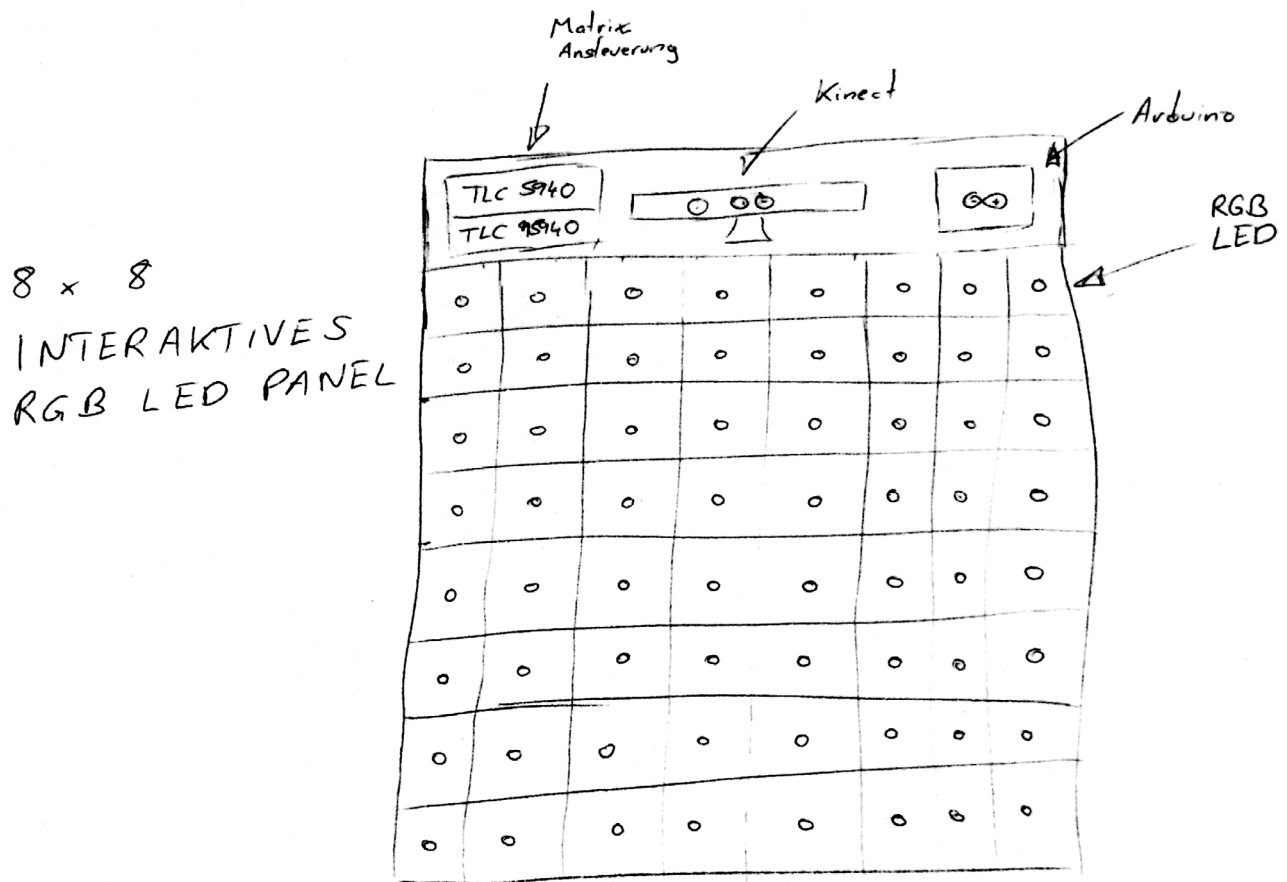


# 8x8 Interactive RGB LED Panel



"Die Idee"

## Interaktive Installationen

Konzept Semesterprojekt

Werner Gundacker | mt131021

## Ziel des Projektes

Das Ziel des Projektes ist ein interaktives 8x8 RGB LED Panel herzustellen und zu programmieren, welches mit Hilfe der Kinect Kamera gesteuert werden kann. Wenn man vor dem Panel steht und die Hand bewegt, soll das Panel entsprechend leuchten, bzw. auch wenn man vorbei geht soll die Kamera die Bewegung erkennen und das Panel entsprechend ansteuern.

## Geplanter Aufbau und Bestandteile

Der Aufbau des Panel wird mit Holz, Plexiglas und einer großen Lochrasterplatine erfolgen. Die Größe eines „Pixels“ wird 5x5cm sein. Das ergibt dann eine Gesamtgröße von etwa 40x40 cm.

### Benötigte Materialien (voraussichtlich)

- Arduino Uno (bereits vorhanden)
- Microsoft Kinect
- TLC5940 (16 Kanal Led Treiber) 12 Stück
- 64 Stück RGB LEDs
- 10kOhm Widerstand
- 2kOhm Widerstände 12 Stück
- 50x50cm Lochrasterplatine
- 50x50 Plexiglasplatte (matt, lichtdurchlässig)

## Mögliche Erweiterungen

Wenn es sich von der Zeit her ausgeht, würde ich das Panel gerne noch um einige Features erweitern.

### Buttons zur Steuerung:

- Ein-/Ausschalten
- Motion Tracking ein/ausschalten
- Start vom Demo des Displays (Display leuchtet entsprechend oder zeigt Text)
- Fernsehsimulator starten „Einbrecherschreck“

## Änderungen vom geplanten Aufbau

Da der Aufwand und der Preis für den Aufbau der großen RGB Matrix (64 Leds ca. 80€) zu groß war wurde diese im Laufe des Projekes durch einen fertigen 8x8 RGB LED Matrix Baustein ersetzt. Diese wird wiederum von einem Colors Shield v1.1 angesteuert. Die große Matrix könnte ebenfalls von diesem Colors Shield angesteuert werden. Anstatt der Kinect Kamera wird die Webcam des Laptops verwendet.

### Die somit benötigten Materialien

- Laptop mit Processing und Webcam
- Arduino Uno
- 2 Taster
- 2x 10kOhm Widerstand
- 8x8 RGB LED Matrix
- Colors Shield v1.1
- 2 Steckbretter
- div. Drähte
- USB Kabel

## Umgesetzte Erweiterungen

### Buttons zur Steuerung:

#### Button 1: Menü/Pause

Wenn man den Button gedrückt hält, leuchtet eine grüne LED, welche anzeigt, dass man sich im Menü befindet. Es leuchtet auch eine rote LED entsprechend des aktuellen Modus.

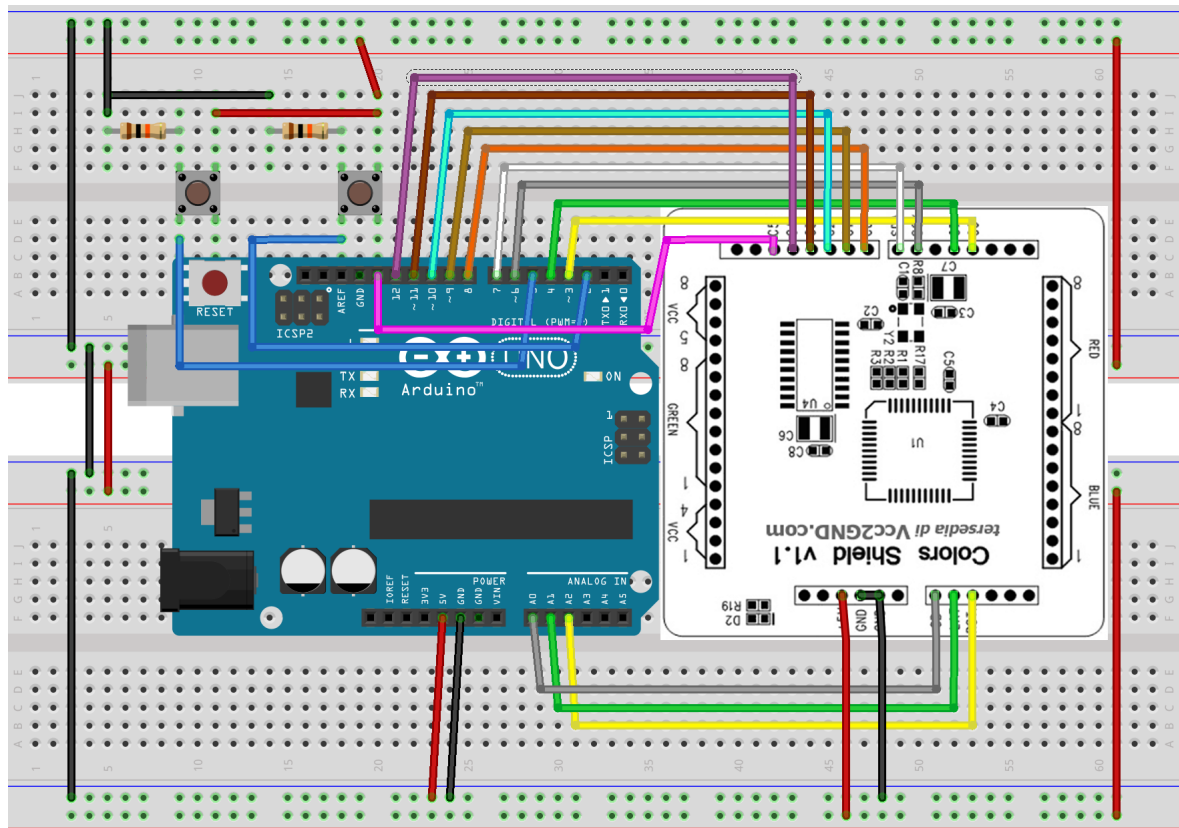
#### Button 2: Moduswahl

Button 1 muss gedrückt sein, dann kann man mit dem Button 2 die einzelnen Modus auswählen. Aktueller Modus wird mit einer roten LED an der entsprechenden Position angezeigt.

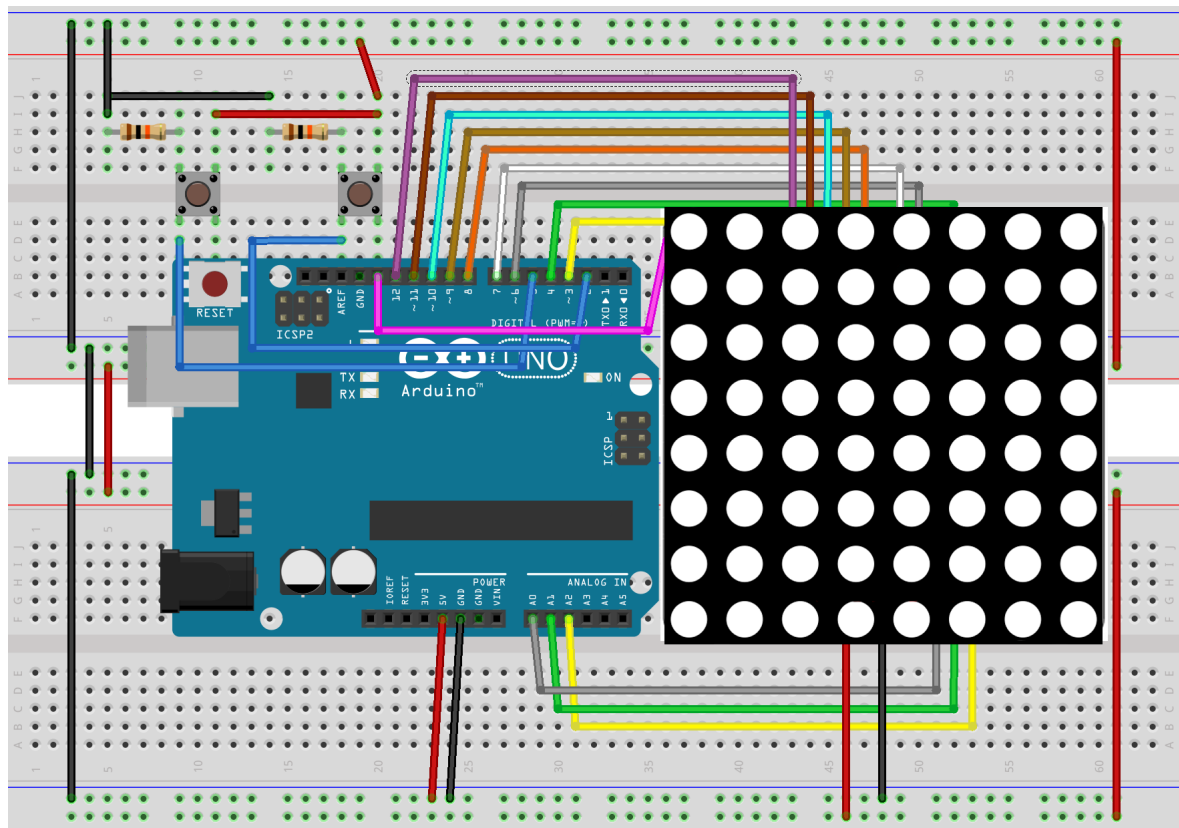
### Mögliche Modi:

- Modus 1: zufällige Farben
- Modus 2: Motion Tracking (Processing über serielle Schnittstelle)
- Modus 3: Farbverlauf

## Skizze des Aufbaus

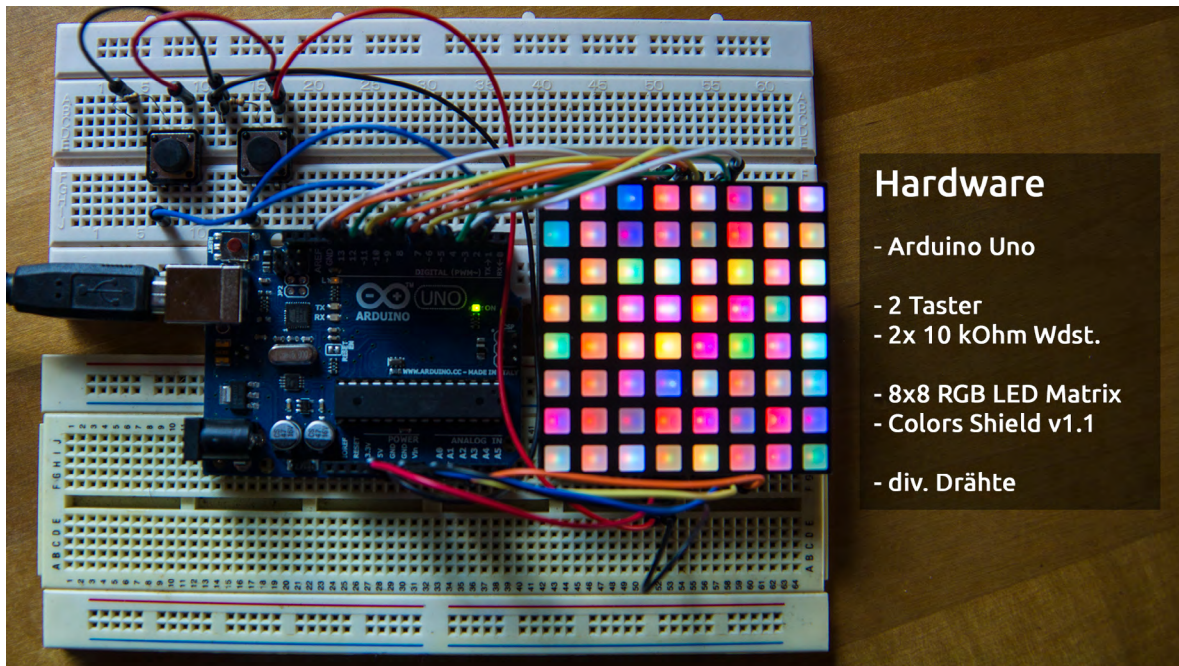


## Skizze mit LED Matrix

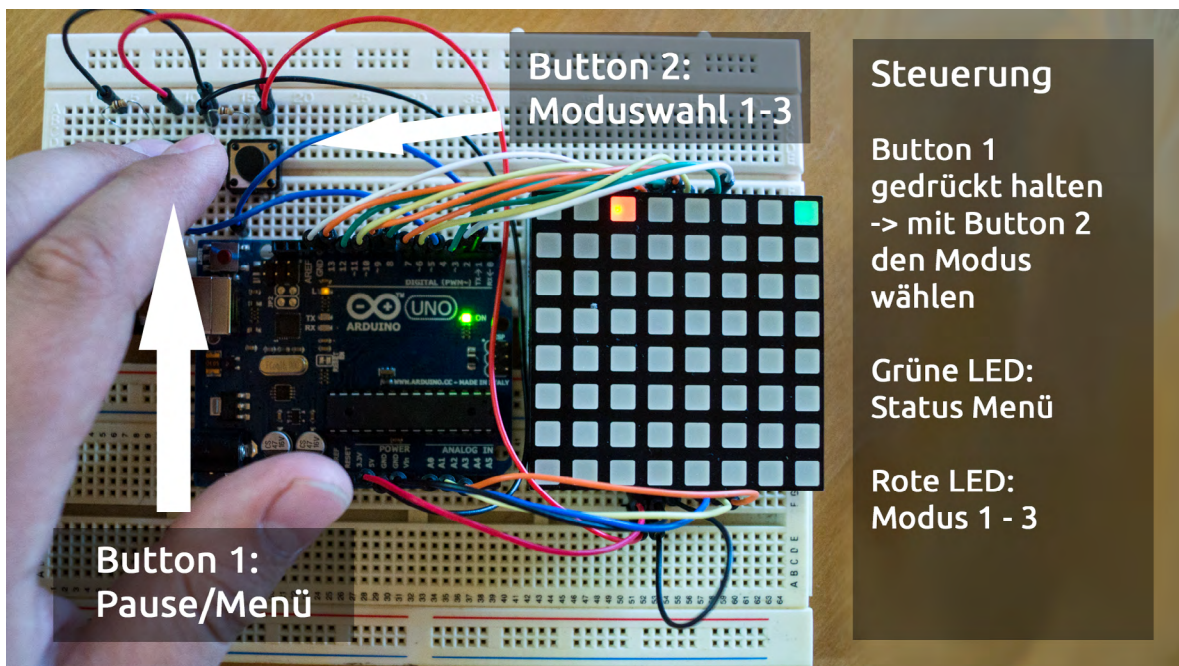




## Foto vom Aufbau



## Erklärung der Steuerung



## Source Code und Promotion Video

Source Code auf GitHub: [https://github.com/Woid4tler/IAI\\_Interactive\\_RGB-LED-Matrix](https://github.com/Woid4tler/IAI_Interactive_RGB-LED-Matrix)

Promotion Video auf Youtube: <https://youtu.be/uX4H2EwhsQQ>

# Installationsanleitung

Die Buttons, Widerstände und das Colors Shield entsprechend der Skizze aufbauen. Die RGB Matrix kann direkt auf das Colors Shield gesetzt werden.

Danach den Arduino mittels USB Kabel mit dem Laptop verbinden. Die Arduino IDE starten und das Projekt „RGBLedMatrix\_Arduino.ino“ öffnen und auf den Arduino übertragen.

Dann können die Modi 1 und 3 bereits getestet werden. Will man das Motion Tracking ausprobieren, muss man am Laptop Processing öffnen und das Projekt „MotionTracking\_Processing.pde“ starten. Hierbei kann es sein das man die serielle Schnittstelle noch anpassen muss, das passiert in Zeile 36.

```
serialPort = new Serial(this, Serial.list()[7], 9600);
```

Hier entsprechend den Arduino auswählen. Mit `println(Serial.list());` kann man sich eine Liste aller verfügbaren seriellen Ports ausgeben lassen. Dann muss nur noch die Zahl 7 entsprechend angepasst werden.

Wenn das Processing-Programm gestartet wird, startet der Arduino automatisch neu, und geht in Modus 2 für das Motion Tracking. Nun wird im Webcambild Bewegung erkannt. Das Bild ist dabei in 64 Quadranten (8x8) unterteilt. Wenn in einem Quadrant eine gewisse Bewegung stattfindet wird der Quadrant Blau hinterlegt und die entsprechende LED auf der Matrix leuchtet ebenfalls in Blau.

## Links

Colors Shield v1.1: <http://www.exp-tech.de/arduino-rgb-led-matrix-driver-shield-colors-shield-v1-1>

8x8 RGB Led Matrix: <http://www.exp-tech.de/iteadstudio-60mm-square-8x8-led-matrix-rgb-square>