

## Mini LEDCube v0.2 cthn.de

Zuvor sollte die Liste der Bauteile mit der in der Verpackung enthaltenen Anzahl verglichen werden.

### Allgemeines zum Aufbau:

Begonnen wird mit dem Bauteil, das die geringste Höhe nach dem Einlöten besitzt. Dies führt dazu, dass die höheren Bauteile beim Einlöten der anderen nicht im Weg sind.

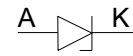
**Warnhinweise!** Der Mikrocontroller wird erst nachdem die Schaltung komplett aufgebaut ist und optisch und elektrisch getestet wurde eingesetzt. Bevor der LEDCube an die Schaltung angeschlossen wird, muss der Mikrocontroller konfiguriert (FuseBits gesetzt) und programmiert werden. Zudem muss die Funktion der Firmware zwingend geprüft werden. Sollte ein Fehler in der Schaltung (Bauteile vertauscht, ungewollte Lötbrücken, etc.) oder in der Firmware vorhanden sein, können die LEDs des Cube irreparabel zerstört werden.

### Werkzeuge/Hilfsmittel:

- Seitenschneider
- Elektronikzange
- LötKolben mit möglichst feiner Spitze
- Dritte Hand mit Lupe
- Entlötlitze

## Aufbau:

### 1. Zehner-Dioden



D1,2 – 3V6 Zehner-Diode

Der schwarze Ring an den Zehner-Dioden kennzeichnen die Kathode. Es muss auf die korrekte Polung beim einlöten geachtet werden. Die Bauteile werden liegend eingebaut.

In Durchlass-Richtung funktionieren Z-Dioden wie eine normale Diode. In Sperr-Richtung dient sie zur Spannungsbegrenzung. In dieser Schaltung werden die beiden Z-Dioden in Sperr-Richtung betrieben. Sie dienen dazu die Spannungen an den beiden USB-Datenleitungen (D+ und D-) auf 3,3 Volt (von dem 5V Pegel des Mikrocontrollers) herab zu senken. Siehe <http://vusb.wikidot.com/hardware> (Solution B).

### 2. Dioden



D3,4 – 1n4148

Auch hier kennzeichnet der schwarze Ring die Kathode und muss beim Einbau beachtet werden. Auch diese Dioden werden liegend eingebaut.

D3 dient als Verpolschutz an der Versorgungsklemme X2.

D4 verbindet den Ausgang des Spannungsreglers IC2 mit dessen Eingang. Hierdurch wird verhindert, dass am Ausgang eine höhere Spannung anliegen kann als am Eingang.

Dieser Zustand kann zum zerstören des Spannungsreglers führen.

### 3. IC-Sockel

Der Sockel für den IC1 – ATTINY2313. Es muss darauf geachtet werden, dass die Markierung für den ersten und letzten Pin (Ausbuchtung in Form eines Halbkreises am Sockel und Mikrocontroller) korrekt auf der Platine aufgesetzt wird und dass der Sockel eben auf der Platine auf sitzt.

#### 4. Kondensatoren



Alle Kondensatoren werden stehend eingelötet.

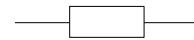
C1,2 – 27pF

Die Kondensatoren C1 und C2 werden dazu verwendet um das Quarz, das den Mikrocontroller mit seinem Takt versorgt, in Schwingung zu versetzen.

C3,4,5 – 100nF

Diese sind Entkopplungskondensatoren und unterdrücken zum einen das rauschen der Versorgungsspannung(en) und stabilisieren diese zusätzlich auf einen möglichst konstanten Wert.

#### 5. Widerstände



Die Widerstände müssen alle stehend eingelötet werden. Die Widerstände sollten dabei möglichst so gesetzt werden, dass die sich gegenseitig wenig durch gegenseitiges berühren unter mechanischer Spannung stehen.

R3,4 – 69 Ohm Farbcode: Blau, Weiß, Schwarz, Gold

Funktionieren mit den Z-Dioden als Spannungsbegrenzung.

R2 – 1,5k Ohm Farbcode: Braun,Grün,Rot,Gold

R2 ist ein PullUp-Widerstand, für die D+-Datenleitung. Ein PullUp-Widerstand hält eine Leitung auf einer konstanten Spannung. Hier auf ca. 3,3V da R2 und D2, wie oben beschrieben, die Spannung von 5V herunter begrenzen.

R1 – 10k Ohm Farbcode: Braun,Schwarz,Orange,Gold

Der Widerstand R1 wird dazu verwendet um den RESET-Pin am Mikrocontroller konstant auf 5V zu halten (PullUp). Der RESET-Pin ist ein sogenannter ActiveLow-Pin und setzt den Mikrocontroller zurück sobald die an diesem anliegende Spannung einen bestimmten Grenzwert unterschreitet (Bei 5V TTL-Pegeln ca. 0,8V). TTL = Transistor-Transistor-Logik.

R5 – 1M Ohm (optional) Farbcode: Braun,Braun,Grün,Gold

Dieser Widerstand wird als PullDown betrieben. Er ist gegen Masse verbunden (Minus). Die „D“-Leitung wird hiermit konstant auf 0V gehalten.

## **6. Stecker**

Diese Stecker sollten rechtwinklig (eben) auf der Platine aufgelötet werden. Die Stecker müssen so ausgerichtet sein, dass die schwarzen Leitungen alle näher beim Mikrocontroller aufgesteckt werden als die roten.

SV1,2,3,4 – 3-Pin Stecker

Hier wird später der LED-Würfel angeschlossen. Die Einbaurichtung muss beachtet werden, da sonst die LEDs im Würfel nicht korrekt angesteuert werden.

An SV4 werden die Leitungen A,B,C ausgeführt, an SV1 die Leitungen 1,2,3 usw.

## **7. ISP-Stecker**

Diese Stecker sollten rechtwinklig (eben) auf der Platine aufgelötet werden.

SV5 – 2x3Pin Stiftleiste

Die 6Pin Stiftleiste muss in 2 gleichgroße Teile getrennt werden. Über diesen 2x3 Pin Anschluss wird später der Mikrocontroller beschrieben. Die 6-Pin Buchse ist bei AVR's der

Standard, um den ISP (= In System Programmer) daran anzuschließen.

## 8. Anschlussklemme

Diese Stecker sollten rechtwinklig (eben) auf der Platine aufgelötet werden.

X2 – Anschlussklemme

Diese Anschlussklemme kann dazu verwendet werden die Schaltung und die LEDs mit Strom zu versorgen. Die maximale Eingangsspannung sollte 5V nicht unterschreiten 12Volt nicht überschreiten.

## 9. Spannungsregler

Dieses Bauteil wird stehend aufgelötet. Es verbleibt eine Lücke von ca. 5mm zwischen Bauteil und Platine. Sieht man auf die flache Seite des Reglers während die Beine nach unten zeigen, dann entspricht das linke Bein dem Ausgang, das mittlere der Masse und das rechte dem Eingang.

IC2 – 78L05

Der 78L05 ist ein Festspannungsregler. Er wandelt die Eingangsspannung in eine feste Ausgangsspannung von +5V. Er benötigt am Ein- und Ausgang Kondensatoren um eine stabile und möglichst konstante Spannung zu erzeugen. Im Gegenteil zu einem 7805 Festspannungsregler kann der 78L05 einen maximalen Strom von 100mA liefern.

## 10. Quarz



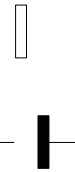
Auch dieses Bauteil wird stehend aufgelötet. Hier wird zwischen Bauteil und Platine möglichst kein platz frei gelassen.

Q1 – 12 Mhz Quarz

Das Quarz wird dazu verwendet um den Mikrocontroller mit einem Takt von 12MHz zu betreiben. Der Quarz wird stets dicht am Mikrocontroller angebracht, um so den Takt nicht

durch andere Komponenten zu verfälschen.

## 11. Elko



Um den Elko aufzubringen, muss bei diesem zunächst die Anschlüsse so umgebogen werden, dass dieser stehend aufgebracht werden kann (nur bei axialen Bauteilen notwendig). Bei einem Elko (Elektrolytkondensator) ist auf die Polung dessen zu achten! Die Kathode (minus) ist hier meist mit einem Streifen, einer Kerbe oder einem oder mehreren „-“-Symbolen gekennzeichnet bzw. einem Pfeil markiert.

C6 – 4,7 $\mu$ F

Der Elko dient dazu um die Ausgangsspannung des Spannungsreglers weiter zu stabilisieren und zu filtern.

## 12. USB Buchse

Auch hier ist wieder darauf zu achten, dass die Buchse eben aufliegt. Es müssen alle Pins und die Verlängerungen des Gehäuses aufgelötet werden. Diese dienen dazu die mechanische Festigkeit zu verbessern.

X1 – USB Buchse

Die Schaltung kann auch per USB betrieben/mit Spannung versorgt werden. Die Kommunikation mit dem Mikrocontroller muss jedoch noch implementiert werden.

## **Inbetriebnahme:**

Bevor die Schaltung in Betrieb genommen werden kann muss diese zunächst geprüft werden. Die folgenden Schritte beschreiben diesen Vorgang. Der IC1 (AVR) ist bei diesem Vorgang noch nicht auf der Platine aufgesteckt!

### **1. Optisch Prüfen und durchmessen**

Die Lötstellen müssen allesamt einzeln darauf geprüft werden ob benachbarte Leiterbahnen unbeabsichtigt mit verbunden sind. Diese Verbindungen können vorsichtig mit dem LötKolben (in Verbindung mit Entlötlitze oder einer Entlötpumpe) entfernt werden.

Sind keine solchen Verbindungen zu erkennen kann als nächstes mit einem Multimeter der Durchgang der einzelnen Lötstellen zu den im Schaltplan vorgesehenen anderen Lötstellen durchgemessen werden. Es empfiehlt sich auch die Leiterbahnen untereinander zu prüfen, die keine Verbindung haben sollten aber sehr nah beieinander liegen. Achtung, durch die Kondensatoren auf der Schaltung kann das Messgerät unter Umständen, zwischen manchen Leiterbahnen, kurz einen Durchgang anzeigen bzw. einen Ton von sich geben auch wenn keine direkte Verbindung vorhanden ist.

### **2. Testen der Spannungen**

Ist die Schaltung geprüft kann nun eine externe Spannungsquelle (nicht USB) an die Schaltung über die Anschlussklemme X2 angeschlossen werden. Es sollte kein Kurzschluss entstehen. Wenn sich ein Kurzschluss abzeichnet muss die Schaltung nochmals geprüft werden.

Zu prüfen sind die folgenden Punkte:

- an der Anschlussklemme X2 liegt die Eingangsspannung an (max. 12V)
- am Spannungsregler (Ausgang) liegen +5V an
- zwischen Pin 1 und Pin4, des USB-Steckers, liegen +5V an
- an allen Kondensatoren liegt eine Spannung von 5V an
- zwischen Pin 10 und 20 des IC-Sockels müssen 5V anliegen, wobei +5V mit Pin 10

- und Pin 20 mit GND (-) verbunden ist
- zwischen Pin1 des IC-Sockels und GND (-) liegen ebenfalls +5V an (dies ist die RESET-Leitung)
  - alle anderen Pins des IC-Sockel führen keine Spannung und sind nicht mit GND verbunden
  - zwischen Pin 2 und Pin 6, des ISP-Steckers, liegen ebenfalls +5V an
  - Pin 5 des ISP-Steckers ist mit Pin 1 des IC-Sockels verbunden

### 3. Mikrocontroller

Wenn die Schaltung erfolgreich getestet wurde, kann der ATTiny2313 eingesetzt werden (Schaltung Spannungsfrei). Darauf wird ein Programmiergerät an den ISP-Stecker angeschlossen und danach erst die Schaltung mit einer Spannungsquelle verbunden. Da die Schaltung bereits geprüft wurde, darf auch der USB-Stecker zur Versorgung verwendet werden. Im Git-Repository befindet sich im „firmware“-Verzeichnis (neben dem Sourcecode) ein Makefile, dass dazu verwendet werden kann um auf den AVR zuzugreifen.

#### 3.1 Fuse-Bits setzen

Über die Fuse-Bits wird das grundlegende Verhalten des AVR festgelegt. Diese Bits werden beim ATTiny2313 durch 3 Byte repräsentiert (wir beachten nur die beiden niederwertigen). Bei diesem Projekt beinhaltet das Byte mit niederer Wertigkeit (Lowbyte) den Hex-Wert 0xE4 und das höherwertige Byte den Wert 0xDF.

Die Standardeinstellungen des Lowbyte ist der Wert 0x64. Der Unterschied zu unseren Einstellungen ist, dass der Systemtakt durch 8 geteilt wird (es wird der interne 8MHz Oszillator verwendet). Mehr Details zu diesen Einstellungen können hier eingesehen werden. – <http://www.engbedded.com/fusecalc/>

Die Fuse-Bits können mit „make fuses“ im AVR festgelegt werden. **ACHTUNG!!!** Falsch gesetzte Fuses könne dazu führen, dass der AVR nicht mehr per ISP programmiert und oder konfiguriert werden kann! Die Bits „RSTDISBL“ und „SPIEN“ sollten in der Standardeinstellung verbleiben.



### 3.2 AVR mit Firmware flashen

Sind die Fuses korrekt übertragen worden, kann die Firmware übertragen werden. Das Makefile beinhaltet bereits die Option „compile“ zum kompilieren und „program“ zum übertragen. Mit der Option „info“ können weitere Informationen zum erzeugten Binary abgerufen werden.

Unter Umständen muss noch ein Reset der Schaltung durchgeführt werden, bevor die Firmware aktiv ist.

**Achtung**, die originale Firmware aus dem Projekt von Mosfetkiller beinhaltet keinen USB-Support und wird daher von keinem System als korrekt funktionierendes USB-Gerät erkannt. Es kann zu diversen Fehlermeldungen kommen, wenn die Schaltung per USB mit Spannung versorgt wird.

### 3.3 Testen der Firmware

In diesem Schritt soll die Firmware mit einem Multimeter getestet werden. Spannungen an den Steckverbindern SV1-4 geprüft werden. Es müssen die folgenden Eigenschaften geprüft werden:

- die Spannungen zwischen den Pins an SV1-3 und den Pins von SV4 müssen deutlich geringer als 5V sein
- zwischen den Pins von SV4 zur Masse muss ebenfalls eine geringere Spannung als 5V gemessen werden können
- +5V (z.B. dem Ausgang des Spannungsreglers) zu den Pins von SV1-3 muss ebenfalls eine geringere Spannung als +5V angezeigt werden

Damit ist der Aufbau abgeschlossen und der LEDCube kann mit der Schaltung getestet werden.