

Interaktive Kunst mit einem Potentiometer

Bei dieser Station sollst du dein eigenes interaktives Kunstwerk mit Hilfe eines Potentiometers erschaffen. Dabei kannst du einen in Processing programmierten Code mit dem Potentiometer steuern.

Ein Beispiel für ein so geschaffenes Kunstwerk kannst du in Abbildung 1 sehen. Bearbeite die folgenden Aufgaben, um dein eigenes interaktives Werk zu kreieren.

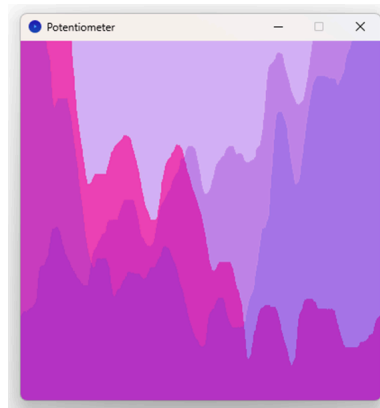


Abbildung 1: Beispiel für ein mit Hilfe eines Potentiometers geschaffenes Kunstwerk.

- 1 Lies dir die **Seiten 48 - 52** im Begleitheft durch. Folge der Schritt-für-Schritt-Anleitung und stelle eine Verbindung zwischen Processing und der Arduino-IDE her.
- 2 Lies dir die Erklärungen für die neuen Programmierbefehle auf den **Seiten 53 + 54** im Begleitheft durch. Beschreibe kurz, wofür die folgenden Befehle verwendet werden.

```
import processing.serial.*
```

erforderlich für die Datenübertragung zwischen Processing und der Arduino-IDE

```
Serial myPort
```

bildet eine Klasse für das Objekt myPort, worüber Daten gesendet und empfangen werden

```
int data
```

speichert die empfangenen Daten

```
myPort = new Serial(this, Serial.list()[0], 9600)
```

wählt den entsprechenden Port (hier: 0) aus, bei der Standard-Baud-Rate von 9600

```
Serial.begin(9600)
```

startet die Datenübertragung mit der Standard-Baud-Rate

```
Serial.println()
```

sendet gewünschtes OUTPUT-Signal an den Port

- 3 Lies dir die **Seite 55** im Begleitheft durch und verschalte mit Hilfe der gelesenen Informationen das Potentiometer.

- 4 Nutze die zuvor erlernten Informationen und schreibe den Arduino-Code und lade ihn hoch, um die Daten des Potentiometers an Processing senden zu können.



Hier wird der Wert des analogen Pins als OUTPUT-Signal an den Port gesendet.

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  // sendet den Wert des analogen Pins A0:  
  Serial.println(analogRead(A0));  
}
```

- 5 Nun ist es an der Zeit den Processing-Code zu schreiben. Nutze dafür das gesamte gesammelte Wissen aus den vorherigen Projekten.
- Im Sketch-Fenster soll eine farbige Linie entlang der x-Richtung von links nach rechts wandern.
 - Wenn die Linie am rechten Rand des Fensters angekommen ist, soll eine neue farbige Linie wieder am linken Rand starten. Du kannst nach deinen Wünschen beliebig viele Farben verwenden.
 - In y-Richtung soll sich die Linie entsprechen der Werte, die über das Potentiometer an Processing gesendet werden, bewegen. Integriere dafür die in Aufgabe 2 beschriebenen Programmierbefehle in deinen Code.
- 6 Lies dir Seite 56 im Begleitheft durch und ergänze die serialEvent-Funktion in deinem Code.

```
import processing.serial.*;

Serial myPort;
int xBlack = 0;
int xFarbel = 0;
float data;

void setup ()
{
  size(400,400);
  myPort = new Serial(this, Serial.list()[0], 9600);
  background(0);
}

void black()
{
  stroke(0,0,0);
  line(xBlack, height, xBlack, height - data);
  xBlack++;
}

void Farbel()
{
  stroke(0,0,0);
  line(xFarbel, height, xFarbel, height - data);
  xFarbel++;
}

void draw ()
{
  black();

  if (xBlack >= width)
  {
    Farbel();
  }
}

void serialEvent (Serial myPort)
{
  String inString = myPort.readStringUntil('\n');
  if (inString != null)
  {
    data = float(inString);
    println(data);
  }
}
```