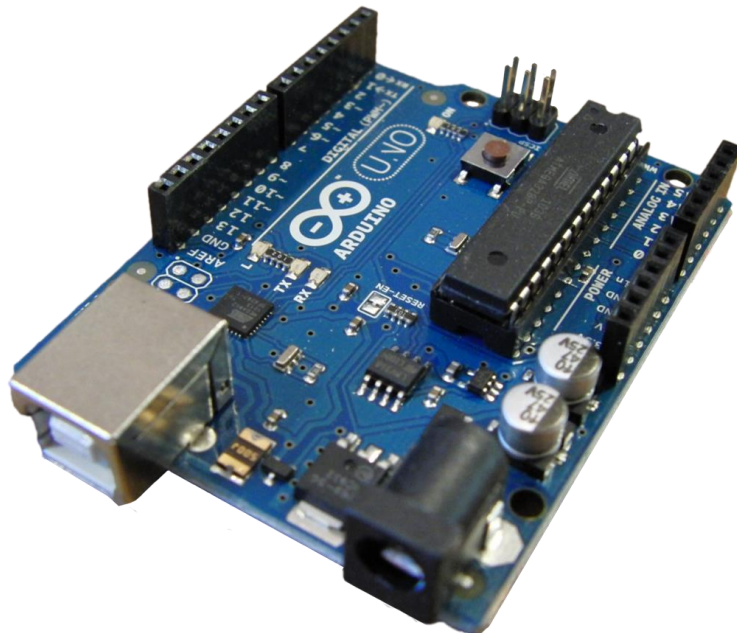


# LEDruino



## Inhalt

1	Einleitung .....	4
1.1	Abgrenzungen .....	4
1.1.1	Genutzte Hilfsmittel für die Umsetzung des Projektes .....	4
2	Pflichtenheft.....	6
2.1	Auftrag.....	6
2.1.1	Projektgrobidee.....	6
2.2	Team Organisation .....	6
2.2.1	Kriterien für das Endprodukt.....	7
3	Planung .....	9
3.1	Terminplanung .....	9
3.2	Konzept Endprodukt.....	10
3.2.1	Einfluss der Sensoren auf die WS2812 Matrix .....	10
3.2.2	Modi .....	10
3.3	Hardware.....	11
3.4	Case .....	11
3.5	Elektronisches Schema.....	12
3.6	Code .....	13
3.6.1	Coding Standards .....	13
3.6.2	Versionisierung.....	13
3.6.3	Module .....	13
3.6.4	Flussdiagramm .....	14
4	Inbetriebnahme .....	15
4.1	Code .....	15
4.1.1	Ablauf der Umsetzung des Codes .....	15
4.1.2	Genutzte Bibliotheken .....	16
4.1.3	Aufbau der Module .....	16
4.1.4	Funktionen .....	17
4.1.5	Besonderheiten im Code.....	17
4.2	Verkabelung .....	18
4.2.1	Stromversorgung LED Matrix .....	18
4.2.2	Implementierung der IR-Fernbedienung .....	18
4.2.3	Implementierung der Stromversorgung ohne USB-Kabel.....	19
4.2.4	Flussdiagramme .....	20
5	Kontrolle.....	21
5.1	Testprotokoll.....	21

5.2	Soll-Ist vergleich .....	23
6	Anhang .....	25
6.1	Quellen.....	25
7	Kurzfazit.....	25

# 1 Einleitung

Das vorliegende Projekt dient dazu, ein interaktives Kunstobjekt auf Basis eines Arduino-Systems zu entwickeln.

Ziel ist es, technische Kenntnisse in der Programmierung, Elektronik und Gestaltung zu kombinieren, um ein ästhetisches und funktionales Ergebnis zu schaffen, das auf Umweltreize reagiert. Die Arbeit verbindet Kreativität mit technischem Know-how und bietet die Möglichkeit, verschiedene Technologien praktisch anzuwenden.

## 1.1 Abgrenzungen

### 1.1.1 Genutzte Hilfsmittel für die Umsetzung des Projektes

In diesem Abschnitt werden die unterstützenden Ressourcen und Werkzeuge aufgelistet, die während des Projekts genutzt wurden. Diese Hilfsmittel reichen von sozialen Kontakten und Experten bis hin zu digitalen Tools, die für die Konzeption, Umsetzung und Dokumentation hilfreich waren.

#### Soziale Kontakte

Soziale Kontakte spielen eine zentrale Rolle bei der Umsetzung technischer Projekte. Sie bieten Unterstützung und Wissen, das über die eigene Expertise hinausgeht. Für dieses Projekt wurden folgende Stakeholder konsultiert:

Bei Technischen Fragen werden folgende Stakeholder konsultiert.

Schwager	Elektrotechnik Bachelor Selbst erfahren in Bereich Arduino
Lehrer	Fachliche Erfahrung Feedback zu Projektumsetzung

#### Internetressourcen

YouTube	Videos zur Klärung von Fragen und bestehende Herangehensweisen
Weitere	Webseiten wie GitHub für Tools und Coding Beispiele

#### Bestehendes Wissen

Dieser Punkt beschreibt das bereits vorhandene Wissen, das für das Projekt genutzt werden kann. Dazu zählen Programmierkenntnisse, Kenntnisse in Logik und Digitaltechnik sowie Grundlagen der 3D-Modellierung.

Arduino Programmierung Grundwissen	Gelernt im Unterricht, sehr grundsätzlich
Grundwissen Logik und Digitaltechnik	Aus Teko Unterricht
Programmierungswissen	Wissen in anderen Programmiersprachen
Basiswissen 3D Modellierungsprogramme	

## LLMs

Generative KI, insbesondere ChatGPT, wurde während des Projekts als Assistenztool in folgenden Bereichen genutzt:

- **Beratung bei Textpassagen:** Kleinere Textbausteine wurden übernommen
- **Technische Fragen:** Schnelle Klärung von Unsicherheiten in Elektronik, Arduino-Programmierung und Sensorik.
- **Ideenfindung:** Unterstützung bei den Realistischen Möglichkeiten eigener Ideen
- **Codeentwicklung und Workflow:**
  - **Ideenfindung und erste Umsetzung:** Gemeinsam mit ChatGPT wurden grundlegende Codeideen entwickelt und ein erstes funktionierendes Grundgerüst erstellt.
  - **Iterative Anpassung:** Der Code wurde schrittweise verbessert und neue Anforderungen oder Fehler wurden unmittelbar in die Weiterentwicklung integriert.
  - **Fehleranalyse und Debugging:** Bei auftretenden Problemen unterstützte ChatGPT durch gezielte Debugging-Hilfen und Optimierungsvorschläge.
  - **Feinschliff und Validierung:** Nach den Anpassungen wurde der Code final überprüft, um sicherzustellen, dass alle Funktionen wie geplant umgesetzt wurden.

Die Zusammenarbeit mit ChatGPT ermöglichte einen dynamischen Workflow, der sowohl Kreativität als auch technische Präzision förderte. Die menschliche Expertise blieb dabei entscheidend, insbesondere bei der praktischen Umsetzung und Validierung.

## 2 Pflichtenheft

### 2.1 Auftrag

Hier werden die grundlegenden Ziele, Anforderungen und Rahmenbedingungen des Projekts beschrieben. Hier wird geklärt, welche Absicht hinter dem Projekt steht und welche Ressourcen zur Verfügung stehen.

Auftraggeber	Auftragnehmer
Teko – Christian Meier	Valentin Küng – Student HF Systemtechnik

Durch Teko gegebene Vorgabe eines Projekts in der Mikrocomputertechnik zu machen. Was genau umgesetzt wird wurde nicht angegeben. Es soll sich im Rahmen der Mikrocomputertechnik befinden. Entsprechend wurden Vorschläge mit Arduino oder Raspberry Pi gemacht.

Die Vorgaben für die Dokumentation wurden in einer per Bewertungskriterien im Projekt umgesetzt.

#### 2.1.1 Projektgrobidee

In diesem Unterkapitel wird die Vision des Projekts beschrieben. Die Idee ist es, ein Kunstobjekt zu schaffen, das durch den Einsatz moderner Technologien die Umgebung interpretiert und darstellt. Die sensorische Interaktion bietet sowohl einen funktionalen als auch einen künstlerischen Mehrwert.

Der Auftrag wurde vom Studenten selbst definiert. Durch Recherche im Internet und eigene Interessebereiche wurde folgende Idee und ihre Definition gesetzt:

- Eine 16x16 LED-Matrix die Animiert wird durch einen Arduino
- Sensoren, die die Umgebung wahrnehmen und so die Animation bereitstellen
- Die LEDs sollen die Umgebung darstellen mit folgenden Eingängen
  - Nähe einer Person oder eines Gegenstandes
  - Geräusche
  - Wärme
  - Helligkeit

### 2.2 Team Organisation

Das Projekt wurde einzeln von mir allein ausgeführt. Entsprechend ist jede Rolle bei einer Person.

### 2.2.1 Kriterien für das Endprodukt

Die Soll-Situation beschreibt die Zielsetzung des Projekts. Die Muss-Kriterien definieren die minimalen Anforderungen, während optionale Kriterien zusätzliche Funktionen oder Verbesserungen darstellen.

#### Hardware

Muss Kriterien	Kann Kriterien	Abgrenzungskriterien
LEDs zeigen korrekte Farben und Helligkeit in der Startanimation an ohne sichtbare Verzögerung	Ultraschallsensoreingänge werden für komplexere Animationen kombiniert	Nicht wetterfest
Sensoren geben Daten der Realität entsprechend aus	Eigengestaltetes 3D gedrucktes Gehäuse	Genauigkeit der Sensoren ist nicht mehr als 500ms

#### Software

Muss Kriterien	Kann Kriterien	Abgrenzungskriterien
Startanimation beim Einschalten der Matrix läuft direkt über auf Grundmodus Regenbogen		Es werden keine batteriebetriebenen Komponenten genutzt.
Ultraschallfernbedienung wechselt zuverlässig zwischen allen Modi	Parameter der Sensorendaten können modifiziert per Pfeiltasten oben und unten Rechts der Fernbedienung werden.	
Ultraschallsensoren erkennen Objekte von drei verschiedenen Umgebungszonen (Links, Mitte, Rechts)	Parameter der Animationen können modifiziert per Pfeiltasten Links Rechts der Fernbedienung werden.	
Modus zur Erkennung der Lichtverhältnisse nach Lichtgrad bei eingeschaltetem Licht (dunkel) und ausgeschaltetem Licht (hell)		
Modus für Erkennung der Umgebungsgeräusche mit Genauigkeit auf Variation der Geräusche 1-10	Musikvisualisierung kann ein- oder ausgeschaltet werden per Fernbedienung	Reagiert nicht ausserhalb des angegebenen Bereichs

Modus für Erkennung der Temperatur von 10 bis 40 Grad	Der Farbwechsel kann schrittweise in 5 Grad Schritten geändert werden	Temperaturwerte ausserhalb des Bereichs werden nicht berücksichtigt
Modus für alle Sensoren der alle Inputs erkennbar wiedergibt	Sensorprioritäten können manuell über die Fernbedienung gesetzt werden	



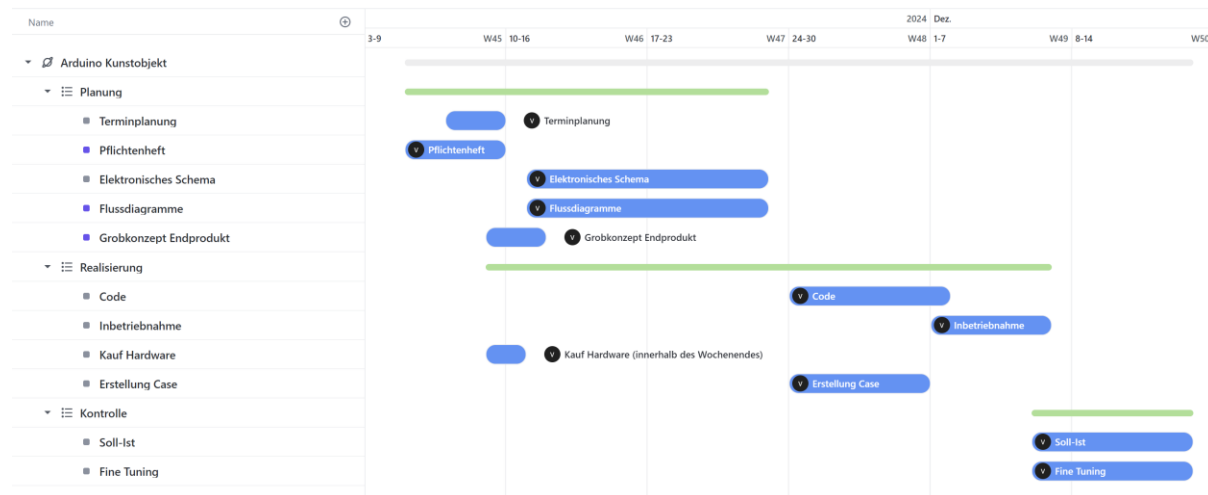
## 3 Planung

In der Planungsphase werden die zeitlichen und technischen Schritte des Projekts detailliert beschrieben.

### 3.1 Terminplanung

In der Planungsphase werden die zeitlichen und technischen Schritte des Projekts detailliert beschrieben.

#### GANTT



#### Tabelle

Task	Fertigstellungsdatum	Meilenstein
Terminplanung	09.11.2024	
Pflichtenheft	09.11.2024	
Grobkonzept Endprodukt	09.11.2024	X
Kauf Hardware	09.11.2024	
Flussdiagramme	16.11.2024	
Elektronisches Schema	23.11.2024	X
Code	23.11.2024	
Inbetriebnahme	30.11.2024	
Tests und Fehlerbehebung	14.12.2024	X

## 3.2 Konzept Endprodukt

Das Endprodukt wird in einzelne Module zerlegt und schrittweise entwickelt. Am Ende erfolgt die Integration der Module zu einem Gesamtprodukt.

### 3.2.1 Einfluss der Sensoren auf die WS2812 Matrix

Dieser Abschnitt beschreibt die geplante Interaktion der Sensoren mit der LED-Matrix und die spezifischen Effekte, die durch die Sensoren ausgelöst werden sollen.

Sensor	Animation
Ultraschallsensor Rechts	Farbänderung von rechts nach links Änderung der Sättigung
Ultraschallsensor Links	Farbänderung von links nach rechts Änderung der Sättigung
Ultraschallsensor Mitte	Farbänderung von unten nach oben Änderung der Sättigung
Lichtsensoren	Farbhelligkeitsänderung Helligkeitsänderung der LEDs
Mikrofon	Lässt Wellen schlagen in die LED Darstellung Musikmodus
Temperatursensor	Ab 25 grad von Farbtemperatur blau zu Farbtemperatur rot. Langsamer überlauf

### 3.2.2 Modi

Modus	Verhalten	Remote Taste
Grundmodus	Regenbogenmuster ohne Sensoren gesteuerte Animation	0
Ultraschnall	Kombiniert den Input der 3 Ultraschall Sensoren zu der Farbänderung, die besteht	1
Lichtsensoren	Farbhelligkeit und Helligkeit der LEDs	2
Mikrofon Umgebungsgeräusche	Von der Mitte gegen aussen Bei Geräusche werden LEDs in der Farbe geändert	3
Mikrofon Musik	Musikvisualisierungsmodus	4
Temperatursensor	Farbwärme	5
Alle Sensoren	Kombination aus obigem	6
Grundfarbton Rot	Grundfarbe ist Rot	7
Grundfarbton Blau	Grundfarbe ist Blau	8
Grundfarbton Grün	Grundfarbe ist Grün	9

### 3.3 Hardware

Die vorhandene Hardware im Arduino Starter-Kit wird aufgelistet. Sie bildet die Basis für die technische Umsetzung und beeinflusst die Planung des Projekts.

#### Bestehende Hardware

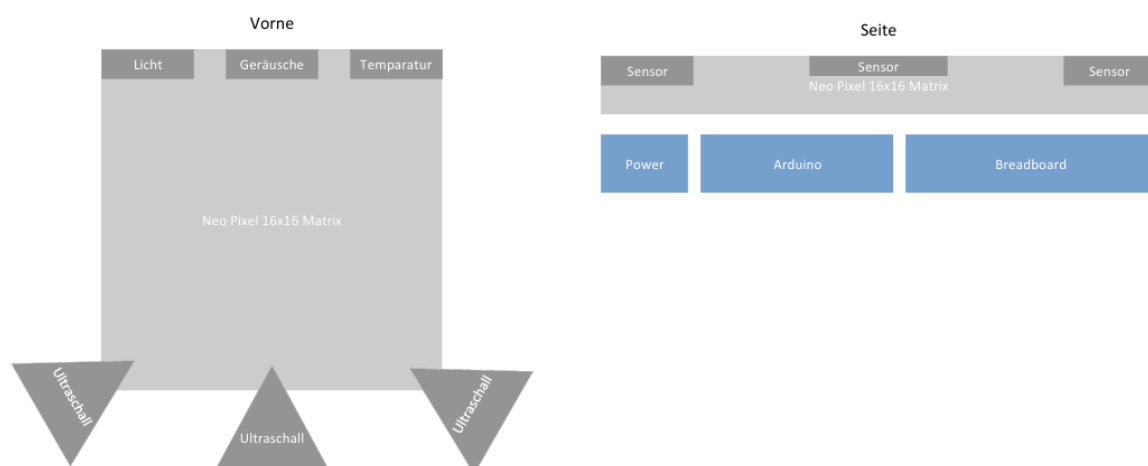
<b>Arduino Board Fundiono Mega 2560</b>	<b>1x</b>
<b>Breadboard</b>	<b>1x</b>
<b>Lichtsensord</b>	<b>1x</b>
<b>Ultraschall Distanzmesser HC-SR05</b>	<b>1x</b>
<b>Wärmesensor</b>	<b>1x</b>
<b>Infrarot Fernbedienung und Empfänger</b>	<b>1x</b>

#### Zu kaufende Hardware

<b>NeoPixel FlexMatrix 1cyc6x16 – 25dfafsd6 x WS2812B RGB LED</b>	<b>1x</b>
<b>Ultraschall Distanzmesser HC-SR05</b>	<b>2x</b>
<b>Mikrofon Geräuschsensor KY-037</b>	<b>1x</b>
<b>DC-Buchse Female Barrel Jack 5.5mm / 2.1mm mit Schraubklemmen</b>	<b>1x</b>
<b>5V DC 3000mA Stecknetzteil AC/DC-Adapter 5.5mm/2.1mm Stecker</b>	<b>1x</b>

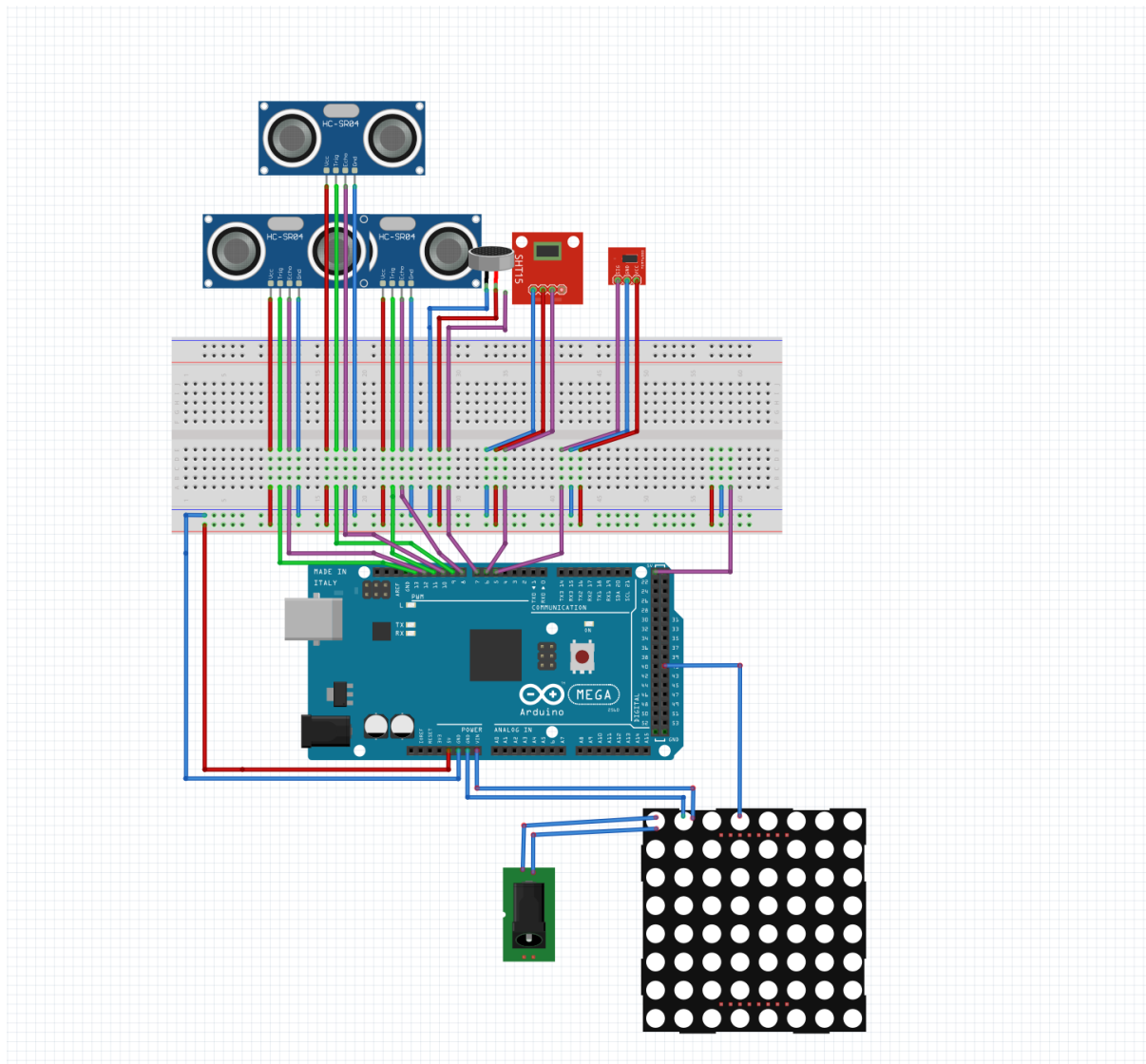
### 3.4 Case

Es folgt ein visuell vereinfachtes Mock-Up des Case das erstellt wird. Es dient dazu die Position der Elemente zu bestimmen innerhalb des Case.



### 3.5 Elektronisches Schema

Hier wird die Schaltung des Projekts dokumentiert, einschließlich aller Verbindungen zwischen den Komponenten und der Arduino-Plattform.



## 3.6 Code

Hier wird die Planung der Entwicklung des Codes thematisiert. Der Fokus liegt auf der Strukturierung und Umsetzung der Programmierung für die Steuerung der Hardware und LEDs.

### 3.6.1 Coding Standards

#### Trennung und Logik

Erstellte Module wird klar nach Logik der Funktionaler Ebene des Programms gemacht:

- Logische Verarbeitung
- Ein- und Ausgänge

Die config.h Datei dient zur Anpassung von allen Relevanten Parameter für Erweiterung.

#### Funktionen als weitere Modulisierungsebene

Das Programm verwendet Funktionen, um alle wichtigen Teile des Programm dezentral anzusteuern. Diese werden in den Config Files der einzelnen Module definiert.

#### Namenskonventionen

Alle Variablen im Config File sind in CAPSLOCK und bei mehreren Wörtern mit Unterstrich definiert

Alle Funktionen sind im Camel-Case mit erstem Buchstaben klein definiert.

### 3.6.2 Versionisierung

Die Versionisierung wird per Ordner gemacht. Abgelegt wird diese Versionisierung auf GitHub.

### 3.6.3 Module

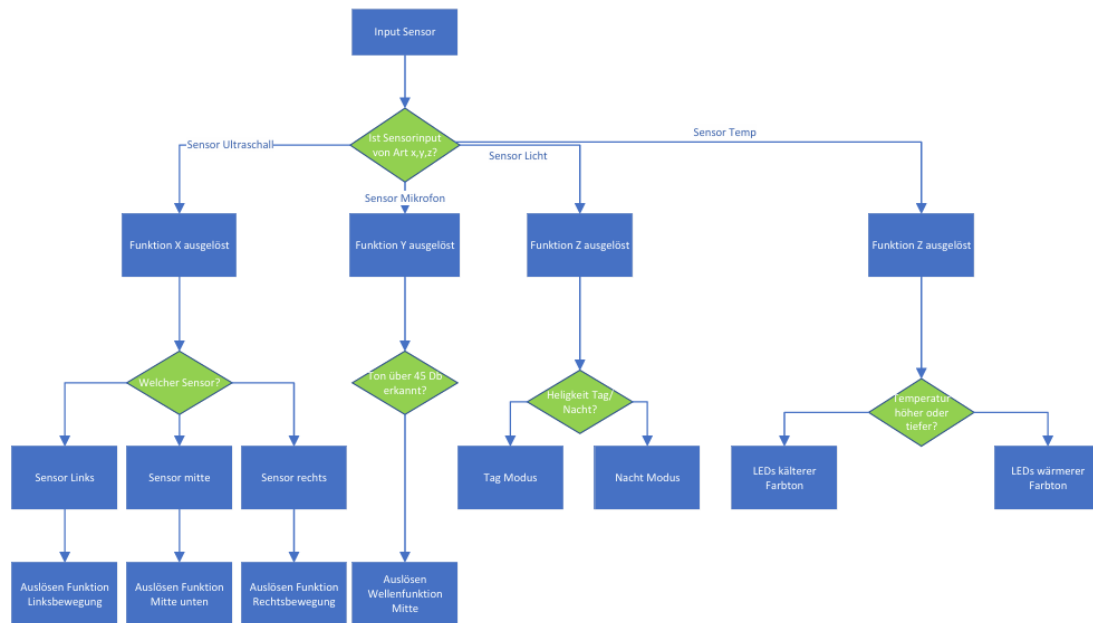
Die Module stellen einzelne Funktionseinheiten dar, die unabhängig voneinander entwickelt und getestet werden, bevor sie in das Gesamtsystem integriert werden.

Die Module werden in logische Ebenen eingeteilt

- Logische Verarbeitung und Operative Kontrolle
  - Oberste Ebene die den Hauptprogrammablauf definiert
  - Verarbeitung der Sensoren Signale und LED Output
- Steuerung von Input und Output
  - Alle Sensoren
    - Ultraschall
    - Mikrophon
    - Licht
    - Temperatur
  - Das LED Ausgangs Modul
  - Infrarot Fernbedienung Input

### 3.6.4 Flussdiagramm

Es folgt das Flussdiagramm nach der Planung.



## 4 Inbetriebnahme

Die Realisierungsphase umfasst die praktische Umsetzung des Projekts, von der Programmierung bis zur physischen Montage.

### 4.1 Code

#### 4.1.1 Ablauf der Umsetzung des Codes

Das LLM GPT-4o und GPT-o1 wurden konstant zum Draften des Codes und schreiben gebraucht. Code selbstgeschrieben wurde nur minimal. Es wurde der Code iterativ getestet und angepasst durch die Menschliche Komponente.

Iteration	Umsetzungsschritt
1	Ultraschallsensor mit 8 LEDs
2	Infrarot Remote für Wählen der Modi
3	Implementierung der Codestruktur des Hauptprojekt mit 16x16 Grid
4	Anpassung, Troubleshooting und Verbesserung Module
5	
6	Mikrophon mit 8 LEDs
7	Temperatursensor mit Serieller Ausgabe
8	Lichtsensoren mit Serieller Ausgabe
9	Zusammenfügen aller Sensoren in einem Programm
10	Anpassung, Troubleshooting und Verbesserung der Zusammengeführten Module
11	Zusammengefügender Modus ausfeilen
12	Einzelne Sensoren Modi aus Zusammengeführten Modus generieren
13	Abkoppeln der Variablen für Animations- und Sensorparameter
14	Troubleshooting und Verbesserungen

### 4.1.2 Genutzte Bibliotheken

- FastLED 3.9.4
- Keypad 3.1.1

### 4.1.3 Aufbau der Module

#### Logische Verarbeitung und Operative Kontrolle

main.ino  
sensorLedManager.cpp  
modes.cpp

#### Verarbeitung von Input und Output

ledModule.cpp  
remoteControl.cpp  
ultrasonicSensorModule.cpp  
microphoneModule.cpp  
tempSensorModule.cpp  
lightSensorModule.cpp

Modulname	Zweck	Ebene
main.ino	Low code Hauptebene	Logische und Operative Kontrolle
sensorLedManager	Verwaltung und Zusammenführung der Sensorinputs und Ausgabe auf LEDs	Logische und Operative Kontrolle
modes.cpp	Modi, die Daten der Sensorinputs verschieden verwenden	Logische und Operative Kontrolle
ledModule.cpp	Output auf LEDs	Verarbeitung von Input und Output
ultrasonicSensorModule.cpp	Verwaltung der Ultraschallsensoreingänge	Verarbeitung von Input und Output
microphoneModule.cpp	Eingangsverarbeitung Mikro	Verarbeitung von Input und Output
tempSensorModule.cpp	Eingangsverarbeitung Temperatur	Verarbeitung von Input und Output
lightSensorModule.cpp	Eingangsverarbeitung Licht	Verarbeitung von Input und Output



#### 4.1.4 Funktionen

Modul	Funktionen
ledModule.cpp	XY initLEDs
lightSensorModule.cpp	initLightSensor readLightSensor
microphoneModule.cpp	initMicrophone readMicrophone
modes.cpp	updateMultipleSensorsMode updateUltrasonicMode updateLightSensorMode updateTempSensorMode updateRainbowMode updateMicrophoneMode updateTestPatternMode (für Testzwecke)
remoteControl.cpp	checkRemoteInput
sensorLedManager.cpp	initSensorsAndLEDs readAllSensors updateLEDsBasedOnMode printSensorValues (für Testzwecke)
tempSensorModule.cpp	initTempSensor readTempSensor
ultrasonicSensorModule.cpp	initUltrasonicSensors, smoothUltrasonicValue readUltrasonicDistance
_main.ino	setup, loop (Standarfunktionen)

#### 4.1.5 Besonderheiten im Code

In diesem Abschnitt werden einzelne, ausgewählte Besonderheiten, die im Code verwendet wurden dokumentiert.

##### Millis als nicht Programm-Blockierender „Unterbruch“

In der Read all Sensors Funktion, wird die Millis Funktion in der Iteration verwendet, um Ultraschall sowie Geräusche Priorisiert (5ms) im Gegensatz zu Temperatur und Licht (5000ms) abzufragen.

Das hat den Vorteil, dass die Priorität auf Geräuschen und Ultraschall ist. Das liegt daran, dass es dort schnellere Reaktion auf Veränderung benötigt.

##### XY Funktion

Diese Funktion indiziert das XY Array in dem Zick-Zack Muster, in welchem die LEDs aufgebaut sind. Der Index für jeden LED i ist eine Zahl von 256. Entsprechend ist das Array eindimensional, was Leistungsfähiger ist.

## 4.2 Verkabelung

Hier wird die physische Verbindung der elektronischen Komponenten, inklusive der Verwendung des Breadboards und der Kabel.

### 4.2.1 Stromversorgung LED-Matrix

Es folgt der Ablauf der Inbetriebnahme der Stromversorgung für die LED-Matrix

Iteration	Umsetzung	Ergebnis	Grund
1	Direkte Verbindung mit Arduino	Nur 1 Pixel bekam schwach licht	Zu wenig Leistung auf die Matrix
2	Verbindung mit Klemmadapter	Die Matrix wurde bei 8 Pixel korrekt beleuchtet. Bei Versuchen mit 256 war die Darstellung falsch	Vermutlich erster LED zerstört da kein Kondensator verwendet wurde. Die Matrix war somit zerstört
3	Verbindung mit Klemmadapter und Kondensator dazwischen		
4	Verbindung mit Klemmadapter und Kondensator und GND für Data	Erfolgreich	Datensignal benötigt saubere Referenzspannung

### 4.2.2 Implementierung der IR-Fernbedienung

Iteration	Umsetzung	Ergebnis	Grund
1	Anschluss des IR Remote nach Schaltplan und Übernahme bestehenden Codes	Befehl immer 0	Unklar
2	Kleiner Sketch mit nur dem Receiver und Remote	Befehle werden korrekt dekodiert	Keine Störsignale oder Hauptprogramm
3	Übernahme Code aus Sketch und hinzufügen einer Ausgabe der Rohdaten	Rohdaten sind bei Drücken der Taste immer gleich	Störsignale oder Programm
4	Code angepasst, um Unterbrechungen bei dem Empfang des Signals zu verhindern	Rohdaten sind bei Drücken der Taste immer unterschiedlich	Störsignale oder Programm
5	Als Workaround Keypad angeschlossen und das Modul umgeschrieben	Erfolgreich	Keine Störsignale, andere Verarbeitung der Daten

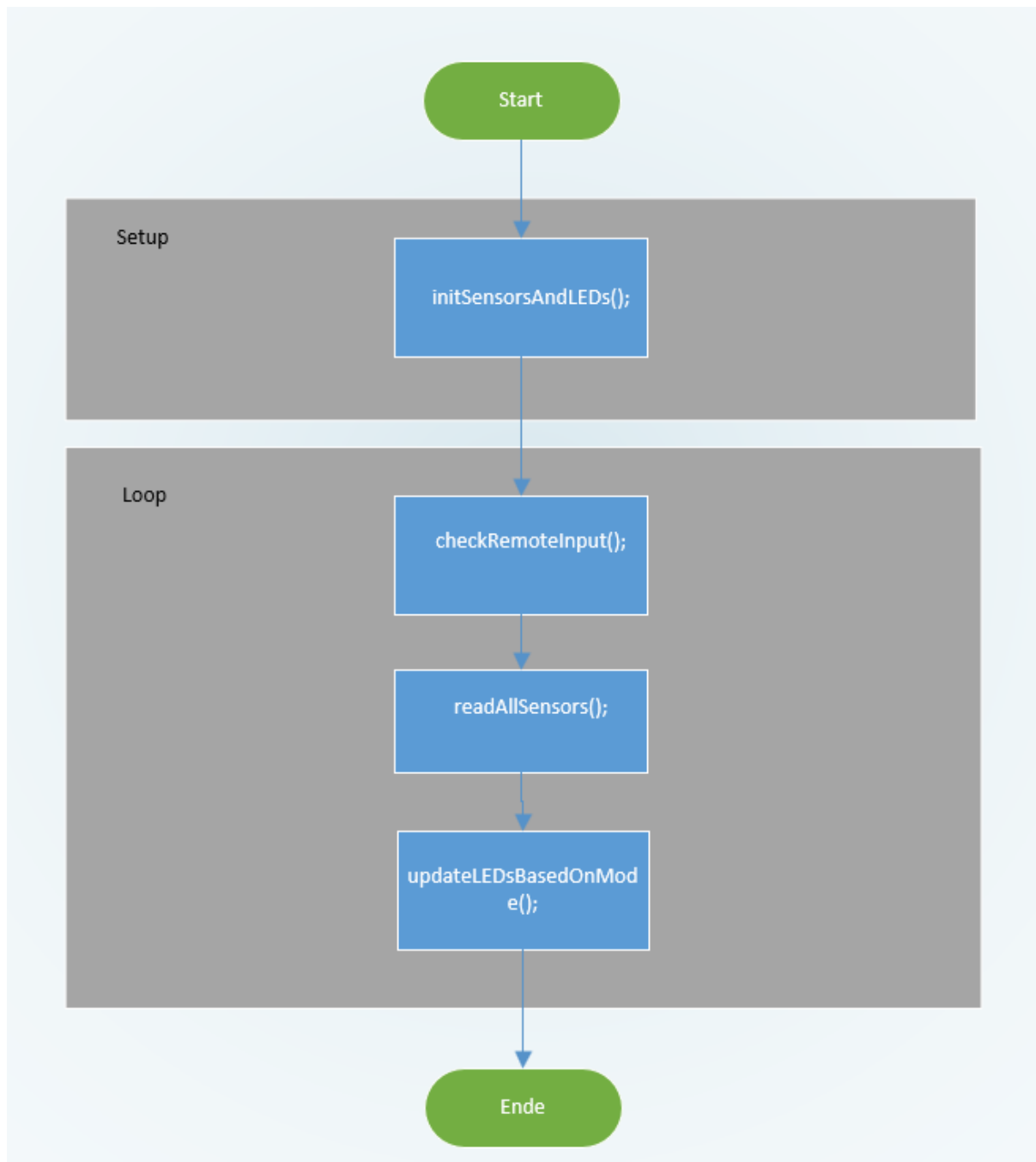
### 4.2.3 Implementierung der Stromversorgung ohne USB-Kabel

Es folgt der Ablauf der Inbetriebnahme der Stromversorgung für die LED-Matrix

Iteration	Umsetzung	Ergebnis	Grund
1	Wegschneiden der Plastik des Stromstecker mit Männlichen Pins und anschrauben des Klemmadapter	Stromversorgung dafür Matrix, jedoch keine Daten und Arduino kein Strom	
2	Anschluss des Arduino an die LED-Matrix per VIN Port und bestehendem GND	Nicht erfolgreich, nach paar Sekunden LED-Darstellung falsch	VIN Erwarte höhere Spannung
3	Anschluss des Arduino an die LED-Matrix per 5V Port und bestehendem GND	Erfolgreich	5V ist der korrekte PIN

#### 4.2.4 Flussdiagramme

Es folgt eine stark vereinfachte Version des ganzen Programms. Das Diagramm, das alles beschreibt ist im Ordner „Grafiken“ zu finden.



## 5 Kontrolle

Die Kontrollphase dient dazu, das Projekt zu bewerten und den Erfolg der Umsetzung zu prüfen.

### 5.1 Testprotokoll

Das Testprotokoll dokumentiert die durchgeführten Tests und deren Ergebnisse. Es wird überprüft, ob die Soll-Kriterien erfüllt wurden.

#### Hardware

Testfall	Erwartetes Ergebnis	Ergebnis	Abweichung
Einstecken der LED-Matrix	Einstecken der LED-Matrix zeigt Startanimation	Erfolgreich	-
Farben und Helligkeit wird nach Definition im Code beim Einschalten gezeigt	Rot Blau Grün wird Pixelweise über die Matrix geschoben	Erfolgreich	Kleine Wiederholung beim Anschalten der Matrix
Ultraschallsensor vorne erkennt, wenn jemand auf vor dem mittigen steht	Der Wert der Nähe ändert sich entsprechend der wirklichen Distanz	Erfolgreich	-

**Code**

<b>Testfall</b>	<b>Erwartetes Ergebnis</b>	<b>Ergebnis</b>	<b>Abweichung</b>
Farben und Helligkeit wird nach Definition im Code beim Einschalten gezeigt	Rot Blau Grün wird Pixelweise über die Matrix geschoben	Erfolgreich	Kleine Wiederholung beim Anschalten der Matrix
Ultraschallsensor vorne erkennt, wenn jemand auf vor dem mittigen steht	Der Wert der Nähe ändert sich entsprechend der wirklichen Distanz	Erfolgreich	-
Ultraschallsensor rechts erkennt, wenn jemand auf vor dem rechten steht	Der Wert der Nähe ändert sich entsprechend der wirklichen Distanz	Erfolgreich	-
Ultraschallsensor links erkennt, wenn jemand auf vor dem linken steht	Der Wert der Nähe ändert sich entsprechend der wirklichen Distanz	Erfolgreich	-
Temperatur wird erkannt entsprechend der Raumtemperatur	Temperatur des Raumes wird nach 5 Sekunden korrekt angezeigt	Teilweise erfolgreich	Temperatur der LED-Farben ändert zufällig. Bei kalter Temperatur jedoch in Richtung Blau und Raumtemperatur Grün.
Geräusch wird nach Lautstärke erkannt	Bei blasen in das Mikrophon erhöht sich der Wert um mindestens 5	Erfolgreich	-
Ausschalten des Licht im Raum	Ändert den Wert der Lichtverhältnisse um mindestens 5	Erfolgreich	-
Infrarotfernbedienung funktioniert korrekt	Drücken der Fernbedienung ändert alle Modi zuverlässig	Keypad als Workaround vorhanden Hiermit erfolgreich	-
Benutzer kann Sensordatenwerte Ändern	Pfeil nach rechts Verstärkt die Intensität des	Funktion nicht vorhanden	-

	Signal um den Faktor 1.5 Pfeil nach links vermindert das Signal um die 0.75		
Benutzer kann Modi Parameter ändern	Pfeil nach oben verändert die Sensibilität des Modus um den Faktor 1.5 Pfeil nach unten verändert die Sensibilität des Modus um den Faktor 0.75	Funktion nicht vorhanden	-

## 5.2 Soll-Ist vergleich

Hier wird überprüft, inwieweit die im Pflichtenheft definierten Soll-Kriterien erreicht wurden.

Muss Kriterien	Erfüllt	Grund (falls nicht oder fast erfüllt)
LEDs zeigen korrekte Farben und Helligkeit in der Startanimation an ohne sichtbare Verzögerung	✓	
Sensoren geben Daten der Realität entsprechend aus	~✓	Linker Ultraschallsensor ist anderes Modell als andere, LEDs schalten manchmal aus.
Startanimation beim Einschalten der Matrix läuft direkt über auf Grundmodus Regenbogen	✓	
Ultraschallfernbedienung wechselt zuverlässig zwischen allen Modi	~✓	Fernbedienung wurde durch Keypad ersetzt
Ultraschallsensoren erkennen Objekte von drei verschiedenen Umgebungszonen (Links, Mitte, Rechts)	✓	
Modus zur Erkennung der Lichtverhältnisse nach Lichtgrad bei eingeschaltetem Licht (dunkel) und ausgeschaltetem Licht (hell)	✓	
Modus für Erkennung der Umgebungsgeräusche mit Genauigkeit auf Variation der Geräusche 1-10	✓	
Modus für Erkennung der Temperatur von 10 bis 40 Grad	✗	Temperatursensor gibt falsche Werte aus
Modus für alle Sensoren der alle Inputs erkennbar wiedergibt	✓	

Wunsch Kriterien	Erfüllt	Grund (falls nicht oder fast erfüllt)
Ultraschallsensoreingänge werden für zu einer Animation kombiniert	✓	-
Eigengestaltetes 3D gedrucktes Gehäuse	✗	Keine Priorität, entsprechend fehlte die Zeit
Parameter der Sensordaten können modifiziert per Pfeiltasten oben und unten und rechts der Fernbedienung werden	✗	Keine Priorität, entsprechend fehlte die Zeit
Parameter der Animationen können modifiziert per Pfeiltasten Links Rechts der Fernbedienung werden	✗	Keine Priorität, entsprechend fehlte die Zeit
Musikvisualisierung kann ein- oder ausgeschaltet werden per Fernbedienung	✗	Keine Priorität, entsprechend fehlte die Zeit
Der Farbgrundton kann per Fernbedienung geändert werden	✗	Keine Priorität, entsprechend fehlte die Zeit
Sensorprioritäten können manuell über die Fernbedienung gesetzt werden	✗	Keine Priorität, entsprechend fehlte die Zeit

..



## 6 Anhang

### 6.1 GitHub Link

<https://github.com/valvinc/LEDduino-1.0>

### 6.2 Quellen

ChatGPT – bei weitem nicht alle Chats.

Code

<https://chatgpt.com/share/67606059-fcec-8013-83d9-3cd23bd75409>

<https://chatgpt.com/share/67606068-3d2c-8013-816d-d944d595eb4b>

Stromversorgung

<https://chatgpt.com/share/676060a2-402c-8013-99df-9a8d4c4e3d48>

YouTube - LED Matrix - Connect, Power & Control - Setup Guide

[https://www.youtube.com/watch?v=6XGeM2\\_Zx4&pp=ygUbYXJkdWlubysZWQgbWF0cmI4IHR1dG9yaWFs](https://www.youtube.com/watch?v=6XGeM2_Zx4&pp=ygUbYXJkdWlubysZWQgbWF0cmI4IHR1dG9yaWFs)

## 7 Kurzfazit

Im Endeffekt bin ich auf das Projekt stolz. Während der Umsetzung ist mir aufgefallen, dass ich viel vorgenommen habe. Es ist auch nur ein Teil der Idee Projekts fertig.

Der Versuch ein Programm mit viel Inhalt hauptsächlich durch GPT zu schreiben ist sicherlich gelungen. Jedoch darf wie bereits erwähnt nie die Menschliche Verantwortlichkeit bei dem ganzen unterschätzt werden. Ein Prompt „Mach mir ein Arduino Projekt mit LEDs“ war es sicher nicht.

Ein grossteil der Zeit wurde für die Realisierung selbst als die Dokumentation aufgewendet. Auch gab es viel Zeitverlust durch die Stromversorgung der Matrix. Entsprechend war für den Feinschliff des Dokuments sehr wenig Zeit. Hier könnte sehr viel noch einheitlicher sein. Auch sind einzelne Variablen noch auf ausserhalb des config.h, das Stört, Gerne hätte ich das Flussdiagramm auch noch besser dargestellt. Es gab Teilweise sehr viel Chaos, auch in der Umsetzung, das kann ich nächstes Mal sicher besser machen.

Aus dem Grund des Fokus an anderen Orten entschuldige ich mich beim Leser für allfällige Rechtschreibfehler und Stilkatastrophen. Ich habe mich auf die alles in allem 663 Zeilen Code konzentriert.