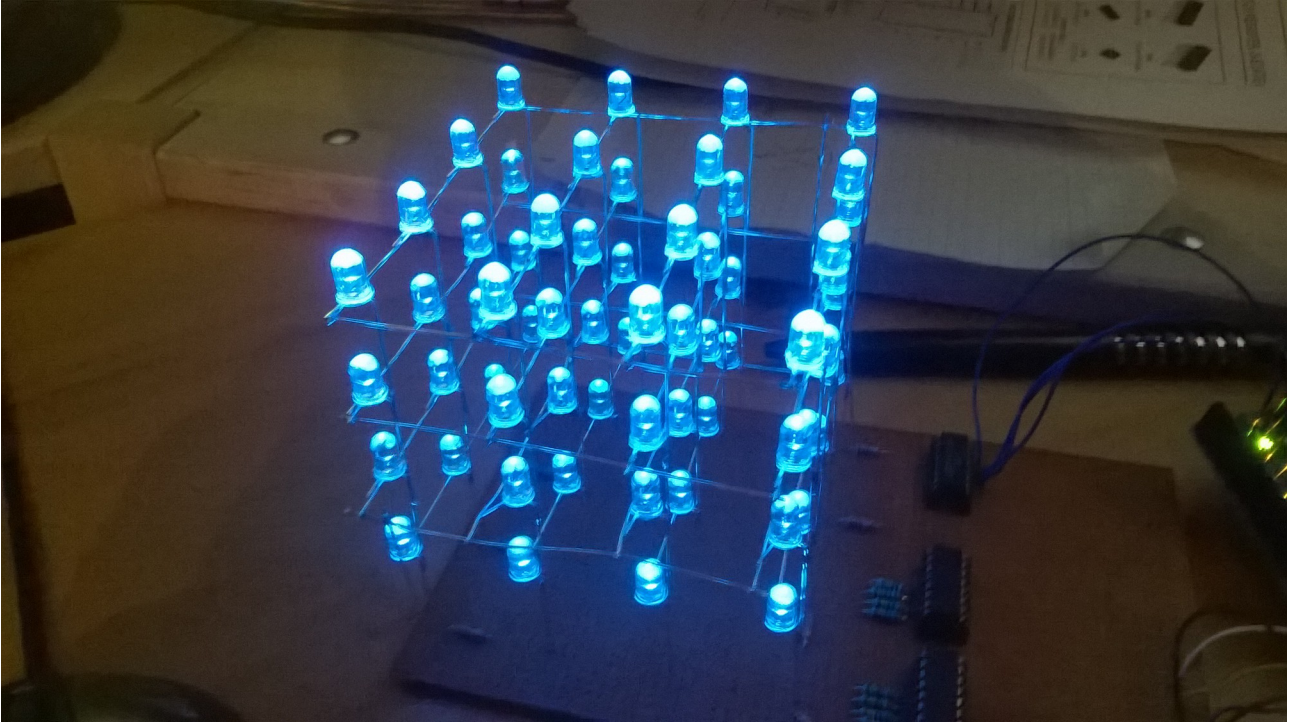


# LED CUBE 4x4x4



von Josua Benz und Patrick Koss

im Fach Computer Technik

betreut von Herr Mezger

## **LED Cube**

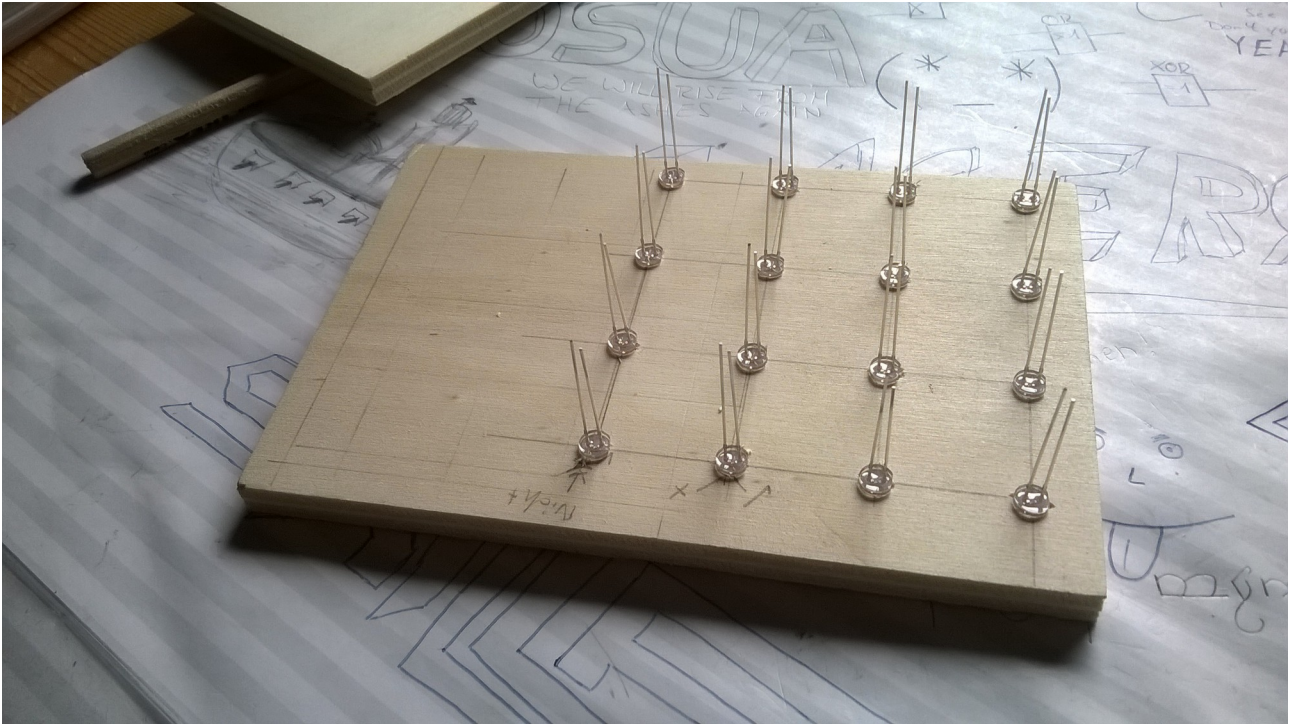
Aufgrund zahlreicher faszinierender Videos von LED Cubes haben wir uns dazu entschlossen, selbst auch einen Cube im Rahmen des Unterrichts im Fach CT zu bauen. Eine Großzahl der benötigten LEDs hatten wir schon privat aus anderen Projekten. Wir entschieden uns, einen 4x4x4 großen Cube zu bauen, da dieser einfach anzusteuern ist und trotzdem sehr interessante und kreative Muster möglich sind, was bei der kleineren 3x3x3 Version nur wesentlich eingeschränkter möglich ist. Bei einem Cube der nächsten Größenordnung (125 LEDs) wären es wesentlich höhere Materialkosten und auch die Ansteuerung wäre weniger Effizient da pro Ebene 25 LEDs anzusteuern gewesen wären.

## **Funktionsweise**

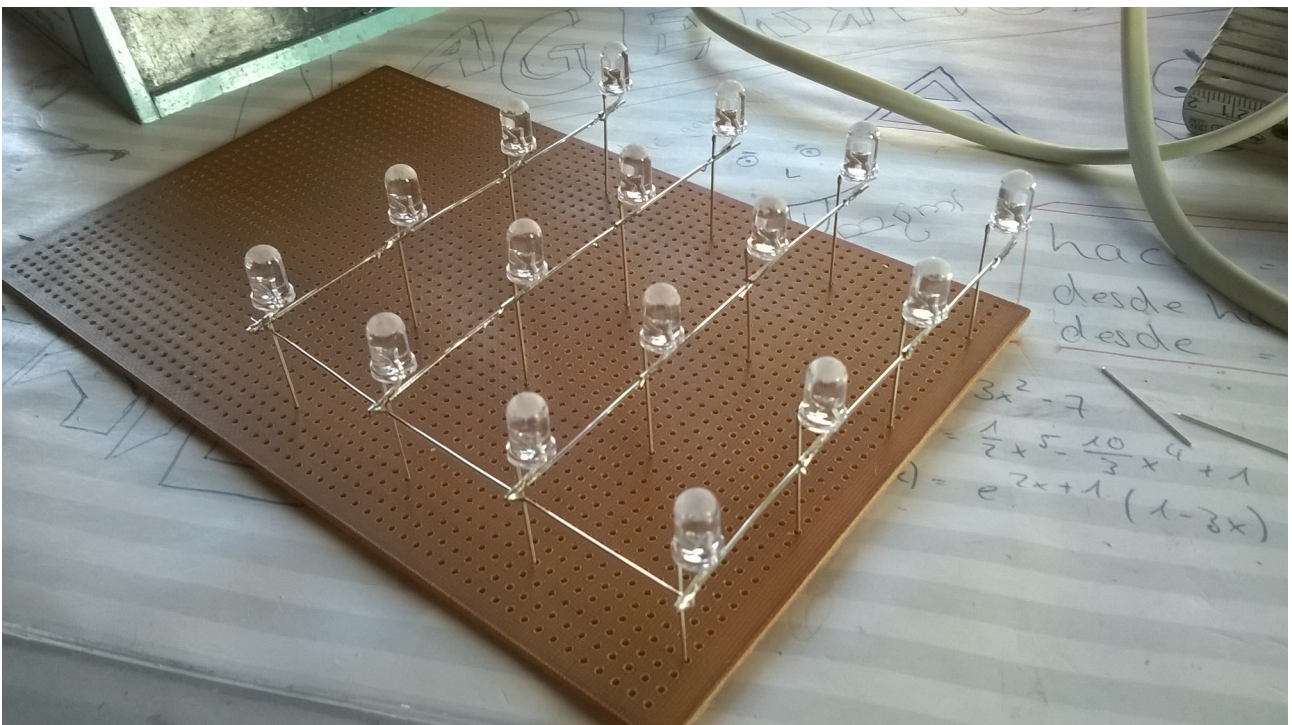
Wie schon beschrieben benutzen wir zur Ansteuerung des Cubes Schieberegister. In diesen wird ein Muster für alle LEDs einer Ebene angelegt. Wenn nun das Strobe-Signal kommt, werden alle Bits vom Schieberegister auf den Output übertragen. Dann liegen an den LEDs 1er oder 0er an. Um nun nur die LEDs einer Ebene anzusteuern, benutzen wir den Chip ULN2003. Dieser kann einen Port gegen Ground schalten. Wir haben also Anoden vertikal mit den darunter liegenden Anoden verbunden und alle Kathoden einer Ebene horizontal verbunden. Nun müssen die Anoden an die 16 Ports der Schieberegister angeschlossen und die 4 Ebenen an die Ports des ULN2003-Chips angeschlossen werden. Durch schnelles wechseln der Ebene und der Muster der LEDs entsteht der Eindruck, dass alle LEDs gleichzeitig leuchten, was aber nicht richtig ist. Dies liegt daran, dass unsere Augen das schnelle wechseln der Ebenen nicht wahrnehmen können.

## Aufbau

Bevor wir anfangen, die LEDs zusammenzulöten, bauten wir eine Schablone sodass alle LEDs im gleichen Abstand zueinander liegen. Bei den LEDs handelt es sich um klare, blaue LEDs mit 1500mcd die wir zuvor mit Schmirgelpapier an der Oberfläche anrauhten damit sich das Licht mehr streut.



Anschließend löteten wir bereits beschrieben die Kathoden alle mit Silberdraht zusammen und steckten die Anoden in die Löcher der Lochrasterplatine um diese dort zu verlöten.





Nachdem dies getan war, Plazierten wir die Chips auf der Platine. Wir verwendeten 2 74HC4049 Chips als Schieberegister, die wir in Reihe hintereinander Schalteten um 16 Ports für die 16 LEDs in jeder Ebene zu besitzen. Von diesem Chip aus gehen nur 3 Leitungen zum Arduino, eine Leitung für die Daten, eine für die Clock und eine „Strobe“-Leitung die die Daten in den Registern an die Ports ausbigt. Der andere Chip den wir brauchen ist der ULN2003 dessen Funktion wir schon weiter oben erläuterten. Anschließend verbauten wir noch die Widerstände und löteten dann die Drähte auf der Unterseite der Platine an. Auch 2 Kondensatoren wurden angebracht um die Spannungsschwankungen an den Schieberegistern aufzufangen. Nach einigen Tests mit der Ersten Ebene stapelten wir danach die nächsten Ebenen mit einem Abstandsholz aufeinander und befestigten die Anoden an der Oberseite der Anoden der Ebene darunter. Die Kathoden wurden mit einem kleinen Draht nach unten zur Platine geleitet von wo aus sie dann an den ULN2003 angeschlossen wurden.

### **Programmierung**

Der Programmcode besteht im großen und ganzen aus 3 Teilen. In ersten Teil initialisieren wir einige Variablen für die Ausgänge. Wir benötigen 3 für das Schieberegister und 4 für den ULN2003 um die 4 Ebenen anzusprechen. Außerdem sind einige Arrays vom Datentyp „byte“ gespeichert, die für die Muster zuständig sind. Immer 2 Bytes sind für eine Ebene zuständig, 8 Bytes reichen uns also um ein Muster zu speichern. Für Animationen brauchen wir als für jeden Frame 8 Byte Datenspeicher.

Der zweite Teil ist auch der größte. Es ist die loop()-Methode in der alle „Events“, wie wir sie genannt haben gespeichert sind. Diese sind alle separat und individuell gespeichert, um ein schnelles hinzufügen und ändern der Muster möglich zu machen. Jedes Event besteht aus mehreren for()-Schleifen, die ganz schnell das Muster für die Ebene anlegen, dieses kurz anzeigen und anschließend auf die nächste Ebene wechselt.

Der dritte Teil ist die shiftOut()-Methode, mit der wir die Bytemuster für die Ebenen ausgeben. Diese Methode ist das Herzstück des Programms. Mit der Variablen pinState wird am Ende der Methode das Datensignal an das Schieberegister angelegt. Zunächst wird die Datenleitung und der Takt für das Schieberegister auf Null gesetzt um eventuelle Fehler zu vermeiden. Die for()-Schleife läuft insgesamt 8 mal durch, da ein Byte 8 Bit ist und wir jedes Bit einzeln in das Schieberegister einschieben müssen. Deswegen wird mit einer if Verzweigung das jeweilige Bit des Byte ausmaskiert. Mit  $1 \ll i$  wird auf die Stelle i des Bytes gezeigt, ist diese 1 wird pinState auf 1 gesetzt, andernfalls auf 0. Danach wird das Bit i an das Schieberegister als Datenbit angelegt. Wenn nun der Takt kommt wird das Bit im Schieberegister an einem Ausgang übernommen.

```

void shiftOut(int myDataPin, int myClockPin, byte myDataOut) {

    int i=0;
    int pinState;
    pinMode(myClockPin, OUTPUT);
    pinMode(myDataPin, OUTPUT); //Ausgaenge setzen fuer Datenpin und Clock pin des Schieberegisters
    digitalWrite(myDataPin, 0);
    digitalWrite(myClockPin, 0); //Leitung auf 0 setzen um Fehler zu vermeiden,
                                //wenn vorher andere Signale angelegt waren
    for (i=7; i>=0; i--) { //acht Bit werden eingeschoben
        digitalWrite(myClockPin, 0); //Clock wird 0 gesetzt damit erstes Bit angelegt werden kann
        if ( myDataOut & (1<<i) ) { //Maskierung der jeweiligen Stelle der 8 Bit
                                //z.B. i=6 und Daten = 11010100
                                //ausgabe 01000000 somit 6te PIN ausmaskiert

            pinState= 1;
        }
        else {
            pinState= 0;
        }
        digitalWrite(myDataPin, pinState); //anlegen des Bits
        digitalWrite(myClockPin, 1); //anlegen an Ausgang des Schieberegisters
        digitalWrite(myDataPin, 0);
    }

    digitalWrite(myClockPin, 0); //fertig mit shiften
}

```

## Probleme

Wir hatten anfangs einige Probleme beim Löten. Zuerst verwendeten wir bleifreies Lötzinns, was das Löten wesentlich schwieriger machte, weshalb wir auch sehr schnell auf bleihaltiges Lötzinns umstiegen. Die Metallringe auf der Lochrasterplatine lassen sich außerdem sehr leicht ablösen, nach dem Benutzen einer Lötpumpe waren alle Ringe im Umkreis verschwunden und auch das unsanfte Berühren mit dem LötKolben reichte einige Male aus, um die Ringe zu lösen. Auch der Kupferlackdraht machte Probleme, da wir teilweise nicht lange genug den Draht erhitzen und somit noch Lack zwischen dem Lötzinns und dem Draht war, was den Stromfluss verhinderte. Nach langem Nachlöten fanden wir aber die betroffenen Stellen und der LED-Cube funktionierte.

## Arbeitsaufteilung:

Josua Benz: Löten, Muster entwickeln, Programm entwickeln, Dokumentation

Patrick Koss: Schaltplan zeichnen