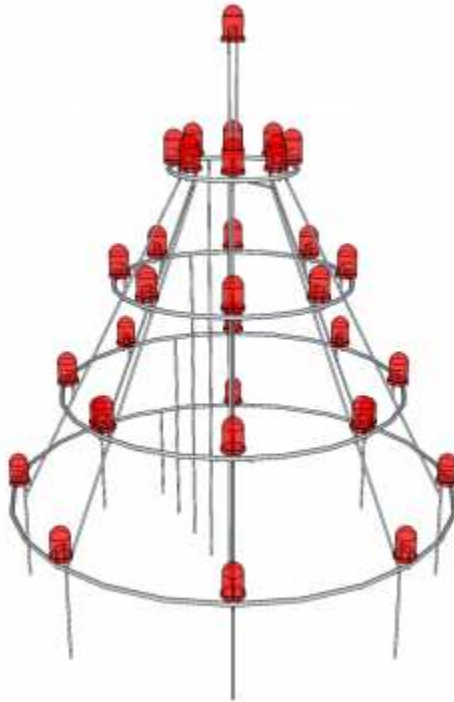


Xmastree

Aus [a x e d i]

So soll er mal aussehen, der LED Xmas Tree



Inhaltsverzeichnis

- 1 Idee & Planung
- 2 LED Tree
 - 2.1 Material
 - 2.2 Zeichnungen
 - 2.3 Zusammenbau
- 3 Sockel
- 4 Steuerung
 - 4.1 Hardware
 - 4.2 Zusammenbau
 - 4.3 Software
- 5 Video

Idee & Planung



Er besteht aus 33 LEDs. Diese sind aufgeteilt auf 5 Ebenen. Auf den unteren

4 Ebenen befinden sich jeweils 8 LEDs. Die fünfte Ebene (Spitze) hat nur eine LED. Gesteuert wird er daher über 8 Spalten und 5 Ebenen. Die Spitze ist eine eigene Ebene und hängt an der Spalte 1. Es werden daher 13 Steuerleitungen (8 Spalten + 5 Ebenen) benötigt.

LED Tree

Material

■ LEDs

Technische Daten

Farbe: Pur -Weiß

Öffnungswinkel: 20°

Intensität: 8.000 - 10.000 mcd

Farbtemperatur: 6000°K

Spannung: Typ. 2,8V - Max. 3,6V

Strom: 20mA Typ., 30mA Max

Chip Material: InGaN

Durchmesser: Ø 5mm

Gehäuse: Wasserklar 5mm

■ Widerstände

150 Ohm

■ Silberdraht

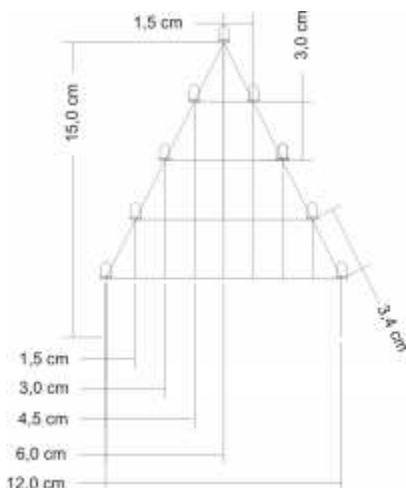
Kupferdraht versilbert

Durchmesser 1mm

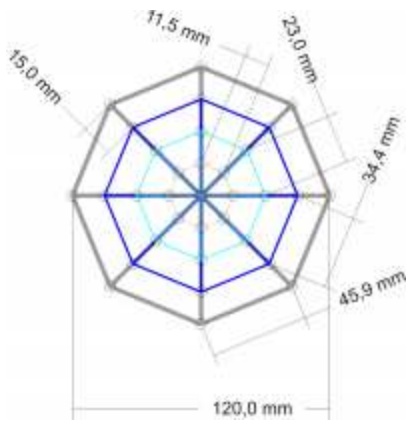
Länge: 4m

Zeichnungen

Wir baut man die Ebenen auf? Zunächst mal muss man sich über die Breite und Höhe des Baums Gedanken machen. Das Verhältnis muss passen, damit es nach einem Baum aussieht. Dieser Baum hier soll eine Höhe von 15cm und eine Breite von 12cm haben.

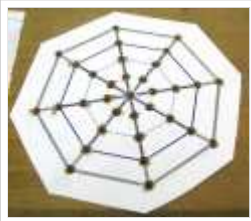


Von oben betrachtet würde der Baum dann so aussehen.



Zusammenbau

Für den Zusammenbau des Xmas Trees benötigt man eine Hilfskonstruktion, um ihn einigermaßen gleichmäßig hin zu bekommen. Dafür habe ich die Zeichnung LED Tree von oben auf ein Holzbrett geklebt (mit Holzleim, hatte gerade nix anderes). Anschließend werden mit einem 5mm Bohrer Löcher an den entsprechenden Schnittpunkten gebohrt (mindestens 8mm tief). Da der Holzleim nicht ganz glatt aufgestrichen war und sich damit ein paar Hügel ergeben haben, wurde das Brett einfach schnell abgeschliffen mit einem Schwingschleifer. Danach kann man die innersten LEDs (Ebene 2) in die Löcher stecken. Dabei ist darauf zu achten, dass die Kathode (oder die abgeflachte Seite) immer nach außen zeigt.



Brett mit aufgeklebter Zeichnung



LEDs auf Brett gesteckt

Die Kathode (kurzes Beinchen) wird zur nächsten LED umgebogen. Dort, wo das Beinchen die nächste LED berührt, bekommt es noch einen Knick, so das es parallel zum nächsten Beinchen besser zu verlöten ist. Ab der dritten Ebene reicht die Länge der Beinchen nicht mehr aus, um zur nächsten LED zu kommen. Jetzt kommt der Silberdraht zu Einsatz. Dieser wird als Verlängerung außen um alle LEDs geführt und mit den Beinchen (Kathode) verlötet. Der Abstand zwischen den LEDs vergrößert sich immer mehr bis zur untersten Ebene.



Ebenen in blau mit
Steuerleitungen



Ebene 2 verlötet



Ebene 4 verlötet

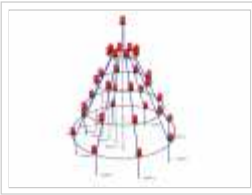


Ebenen 2, 3 und 4
verlötet



Ebenen 2 bis 5 verlötet

Jetzt muss man in die Höhe bauen. Die einzelnen Ebenen müssen miteinander verbunden werden. Auch hier ist darauf zu achten, dass die Ebenen parallel und die LEDs einer Spalte jeweils gerade untereinander verlaufen. Dazu habe ich in das Holzbrett die Ebene 4 gesteckt. In die Löcher der Ebene 3 habe ich dann einfach 8 Dübel der Länge 3 cm gesteckt. Das ist genau der Abstand, der hier geplant war. Dann die Ebene 3 auf die Dübel setzen. Damit das auch hält sind noch ein paar Gewichte drauf gekommen. So, nachdem nun die Halterung fertig ist, kann gelötet werden. Dazu habe ich immer 5cm Stücke vom Silberdraht genommen und zwischen den Anoden der Ebenen 3 und 4 befestigt. Das gleiche Spielchen mit den nächsten Ebenen 2 und 1.



Spalten in blau





Als letztes wird noch die LED an der Spitze angelötet (Ebene 5). Damit sind dann alle Ebenen zusammengebaut und alle 8 Spalten mit allen LEDs vollständig. Jetzt müssen noch die 5 Steuerleitungen der Ebenen nach unten geführt werden.



Sockel

Der Sockel besteht aus einer 5mm dicken Holzplatte. Größe: 160 x 160 x 40 mm

Für die 13 Bohrlöcher habe ich eine Schablone erstellt. In die Bohrlöcher werden Hülsen gesteckt. In die Hülsen werden die 13 Leitungen des LED Trees gesteckt. Von unten werden dann die Leitungen mit den Hülsen verlötet.



Steuerung

Die Schaltung für die Steuerung des LED Tree basiert auf einem Amtel Atmega 328 mit Bootloader aus dem Arduino Projekt [1] (<http://www.arduino.cc>) .

Hardware

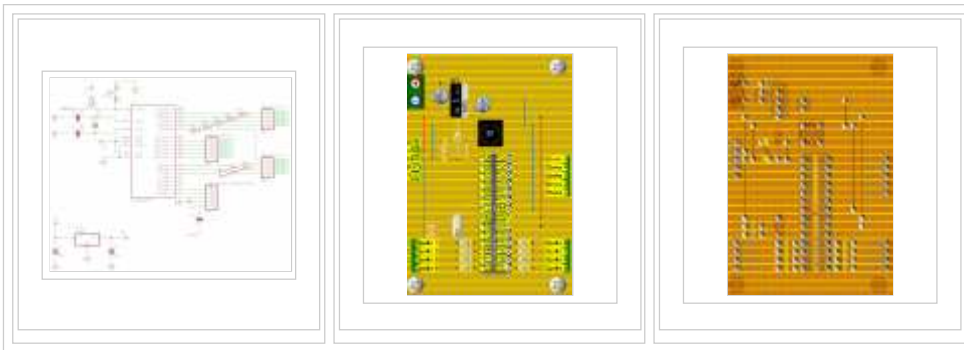
Folgende Bauteile werden benötigt:

- 2x Kondensator 22pF [2] (<http://www.reichelt.de/?ACTION=3;ARTICLE=9281;PROVID=2402>)
- 1x Kondensator 10uF [3] (<http://www.reichelt.de/?ACTION=3;ARTICLE=23716;PROVID=2402>)
- 1x Kondensator 100nF [4] (<http://www.reichelt.de/?ACTION=3;ARTICLE=9265;PROVID=2402>)
- 1x IC-Sockel, 28-polig [5] (<http://www.reichelt.de/?ACTION=3;ARTICLE=8222;PROVID=2402>)
- 1x Widerstand 1KOhm [6] (<http://www.reichelt.de/?ACTION=3;ARTICLE=1338;PROVID=2402>)
- 1x Spannungsregler 7805 [7] (<http://www.reichelt.de/?ACTION=3;ARTICLE=23443;PROVID=2402>)
- 1x 36-polige Stiftleiste [8] (<http://www.reichelt.de/?ACTION=3;ARTICLE=19504;PROVID=2402>)
- 1x Kurzhubtaster 6x6mm [9] (<http://www.reichelt.de/?ACTION=3;ARTICLE=27892;PROVID=2402>)
- 1x Quarz 16 MHz [10] (<http://www.reichelt.de/?ACTION=3;ARTICLE=1605;PROVID=2402>)
- 1x Streifenrasterplatine 50x100mm [11] (<http://www.reichelt.de/?ACTION=3;ARTICLE=8275;PROVID=2402>)
- 1x AnreihKlemme 2-polig [12] (<http://www.reichelt.de/?ACTION=3;ARTICLE=36615;PROVID=2402>)
- 1x ATmega328 mit Arduino Bootloader [13] (<http://www.watterott.com/ATmega328-mit-Arduino-Bootloader>)
- 1x FTDI Basic Breakout 5V [14] (<http://www.watterott.com/FTDI-Basic-Breakout-5V>)
- 1x Jumper Wires Premium M/F [15] (<http://www.watterott.com/Jumper-Wires-Premium-M-F-150mm>)

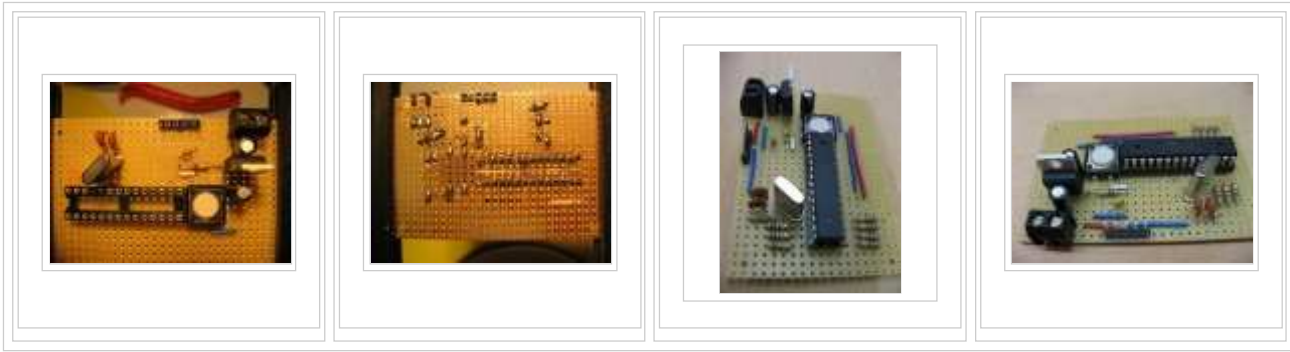
Zusammenbau

Zunächst habe ich mit Eagle einen Schaltplan gezeichnet. Da der Arduino mit 5V läuft muss die Stromquelle entweder 5V liefern oder es muss ein Spannungsregler her. Hier ist 7805 im Einsatz (Schaltplan links unten). Der Atmega328 bekommt einen 16MHz Quarz, einen Resettaster und einen Anschluß für ein FTDI-Board (USB-to-serial-communication). Bei den 9 Spalten ist jeweils ein 150 Ohm Widerstand eingebaut.

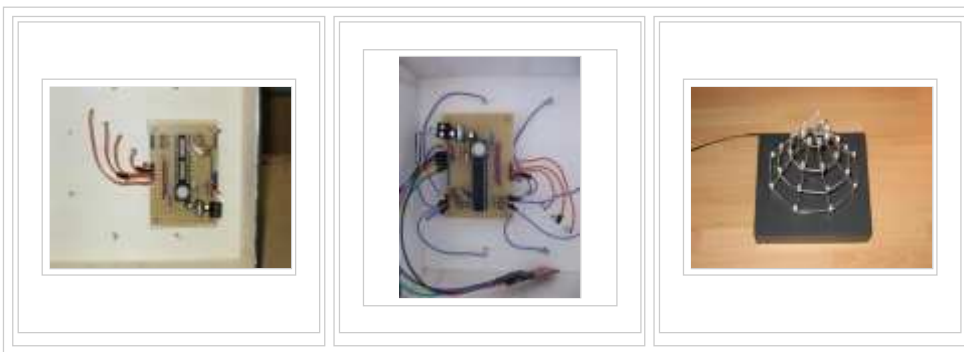
Der Schaltplan muss jetzt irgendwie auf eine Streifenrasterplatine gebracht werden. Dabei hilft die Software Lochmaster.



Jetzt (nur) noch alles zusammenlöten und hoffen, das man keine Fehler gemacht hat.



Die Platine wurde dann mit Hilfe von Holzabstandshaltern von unten auf den Sockel geklebt. Die 5 Ebenen sind mit dem orangen Draht mit dem Atmega verbunden. Die 8 Spalten sind mit lila Drähten verbunden. Als letztes noch den Atmega in den Sockel stecken, Strom anlegen und... nix passiert. Die Hardware ist fertig, jetzt folgt der Programmierteil!



Software

```

#include <avr/pgmspace.h>

#define COLUMN 8          // Anzahl der Spalten
#define PLANE 5           // Anzahl der Ebenen
#define PLANETIME 3333    // Anzeigedauer einer Ebene
#define TIMECONST 15      // Multiplikator für Anzeigedauer eines Bildes (33 LEDs)

// LED Muster: Eine Zeile besteht aus den 5 Ebenen mit je 8 LEDs (außer Ebene 5 mit einer LED) und der Anze
prog_uchar PROGMEM PatternTable[] = {
    B11111111, B11111111, B11111111, B11111111, B1, 5,    // Alle LEDs an
    B00000000, B00000000, B00000000, B00000000, B0, 1,    // Alle LEDs aus
    B11111111, B11111111, B11111111, B11111111, B1, 5,
    B00000000, B00000000, B00000000, B00000000, B0, 10,
    // Bei Anzeigedauer 0 endet die Schleife
    B00000000, B00000000, B00000000, B00000000, B0, 0
};

// Definition der Pins für Ebenen und Spalten (hängt davon ab, wie man sie verlötet hat)
int LedPin[] = {7, 8, 6, 9, 10, 11, 12, 5};
int PlanePin[] = {18, 17, 16, 15, 14};

void setup()
{
    int pin;
    // Pins als OUTPUT initialisieren
    for (pin=0; pin<COLUMN; pin++) {
        pinMode( LedPin[pin], OUTPUT );
    }

    for (pin=0; pin<PLANE; pin++) {
        pinMode( PlanePin[pin], OUTPUT );
    }
}

void loop()
{
    byte PatternBuf[PLANE]; // speichert eine Zeile aus LED Muster
    int PatternIdx;
    byte DisplayTime;
    unsigned long EndTime;
    int plane;
    int patbufidx;
    int ledpin;

    // Index in LED Muster setzen
    PatternIdx = 0;
    // Wiederholen solange Anzeigedauer > 0
    do {
        // Zeile aus LED Muster lesen
        memcpy_P( PatternBuf, PatternTable+PatternIdx, PLANE );
        PatternIdx += PLANE;
        // Anzeigedauer lesen
        DisplayTime = pgm_read_byte_near( PatternTable + PatternIdx++ );
        // Anzeigedauer für Bild berechnen
        EndTime = millis() + ((unsigned long) DisplayTime) * TIMECONST;

        // LEDs der Ebenen immer wieder aufbauen, solange Anzeigedauer für Bild noch nicht erreicht
        while ( millis() < EndTime ) {
            patbufidx = 0; // Am Anfang der Zeile beginnen
            // Schleife über Ebenen
            for (plane=0; plane<PLANE; plane++) {
                // vorherige Ebene ausschalten
                if (plane==0) {
                    digitalWrite( PlanePin[PLANE-1], HIGH );
                } else {
                    digitalWrite( PlanePin[plane-1], HIGH );
                }

                // Spalten der aktuelle Ebene setzen

                // Ebene 5 gesondert behandeln, da nur eine LED
                if (plane==PLANE-1) {
                    digitalWrite( LedPin[0], PatternBuf[plane] );
                } else {
                    // Einzelne Spalten setzen
                    for (ledpin=0; ledpin<COLUMN; ledpin++) {
                        // Bitoperation zur Ermittlung der einzelnen Bitwerte
                        // Beispiel:
                        // LED Zeile & ledpin (left shift (<<)) schiebt ledpin um 1 nach rechts

```


Video

Von „<http://m12s23.vlinux.de/mediawiki/index.php/Xmastree>“

- Diese Seite wurde zuletzt am 11. Dezember 2009 um 19:58 Uhr geändert.