

# CAOS Projekt : 2 x 6 x 12 LED RGB Quader

Joey Zraggen, Moritz Würth, Viktor Gsteiger

January 11, 2020

TODO: BILD VOM PROJEKTERZEUGNIS

# Inhaltsverzeichnis

<b>1</b>	<b>Ziel des Projektes</b>	<b>2</b>
<b>2</b>	<b>Verwendete Materialien</b>	<b>2</b>
2.1	RGB LED's	2
2.2	Schieberegister	2
2.3	Transistoren	2
2.4	Sensoren	2
<b>3</b>	<b>Materialkosten</b>	<b>3</b>
<b>4</b>	<b>Design</b>	<b>3</b>
<b>5</b>	<b>Features</b>	<b>3</b>
5.1	Alphabet	3
5.2	Stars	3
5.3	Firework	4
5.4	Welcome	4
5.5	Game of Life	4
<b>6</b>	<b>Sensoren</b>	<b>4</b>
6.0.1	Temperatur- und Luftfeuchtigkeitssensor	4
6.0.2	Infrarot-sensor	4
<b>7</b>	<b>Aufbau des zweidimensionalen Bildschirms</b>	<b>5</b>
7.1	Bildschirm	5
7.2	Schieberegister	5
7.3	Probleme beim Aufbau	5
<b>8</b>	<b>Zusammenfassung</b>	<b>5</b>

# 1 Ziel des Projektes

Die Grundidee von unserem Projekt war es zuerst einen 5 x 5 x 5 LED RGB Würfel zu bauen, jedoch haben wir uns dann im Prozess der Entscheidungsfindung dazu entschieden, etwas anderes zu bauen was sich vom "typischen" LED RGB Würfel unterscheidet. Deshalb haben wir die Dimensionen ein wenig angepasst, sodass wir am Ende einen dreidimensionalen Bildschirm haben werden. Damit wird es uns immer noch möglich sein, gute 3D-Effekte anzeigen zu können. Die Anordnung der LED's kann als Koordinatensystem verstanden werden, bei dem man mittels x- und y-Koordinaten auf die einzelnen LED's zugreifen kann. Dies erleichtert es massiv, Effekte zu schreiben. Das Projekt lässt einen grossen Spielraum bezüglich Ideen zu. Sollte man also selbst Ideen haben, welche sich gut zu unserem Projekt ergänzen würden, kann man diese ohne weitere Probleme implementieren. Die Grundsoftware hat sehr einfache Schnittstellen, mit denen man einfach weitere Effekte schreiben kann.

## 2 Verwendete Materialien

Im Folgenden ist eine Auflistung der verschiedenen Materialien, welche für das Projekt verwendet wurden.

### 2.1 RGB LED's

Um den LED-Quader ein wenig bunter gestalten zu können wurden RGB's verwendet.:

- LED RGB Common Cathode 4-Pin F5 5MM Diode

### 2.2 Schieberegister

Schieberegister erweisen sich für ein solches Projekt als äusserst nützlich, um die vielen Pins der einzelnen RGB LED's ansprechen zu können.

Wir haben uns dabei an schon bereits existierende Projekte orientiert und uns für die 74HC595 8-bit Schieberegister entschieden. Diese erweisen sich in der Programmierung als intuitiv, weil man einfach mit Byte-Arrays arbeiten kann, um die einzelnen Schieberegister mit Informationen zu "befüllen". Weiter existieren für diese Schieberegister vorgefertigte Libraries, um die Kommunikation zu erleichtern. Dies nimmt einen nicht die Denkarbeit ab, wie man die LED's anordnen will und wie man die verschiedenen Farben anspricht, jedoch erleichtert es die direkte Kommunikation.

### 2.3 Transistoren

Anfangs wollten wir für eine garantierte Langlebigkeit unseres Projektes Transistoren verwenden, aber nachdem wir uns bezüglich der Notwendigkeit von Transistoren für unser Projekt informierten haben wir uns letztenendlich dazu entschieden keine zu verwenden.

### 2.4 Sensoren

Für das Projekt wurden zusätzlich Sensoren verwendet, um weitere Features zu gewährleisten. Folgende Sensoren stehen zur Auswahl:

- Temperatur und Luftfeuchtigkeitssensor:

- Infrarot-Sensor:
- Real-time clock:

### 3 Materialkosten

Produkt	Menge	Preis	Total
RGB LED's	250x	0.072.-	18.-
Kabel (verschiedene)	15m	0.83.-	12.50
Widerstände	440x	0.036.-	16.-
74hc595	57x	0.50.-	30.-
Sensoren (verschiedene)	3x	3.30.-	10.-
Lochrasterplatine (verschiedene)	2x	6.00.-	12.-
Kuperdraht (Blau)	2 Rollen	6.80.-	13.60.-
Isolierband	1 Rolle	1.80.-	1.80.-
			<b>113.90.-</b>

### 4 Design

Wir haben uns überlegt eine kleine Box zu bauen, welcher die ganze Verkabelung in einem kleinen Raum im hinteren Bereich von dieser Box zu verstecken, damit es schöner aussieht. Wir haben uns für das Material Holz entschieden und werden die ganze Box schwarz färben, da dann der Effekt der RGB LED's entsprechend besser zur Geltung kommen. Wir haben schon einige Projekte gesehen, welche einen LED RGB Cube gebaut haben, bei denen auch eine Art Box gebaut wurde und wir fanden diese Idee gut, sodass wir uns für unser Projekt danach orientierten.

### 5 Features

#### 5.1 Alphabet

Durch die Klasse Alphabet ist es einfach möglich, auf dem Bildschirm Buchstaben und Sätze darzustellen. Die Klasse unterscheidet nicht zwischen klein und Grossbuchstaben und kann das gesamte Alphabet plus ein paar Sonderzeichen plus alle Zahlen darstellen. Die einzelnen Buchstaben sind alle Hardgecoded und deshalb macht diese Klasse Sinn, weil man dies sonst jeweils einzeln machen müsste. Die Klasse Alphabet wird vom Willkommens Effekt, von den temperature effects und vom clock Effekt benutzt und ist dadurch ein sehr zentraler Effekt. Die Buchstaben werden dabei von links nach rechts einer nach dem anderen in den Buchstaben-Buffer geladen, dann jeweils, sofern man nicht beim letzten Buchstaben ist, drei mal auf den Hauptbildschirm rausgeschiftet und nach jedem einzelnen shiften angezeigt. Beim letzten Buchstaben wird dieser so lange nach links geschiftet, bis er vom Bildschirm verschwindet und die Kontrolle wird zurück an die rufer Methode gegeben.

#### 5.2 Stars

Im Effekt stars werden zufällig Sterne erzeugt, welche entweder von links nach rechts oder umgekehrt fliegen. Wenn sich zwei Sterne treffen explodieren diese in einem Effekt. Die Sterne haben alle verschiedene Farben, welche auch zufällig sind.

Die Sterne sind alle eigene structs, welche die nötigen Informationen beinhalten. In der Update methode werden alle existierenden Sterne fortgeführt in ihrem Pfad.

### **5.3 Firework**

Der Feuerwerk Effekt funktioniert ähnlich wie der Sternen-Effekt, jedoch fliegen hier die Raketen nur von unten nach oben und explodieren auf einer zufälligen Höhe. Die Farbe der Raketen ist auch zufällig.

Eine Rakete ist ein struct, welche alle nötigen Informationen beinhaltet.

### **5.4 Welcome**

Der Willkommens Effekt ist ein sehr einfacher Effekt, welcher einfach mit der Methode Alphabet eine Willkommensnachricht auf dem Bildschirm anzeigt und einmal nach dem Aufstarten angezeigt wird.

### **5.5 Game of Life**

Der Game of Life Effekt ist eine repräsentation des berühmten Game of Life des britischen Mathematikers John Horton Conway. Dabei kommen zufällig Zellen zum Leben und sterben beziehungsweise pflanzen sich nach bestimmten Regeln fort. Dabei können sie auch über den Bildschirmrand einen Effekt haben und zählen dann zu den Nachbarn auf der anderen Seite. Für mehr Informationen empfiehlt es sich, den Wikipedia Artikel zum Game of Life zu lesen.

## **6 Sensoren**

### **6.0.1 Temperatur- und Luftfeuchtigkeitssensor**

Der Temperatur- und Luftfeuchtigkeitssensor, welchen wir für unser Projekt verwenden ist vom Typ DHT11. Dies ist ein eher einfacher Sensor, welcher jedoch mehr als genügt für unsere Zwecke. Dieser liest die aktuelle Temperatur und Luftfeuchtigkeit jede Sekunde und diese Information wird dann auf dem Bildschirm angezeigt. Mit Hilfe dieser Informationen wird dann aus einem von drei Effekten ausgewählt. Entweder wird dann ein Regen, Wolken oder eine Sonne auf dem zweidimensionalen Bildschirm angezeigt. Die Berechnung hierfür ist jedoch sehr einfach und rudimentär, weshalb der Bildschirm sich nicht für eine genaue Wettervorhersage eignet.

### **6.0.2 Infrarot-sensor**

Der Infrarot-sensor empfängt Input von einer externen Fernbedienung und man kann damit zwischen den verschiedenen Effekten wechseln. Dabei werden bei allen Effekten regelmässig die Informationen vom Sensor abgefragt und dann wird dies abgearbeitet. Wir haben uns hierbei extra gegen einen Interrupt gesteuerten Dateninput entschieden, weil dies das Effekthandling erschweren würde. So ist maximal ein Effekt zur gleichen Zeit am laufen und bricht ab, sobald man mit der Fernbedienung einen anderen Effekt auswählt.

Weiter kann man mit der Fernbedienung das Snake Spiel steuern und so die längste Schlange kreieren.

## **7 Aufbau des zweidimensionalen Bildschirmes**

### **7.1 Bildschirm**

### **7.2 Schieberegister**

Wir bereits angesprochen helfen uns die Schieberegister die vielen verschiedenen Pins der RGB LED's anzusprechen. Die Schieberegister sind auf zwei Platinen angeordnet und sind mit einander mittels den entsprechenden outputs miteinander verknüpft. Wichtig war es dabei zu beachten, dass der Daten in- und output korrekt verlötet wurden, weil wir die Teilaufgaben unseres Projektes untereinander aufgeteilt haben und die Anordnung der Schieberegister mit dem Code übereinstimmen müssen.

### **7.3 Probleme beim Aufbau**

Durch die anspruchsvolle Steuerung von 216 einzelnen Pins kommt es natürlich zu Schwierigkeiten. Unsere erste Schwierigkeit war, dass wir zuerst mit einer zu kleinen und einzelnen Platine gearbeitet haben. Dies hatte zur Folge, dass die Lötarbeit schnell unübersichtlich und ungenau wurde. Auch war es so extrem schwer, die Lötstellen sauber zu verlöten. Wir haben uns nach einigem herumprobieren dafür entschieden, von einer auf zwei Platinen umzusteigen, was die Arbeit massiv vereinfacht hat.

## **8 Zusammenfassung**

Quellen:

-