



LED-WÜRFEL

Beschreibung

4x4x4 Led-Würfel mit Soundsensor, Infrarot-Steuerung und vorgefertigten Mustern auf dem Arduino Uno R3

Modul: APM (Bachelor, Wahlfach)
Betreuer: Dr.-Ing. Hubert Zitt

Selbstständigkeitserklärung

Hiermit versichern wir, dass wir das gesamte Projekt selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Die Stellen unserer Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken und Quellen, einschließlich Quellen aus dem Internet, entnommen sind, haben wir in jedem Fall unter Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.

Ort, Datum Unterschrift – Mirco Roth - 875178

Ort, Datum Unterschrift – Donald Pfaffmann – 875180

Inhaltsverzeichnis

1. Verwendete Komponenten	4
1.1 Software	4
1.2 Hardware	4
2. Bauanleitung	5
2.1 LED-Würfel	5
2.1.1 LED-Vorbereitung	5
2.1.2 LED-Gatter	6
2.1.3 Konstruktion des LED-Würfels ohne Sockel	6
2.2 Sockel des Würfels	8
2.2.1 Bau des Sockels aus einer Holzkiste	8
2.2.2 Zusammenbau der Holzkiste mit dem LED-Würfel	8
2.3 Sensoren	10
2.3.1 Soundsensor	10
2.3.2 Infrarotsensor	11
2.4 Stromversorgung	12
2.4.2 Integration der Powerbank	12
2.5 Schaltplan	13
3. Code	14
4. Quellenverzeichnis	18

Einführung

Dieses Dokument beschreibt den Aufbau eines 4x4x4 LED-Würfels, welcher auf verschiedene Muster programmierbar ist und damit ein individuell einstellbares Lichterspiel darstellt. Hierfür wurde ein Arduino Uno R3 als Grundlage für die Programmierung und Schaltung verwendet.

Mit Infrarot-Sensor ausgestattet, können die einzelnen Muster, aber auch der eingebaute Soundsensor, über eine Fernbedienung gesteuert werden. Mithilfe des Soundsensors kann der Würfel somit zu wahrgenommenen Geräuschen - zum Beispiel Musik - mit vordefinierten Mustern reagieren, wodurch er auch auf einer Party Verwendung finden könnte. Die integrierte Powerbank ist hierfür ebenfalls nützlich. So kann der Würfel über USB-Micro geladen werden und im Anschluss kabellos an einer geeigneten Stelle platziert werden.

Eine große Herausforderung liegt im Zusammenbau des Würfels, da dies stets eine ruhige Hand und Fingerspitzengefühl beim Löten der LEDs erfordert. Ebenso kann das Programmieren von geeigneten Mustern sehr aufwendig und schnell unübersichtlich werden. Die Schaltung hingegen ist relativ einfach. Zunächst folgt eine Auflistung und Beschreibung der benötigten Komponenten zum Nachbauen des Würfels mit einer konkreten Bauanleitung.

1. Verwendete Komponenten

1.1 Software

- Die Software wurde in der Arduino IDE realisiert

1.2 Hardware

- Arduino Uno R3, ATmega328
- 64 x LED-Diode, 8 mm x 5 mm, blau, 2-Pin DC 2,5–3 V
- 4 x 100 Ohm Widerstand
- 6m Kupferdraht, 0.8 mm, verzinkt
- Holzkiste, 180 mm x 190 mm
- Lochraster-Platine, 100 mm x 150 mm
- 74HC595, Shift-Register
- Steckplatine (Breadboard), 170 Kontakte
- ca. 25 x Jumper Kabel, unterschiedliche Größen
- 4 x Holzstücke, 20 mm x 10 mm
- Karton, 170 mm x 180 mm
- USB A / USB B 2.0 Kabel
- Drucktaster
- Powerbank, 2600mAh
- HX1838 VS1838 NEC Infrarot Wireless Remote Control Sensor
- Digitaler Sound Schall Sensor, 3,3V – 5V
- Lötzinn, 15g
- Spraydose, schwarz

2. Bauanleitung

2.1 LED-Würfel

2.1.1 LED-Vorbereitung

Der grundlegende Aufbau einer LED sieht folgendermaßen aus:

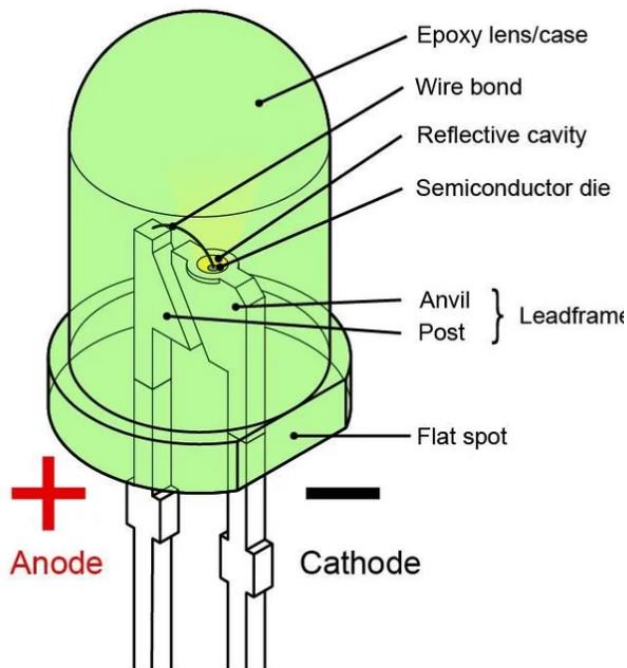


Abbildung 1: Aufbau einer LED

Abb.1 zeigt den Aufbau einer typischen LED. Die Kathoden (kürzeres Drahtende) müssen um einen Nagel schlaufenförmig gebogen (s. Abb. 2) und die Anoden knapp abgeknipst werden (s. Abb. 3). Diesen Vorgang wiederholt man für alle LEDs. Besonders eignen sich im Übrigen diffuse LEDs, da diese eine gleichmäßige Lichtstreuung besitzen und dadurch dem Beobachter heller erscheinen. Zudem wird verhindert, dass benachbarte LEDs angestrahlt werden und somit den Eindruck erwecken ebenfalls eingeschaltet zu sein, wodurch die Konturen von Mustern schlechter wahrgenommen werden.



Abbildung 2: Kathode schlaufenförmig biegen



Abbildung 3: Anode abknippen

2.1.2 LED-Gatter

Um die LED-Gatter zu bauen, wurden mit Hilfe einer Schablone Einsetzlöcher mit jeweils 3cm Abstand zwischen den LEDs auf einer Holzplatte gebohrt (s. Abb. 4-5). Durch das Fixieren der LEDs in den Löchern wird es einfacher, vier identische Gatter zu löten. Ein uniformes Aussehen des Würfels ist später von großer Bedeutung, weshalb man hier auf jeden Fall präzise arbeiten sollte. Insgesamt misst der Würfel 10x10x10cm.

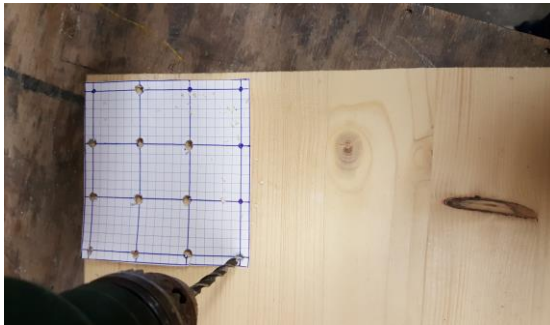


Abbildung 4: Schablone bohren

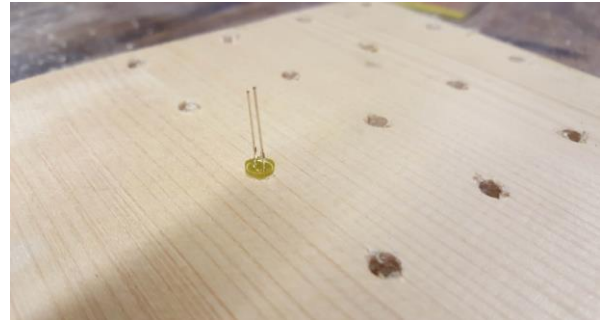


Abbildung 5: fertige Schablone mit LED

Nachdem die vier Gatter gelötet wurden (s. Abb. 6) kann es an den Zusammenbau des Würfels gehen.

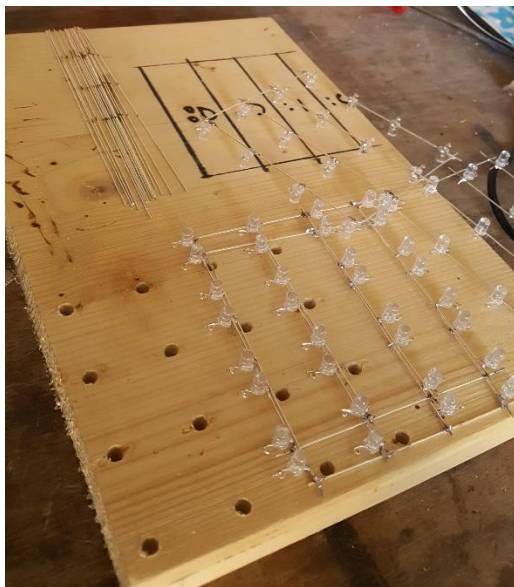


Abbildung 6: fertige LED-Gatter

2.1.3 Konstruktion des LED-Würfels ohne Sockel

Nun müssen die 16 Drahtfüße des Würfels durch die Schlaufen der Kathoden der LEDs gefädelt und gelötet werden. Dabei ist auf den 3cm Abstand zwischen den Ebenen zu achten, als auch darauf, dass die Ebenen gerade liegen (s. Abb. 7)

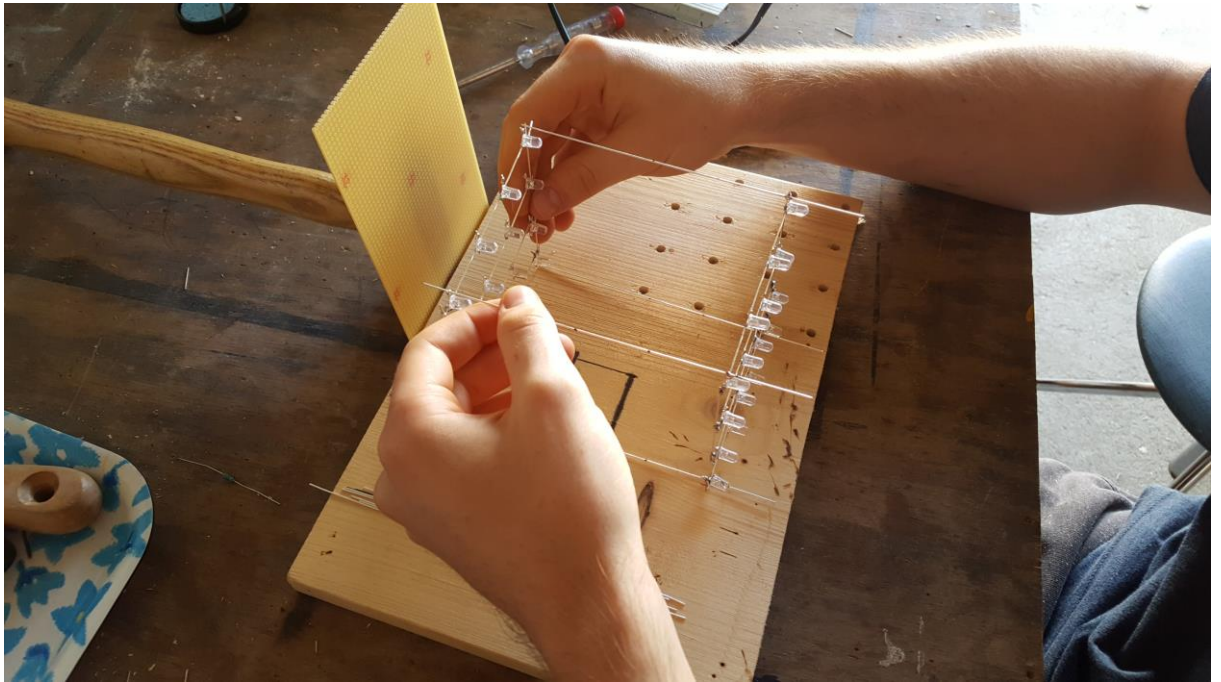


Abbildung 7: LED Gatter werden mit den Drahtfüßen verbunden:

Um den Draht für die Füße des Würfels gerade zu biegen, kann man eine Bohrmaschine verwenden. Dafür wird der Draht zwischen der Bohrmaschine und einer Zange fixiert und langsam gedreht (siehe Abb. 8).



Abbildung 8: Draht mit Bohrmaschine und Zange gerade biegen

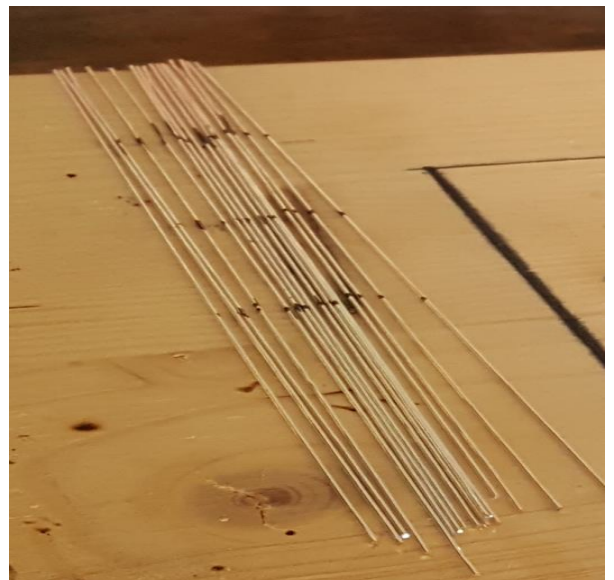


Abbildung 9: Drähte im 3cm Abstand für die Ebenen markieren

Die Drahtfüße werden auf 12cm Länge gekürzt und jeweils alle 3cm für die Positionierung der Ebenen markiert (s. Abb.9).

2.2 Sockel des Würfels

2.2.1 Bau des Sockels aus einer Holzkiste

Für den Sockel wurde eine schlichte Holzkiste schwarz lackiert. Anschließend wurden die 1mm Bohrlöcher für die Füße des Würfels markiert, indem der Würfel zuvor durch die Lochrasterplatine gesteckt und die entsprechenden Löcher markiert wurden (s. Abb. 10). Ebenfalls wurden 4 nebeneinander liegende Löcher an eine der 4 Seiten des Würfels markiert, welche später für die Versorgung der Ebenen dienen. Danach wurden die Löcher mit einem Standbohrer gebohrt (s. Abb. 11).



Abbildung 10: Holzkiste mit Markierungen für die späteren Löcher



Abbildung 11: Standbohrer

2.2.2 Zusammenbau der Holzkiste mit dem LED-Würfel

Damit das Durchführen der Füße des LED-Würfels durch die Platine einfacher ist, werden an den Markierungen die Löcher größer gebohrt (s. Abb. 12). Im Anschluss wird die Lochrasterplatine von der Innenseite an den Deckel der Holzkiste geklebt und die Füße des Würfels durchgesteckt (s. Abb. 13).

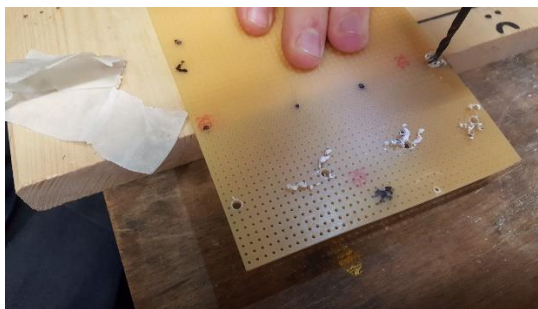


Abbildung 12: Markierungen auf der Lochrasterplatine werden gebohrt



Abbildung 13: Innenseite der Holzkiste mit herausragenden Drahtfüße

Anschließend wird die überstehende Länge der Füße abgeknipst und die selbigen mit Jumper-Wires verlötet (s. Abb. 14). Diese wiederum werden in die digitalen PINs des Arduinos gesteckt.

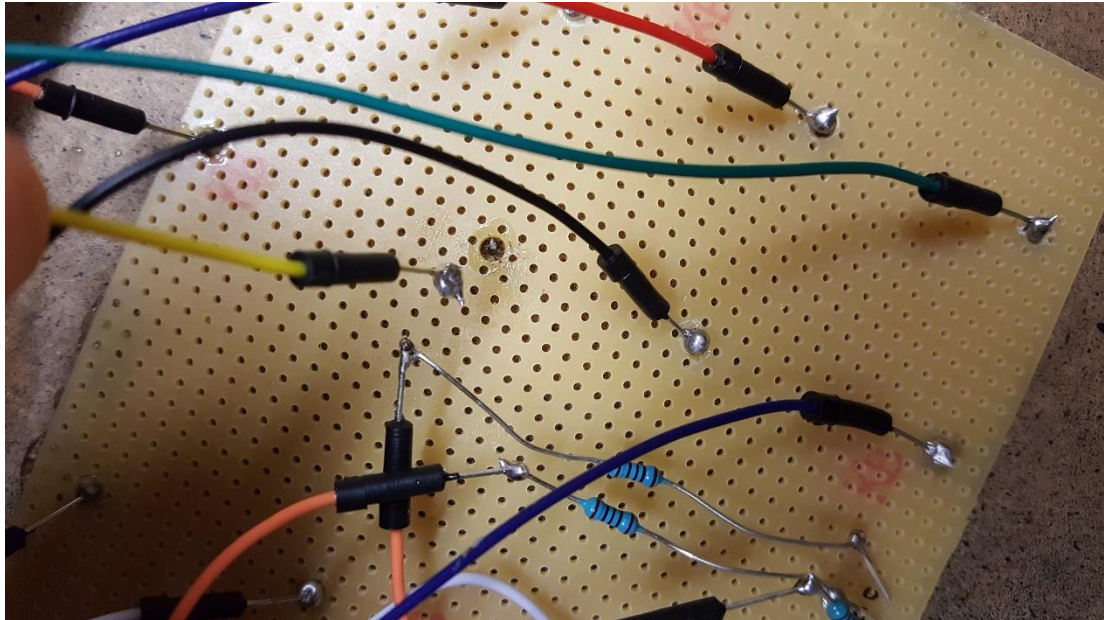


Abbildung 14: Löten der Jumper Wires an die Drahtfüße

Nun werden die Ebenen des Würfels jeweils durch einen Analogen PIN des Arduinos mit 5V versorgt, wobei jeweils ein 100 Ohm Widerstand vorgeschaltet ist. Hierbei führt jeweils noch einmal ein gesondertes Drahtstück an die Ebenen, um diese als Ganzes mit 5V zu versorgen (s. Abb. 15).

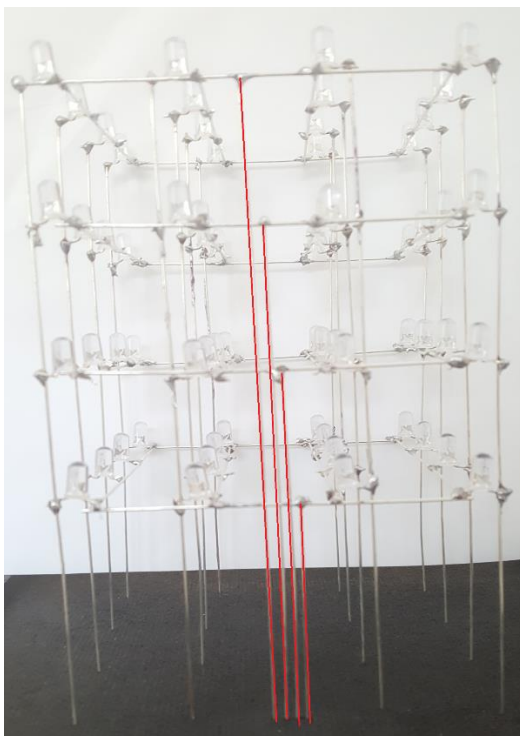


Abbildung 15: 5V Versorgung für die einzelnen Ebenen

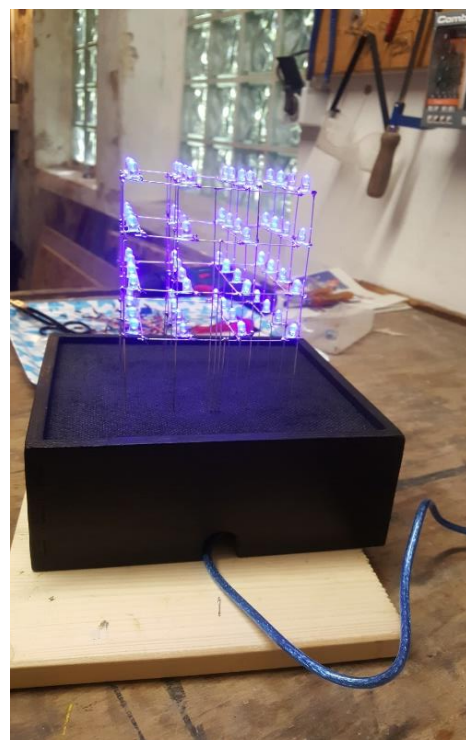


Abbildung 16: Funktionierende LED-Würfel

Damit ist der LED-Würfel im Grunde schon bereit für den Betrieb (s. Abb. 16). Damit die Technik im Inneren des Sockels etwas geschützt liegt (die Kabel nicht herausragen) und man den Würfel besser heben kann, wurde ein Kartonboden eingepasst. Dabei verhindern Stopper aus Holz, dass der selbige zu weit eingedrückt werden kann (siehe Abb. 17-18).

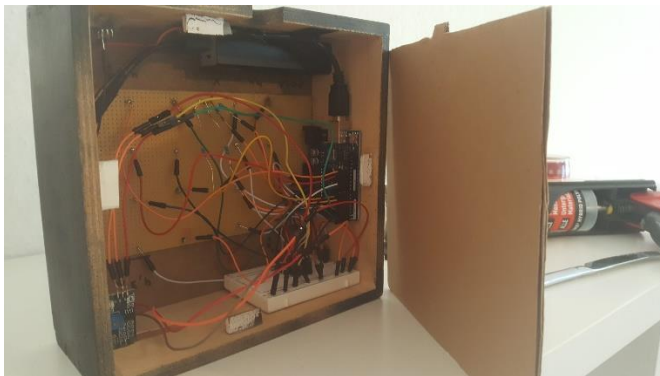


Abbildung 17: Holzstopper



Abbildung 18: Boden aus Karton

2.3 Sensoren

Da wir mehr als nur eine schicke Tischlampe haben wollten, entschieden wir uns dazu, noch einige Spielereien einzubauen. Darunter fiel ein Soundsensor, welcher es dem Würfel erlauben soll, zum Rhythmus eines Liedes zu leuchten. Des Weiteren bauten wir – der einfacheren Bedienbarkeit wegen – noch einen Infrarotsensor ein. Damit ist es möglich zwischen verschiedenen vordefinierten Mustern und dem klangreaktiven Modus zu wechseln.

2.3.1 Soundsensor

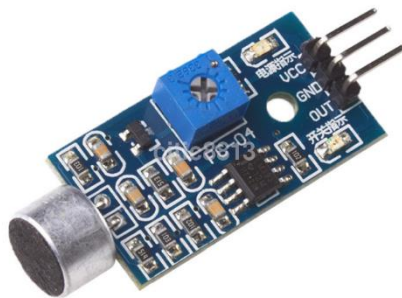


Abbildung 19: Verwendeter Soundsensor



Abbildung 20: Soundsensor eingepasst in die Wand des Sockels

Die Belegung des Soundsensors (s. Abb. 19) erfolgt recht simpel. Hierbei wird der unterste PIN (OUT) an den digitalen PIN 4 des Arduinos angeschlossen. Über diesen können wir später auslesen, ob der Sensor ein Geräusch erkannt hat. Der mittlere (GND) und obere (VCC) PIN werden entsprechend an 0V bzw. 5V angeschlossen. Schließlich wird der Sensor in eine Wand des Sockels eingepasst (s. Abb. 20).

2.3.2 Infrarotsensor



Abbildung 21: HX1838 VS1838 NEC Infrarot Wireless Remote Control Sensor

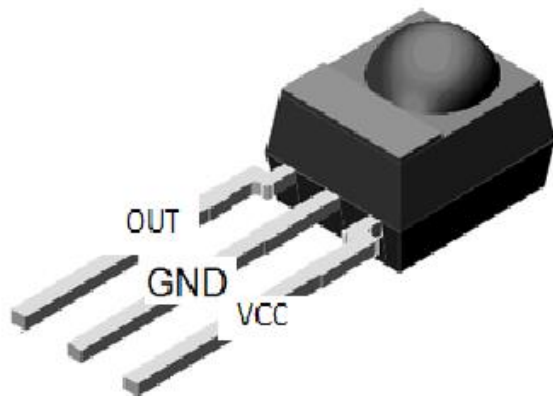


Abbildung 22: IR-Receiver Diode

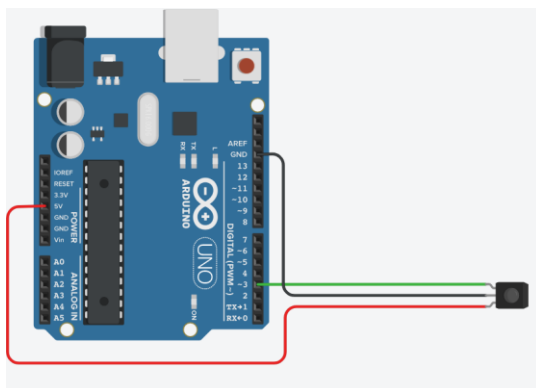


Abbildung 22: Schaltplan der IR-receiver Diode

Um den Würfel zu bedienen, haben wir eine Infrarot-Fernbedienung angedacht (s. Abb. 21). Hierfür muss eine Receiver-Diode (s. Abb. 22) am Arduino angeschlossen werden. Dazu haben wir den linken PIN (OUT) an den digitalen PIN 3 des Arduinos angeschlossen, da über diesen in der Arduino IDE ein Interrupt möglich ist. Über diesen können wir später in der Software die empfangenen Daten auslesen und verarbeiten, d.h. in

unserem Fall erkennen, welcher Knopf gedrückt wurde und das korrespondierende Muster im Würfel abspielen – sogar während ein anderes Muster gerade abgespielt wird. Der zweite (GND) und dritte (VCC) PIN, sind entsprechend an 0V bzw. 5V angeschlossen (s. Abb. 22).

2.4 Stromversorgung

2.4.1 Motivation

Da wir uns bewusst für einen mobilen LED-Würfel entschieden haben, wollten wir auch nicht an das Vorhandensein einer Steckdose in der Nähe gebunden sein. Es soll möglich sein, den LED-Würfel im Raum an einer beliebigen Stelle zu platzieren, sodass auch der Soundsensor optimal funktioniert. Auf eine Batterie wurde bei der Umsetzung verzichtet, da diese meist nicht lange hält und das Austauschen umständlich ist. Deshalb fiel die Wahl auf das Verbauen einer Powerbank, welche beide Probleme löst.

2.4.2 Integration der Powerbank

Die Powerbank ist an den Arduino über ein USB A / USB B 2.0 Kabel angeschlossen. Damit die Powerbank nicht automatisch anfängt, den Arduino mit Strom zu versorgen, wurde das Kabel durchtrennt und die Schaltung durch einen Drucktaster ergänzt (s. Abb. 23). Dies ermöglicht es den Würfel manuell ein – bzw. auszuschalten. Die USB - Micro B-Übchse der Powerbank wurde mit einem Verlängerungskabel an die Außenseite des Würfels gelegt, damit man sie bequem über ein Handyladekabel aufladen kann. Somit kann der LED-Würfel sowohl an einer Steckdose, als auch autark an einer beliebigen Stelle betrieben werden.

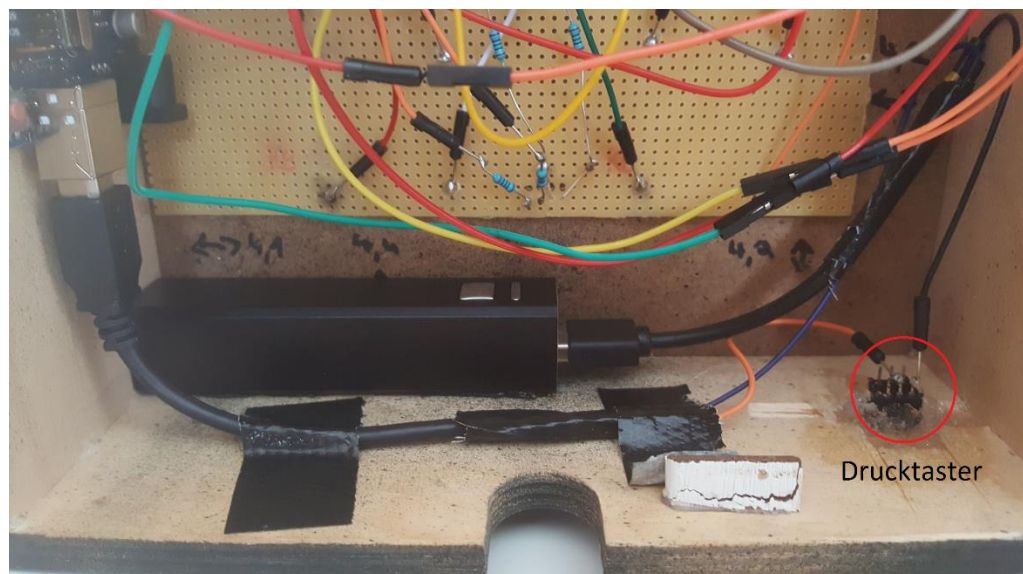


Abbildung 23: Powerbank mit Drucktaster

2.5 Schaltplan

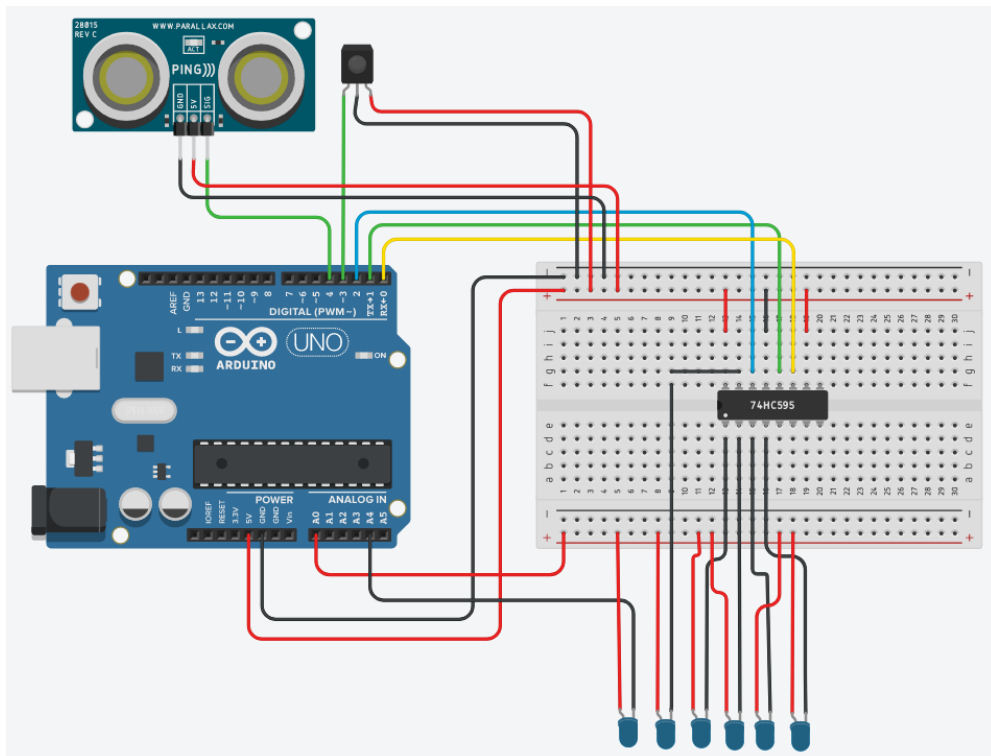


Abbildung 24: Ausschnitt des Schaltplans des Würfels mit Sensoren

Abb. 24 zeigt die Anbindung der Sensoren und den konzeptionellen Anschluss der LEDs. Dieser ist jedoch nicht repräsentativ für die tatsächliche Schaltung, da die 3 dimensionale Schaltung des Würfels schwer darzustellen ist.



Abbildung 25: Struktur des Würfels mit Ebenen (Zeilen) & Drahtfüßen (Spalten)

Es müssen 64 LEDs angeschlossen werden. Diese werden aufgeteilt in 4 x 16 LEDs, d.h. 16 LEDs pro Ebene. Darüber hinaus ist jeder Drahtfuß (Spalte) an die Kathoden von jeweils 4 LEDs gelötet. Dadurch kann man nicht an eine Kathode einer einzelnen LED GND anlegen, sondern immer nur gleichzeitig für 4 LEDs. Durch das Anlegen von 5V auf einer gesamten Ebene kann hierdurch ein Schnittpunkt erzeugt werden, sodass eine LED den Stromkreis schließt. Somit kann auch eine einzelne LED zum Leuchten gebracht werden. Abb. 25 verdeutlicht diese Erklärung bildlich.

Der Vorteil liegt darin, dass nicht jede einzelne LED an den Arduino angeschlossen werden muss, sondern die Anzahl von Steckplätzen von 64 auf 16 gekürzt werden kann. Der Nachteil jedoch ist, dass sich manche Muster nicht realisieren lassen, da man gleichzeitig immer an 4 LEDs GND anlegt.

Abb. 24 zeigt wie der Infrarotsensor und der Geräuschsensor angeschlossen wurden. Die angeschlossenen LEDs hingegen sind eine rein schematische Darstellung. Der analoge PIN 0 versorgt hier die +-Zeile des Breadboards, welche wiederum alle LEDs versorgt. Sie stellt damit in ihrer Funktion eine Ebene dar. GND wird den LEDs hingegen durch einzelne digitale PINs (aber auch A4/5) des Arduinos bereitgestellt.

Da insgesamt schon alle analoge und digitale PINs alleine für den Betrieb des LED-Würfels benötigt werden (4 für die Ebenen + 16 für die Spalten), können die Sensoren nicht ohne weiteres angeschlossen werden. Um zusätzliche Steckplätze zu erhalten, wurde ein Shift-Register eingebaut. Dieses besitzt 8 PINs, wobei 3 an den Arduino angeschlossen werden müssen, wodurch man effektiv 5 Steckplätze gewinnt. Indem nun 5 Spalten des LED-Würfels über das Register gesteuert werden, können die Sensoren angeschlossen werden.

3. Code

Zunächst werden die Ebenen und Spalten definiert. Damit können nun die LEDs angesprochen werden, wobei später erläutert wird, wie man dies bewerkstelligt. Für den Infrarotsensor wurde die Bibliothek aus GitHub (<https://github.com/z3t0/Arduino-IRremote>) eingebunden.

```
#include <IRremoteInt.h>
#include <ir_Lego_PF_BitStreamEncoder.h>
#include <IRremote.h>
#include <boarddefs.h>

//initializing and declaring led layers
int layer[4]={A0,A1,A2,A3};
//initializing and declaring led rows (except the ones that are controlled by the shift register)
int column[11]={A5,A4,5,6,7,8,9,10,11,12,13};
```

Der LED-Würfel besitzt vorgefertigte Muster, welche mithilfe der Fernbedienung abgespielt werden können. Ein Muster wiederholt sich solange, bis eine andere Taste gedrückt wird. Hierfür wurden die unterschiedlichen Zustände in dem ENUM „State“ festgehalten. Der Startzustand ist „S_ON“ und lässt jede LED angehen. Da der

Infrarotsensor hexadezimale Werte an den Receiver übermittelt, wurden Präprozessormakros für die einzelnen Tasten auf der Fernbedienung definiert.

```
//all states the cube can be in
enum State{
    S_SOUND, S_SPIRAL, S_PROPELLER, S_SNAKE, S_FIREWORK, S_LAYERBOUNCE,
    S_WAVE, S_RRAIN, S_RFLICKER, S_PULSING, S_AUTO, S_OFF, S_ON
};

//state of the LED cube; default is turnEverythingOn()
volatile State state = S_ON;

//the infrared remote control (sender) sends hexadecimal values to the
//receiver in order to be able to distinguish between them we have mapped
//them to their respective function
#define OFF      0x00FFA25D      // CH
#define ON       0x00FFC23D      // >||
#define SOUND    0x00FF906F      //EQ
#define AUTO     0x00FF6897      // 0
#define SPIRAL   0x00FF30CF      // 1
#define PROPELLER 0x00FF18E7      // 2
#define SNAKE    0x00FF7A85      // 3
#define FIREWORK 0x00FF10EF      // 4
#define LAYERBOUNCE 0x00FF38C7    // 5
#define WAVE     0x00FF5AA5      // 6
#define RRAIN    0x00FF42BD      // 7
#define RFLICKER 0x00FF4AB5      // 8
#define PULSING  0x00FF52AD      // 9
```

Mithilfe der Methode „attachInterrupt“ kann beim Empfangen eines Signals (auch während gerade ein Muster gespielt wird) dieses direkt durch Aufruf der Methode „isr“ verarbeitet werden.

```
//listen for interrupts on digital pin 3 (INT1)
attachInterrupt(1, isr, RISING);
//start the receiver
irrecv.enableIRIn();
```

Die Methode „isr“ greift sich die empfangenen values ab und wechselt entsprechend in einer der definierten Zustände.

```

//the interrupt service routine is responsible for determining
// the next state depending on the button bressed on the remote control
void isr(){

    while (irrecv.decode(&results)){
        if      (results.value == SOUND) state = S_SOUND;           //button "EQ" pressed
        else if (results.value == SPIRAL) state = S_SPIRAL;        //1
        else if (results.value == PROPELLER) state = S_PROPELLER;  //2
        else if (results.value == SNAKE) state = S_SNAKE;          //3
        else if (results.value == FIREWORK) state = S_FIREWORK;    //4
        else if (results.value == LAYERBOUNCE) state = S_LAYERBOUNCE; //5
        else if (results.value == WAVE) state = S_WAVE;            //6
        else if (results.value == RRAIN) state = S_RRAIN;          //7
        else if (results.value == RFLICKER) state = S_RFLICKER;    //8
        else if (results.value == PULSING) state = S_PULSING;      //9
        else if (results.value == AUTO) state = S_AUTO;            //0
        else if (results.value == OFF) state = S_OFF;              //CH-
        else if (results.value == ON) state = S_ON;                //>||

        //Serial.println(results.value);
        //ready to receive the next value
        irrecv.resume();
    }
}

```

In der „loop“-Methode kann dadurch über einen switch anhand des momentan ausgewählten Zustandes entschieden werden, welche Methode aufgerufen wird.

```

// ...
case S_PROPELLER: propeller();
                  break;

case S_SNAKE: snake();
              break;

case S_FIREWORK: firework();
                  break;

case S_LAYERBOUNCE: horizontalLayerBounce();
                    verticalLayerBounce();
                    break;

case S_WAVE: wave();
              break;
// ...

```

Am Beispiel der Methode „turnEverythingOn()“ - welche für den Zustand „S_ON“ aber auch für viele andere Muster benötigt wird - wird nun das Steuern der LEDs gezeigt.

```
//turn all LEDs on
void turnEverythingOn(){

    //turn all layers on by applying 5V
    for(int i = 0; i<4; i++){
        digitalWrite(layer[i], 1);
    }
    //turn all columns on which are controlled by the arduino by applying ground/0V
    for(int i = 0; i<11; i++){
        digitalWrite(column[i], 0);
    }
    //turn all columns on (apply ground) which are controlled by the shift register
    setShiftRegister(0);
}
```

Um alle LEDs zum Leuchten zu bringen, müssen alle 4 Ebenen auf 5V gesetzt werden.

```
for(int i = 0; i<4; i++){
    digitalWrite(layer[i], 1);
}
```

Anschließend müssen noch alle Drahtfüße auf 0V (GND) gesetzt werden - diese sind standardmäßig auf 5V gesetzt, damit keine LED leuchtet. Dafür werden zunächst die 11, an den Arduino angeschlossenen Spalten, auf 0V (GND) gesetzt.

```
for(int i = 0; i<11; i++){
    digitalWrite(column[i], 0);
}
```

Die 5, an das Shift-Register angeschlossenen Spalten, müssen gesondert über die vorgesehene Methode „setShiftRegister(unsigned char uc)“ angesprochen werden.

```
setShiftRegister(0);
```

Dabei werden alle Output-PINs des Registers auf GND gesetzt, wodurch an den 5 Spalten ebenfalls 0V anliegen. Erreicht wird dies durch folgende Anweisungen:

```
//digital pins 1-5 cannot be accessed directly since they are connected to the shift register
//to access them, we will have to address them seperately through said register
void setShiftRegister(unsigned char uc){

    //make the latchPin low so that the LEDs don't change while sending in bits
    digitalWrite(latchPin, LOW);
    //shift out the bits
    shiftOut(dataPin, clockPin, MSBFIRST, uc);
    //make the latchPin high so the send byte is displayed by the outputs of the register
    digitalWrite(latchPin, HIGH);
}
```

Bei dem Register handelt es sich um ein 8-bit serial-in, serial or parallel-out Shiftregister mit Output-PINs. D.h., da ein Byte seriell, also immer bitweise eingespeist wird, müssen wir die Ausgabe des Registers verzögern bis alle Bits angekommen sind. Dies geschieht durch die Anweisung „digitalWrite(latchPin, LOW)“. Durch die Anweisung shiftOut(dataPin, clockPin, MSBFirst, uc) wird das zu

übertragende Byte, bitweise durch einen Takt gesteuert, an das Register gesendet. Sind alle Bits angekommen, erfolgt die Ausgabe.

Da die Muster zu umfangreich sind, um sie hier aufzuführen, verweisen wir an dieser Stelle auf den gesamten Code auf der CD.

4. Quellenverzeichnis

1: 4x4x4 LED-CUBE (Arduino Uno),

<https://www.instructables.com/id/4x4x4-LED-Cube-Arduino-Uno/>,

Stand: 2.10.2018 – 20:02 Uhr.

2: Serial to Parallel Shifting-Out with a 74HC595,

<https://www.arduino.cc/en/Tutorial/ShiftOut>,

Stand: 2.10.2018 – 20:03 Uhr.

3: Sound Reactive LED strip,

<https://www.instructables.com/id/Sound-Reactive-LED-strip/>,

Stand: 2.10.2018 – 20:07 Uhr.

4: HOW TO SET UP AN IR REMOTE AND RECEIVER ON AN ARDUINO,

<http://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/>,

Stand: 2.10.2018 – 20:09 Uhr.