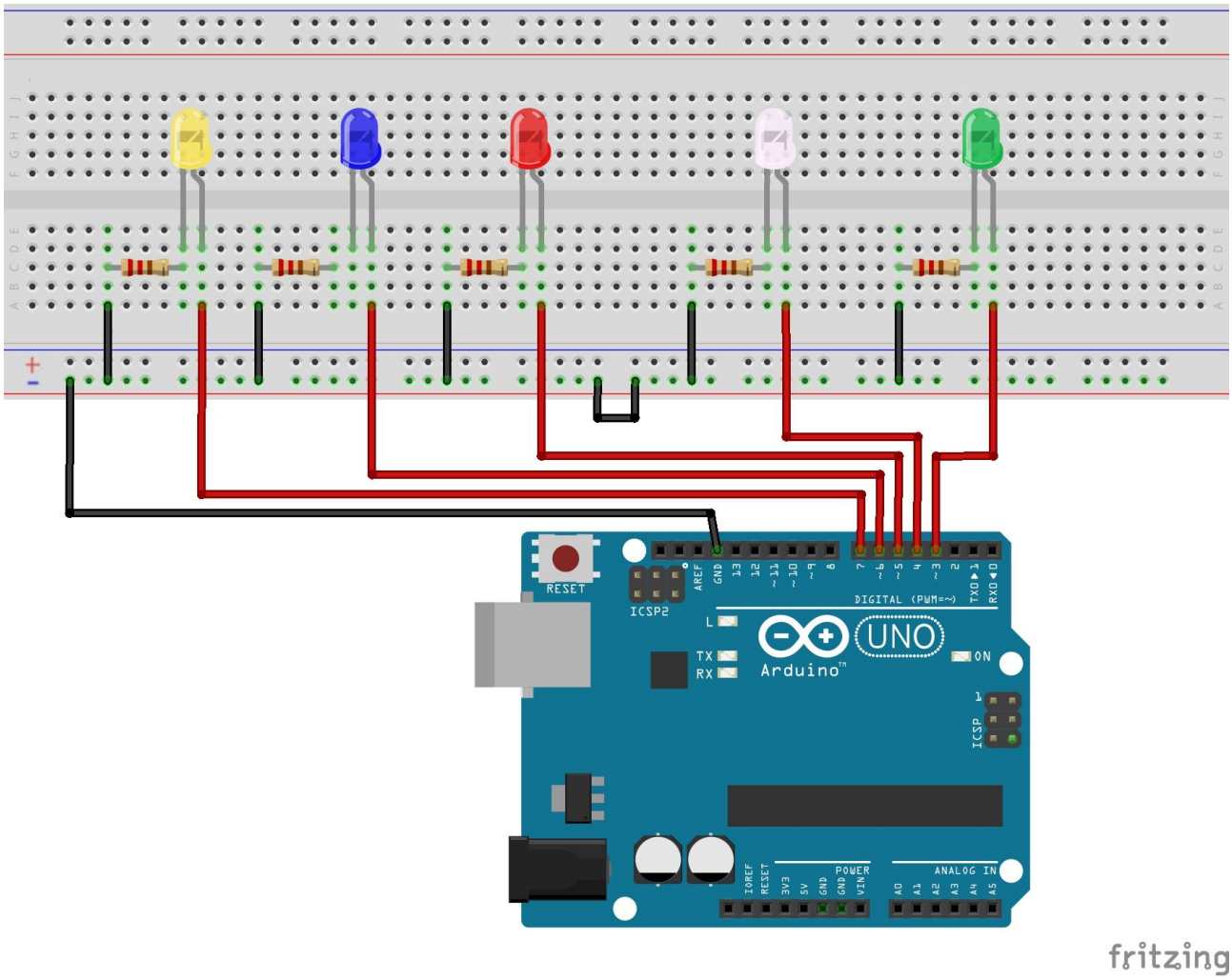
Lauflicht mit blinkenden LEDs

Benötigte Bauteile:

- 5 LEDs
- 5 Widerstände > 100 Ω
- Leitungsdrähte

Baue die Schaltung auf.



Um das zu realisieren, wären viele Programmzeilen nötig, die immer wieder die gleichen Befehle ausführen.

Die Programmiersprache bietet die Möglichkeit, häufig verwendete Befehle in einem gesonderten Programmteil (einer Methode) „auszulagern“.

Jede Methode muss außerhalb des setup- und des loop-Teils stehen.

Beispiel:

```
void LEDBlinken()
{
  digitalWrite(LED, HIGH);
  delay(500);
  digitalWrite(LED, LOW);
  delay(500);
}
```

Die Methode LEDBlinken kann jetzt beliebig oft im loop-Teil aufgerufeń werden.

```
LEDBlinken();
```

Methoden können auch mit Parametern aufgerufen werden:

Beispiel:

```
void LEDBlinken(int LEDNummer, int LeuchtDauer)
{
  digitalWrite(LEDNummer, HIGH);
  delay(Leuchtdauer);
  digitalWrite(LEDNummer, LOW);
  delay(Leuchtdauer);
}
```

Der Aufruf sieht dann so aus:

```
LEDBlinken(6, 1000);
```

Die LED an Pin 6 blinkt eine Sekunde.

Die LEDs werden wieder in einem Array angelegt:

```
int LED[5] = {3, 4, 5, 6, 7};
```

Zusätzlich werden noch Variablen für die Nummer der LED und die Anzahl der Blinkvorgänge benötigt:

```
int AnzahlBlinken;

int LEDNummer;
```

Im setup-Teil erledigt wie beim vorherigen Programm eine for-Schleife die Zuordnung der LEDs.

```
void setup()
{
  for (int i = 0; i <= 4; i ++)
  {
    pinMode(LED[i], OUTPUT);
  }
}
```

Definiere eine Methode LEDBlinken, der als Variable die Nummer der LED und die Anzahl der Blinkvorgänge übergeben werden:

```
void LEDBlinken(int LEDNummer, int Anzahl)
{
  for (int i = 0; i <= Anzahl; i ++)
  {
    digitalWrite(LEDNummer, HIGH);
    delay(100);
    digitalWrite(LEDNummer, LOW);
    delay(100);
  }
}
```

Im loop-Teil wird die Anzahl der Blinkvorgänge zufällig bestimmt und die Methode LEDBlinken mit der Nummer der LED und der Anzahl der Blinkvorgänge aufgerufen:

```
void loop()
{
  for (int i = 0; i <= 4; i ++)
  {
    // Anzahl des Blinkens der LEDs zufällig bestimmen
    AnzahlBlinken = random(1, 5);
    /* aktuelle LED i einschalten
      → Methode LEDBlinken aufrufen
    */
    LEDBlinken(LED[i], AnzahlBlinken);
  }
}
```