

Lötübung Digitaluhr

**Inhalt**

[**1. Sinn und Zweck** 3](#_Toc508465133)

[**2. Grobkonzept** 3](#_Toc508465134)

[**3. Skizze** 3](#_Toc508465135)

[**4. Schaltungsentwicklung** 4](#_Toc508465136)

[4.1 LEDs Ansteuerung 4](#_Toc508465137)

[4.1.1 Beschreibung 4](#_Toc508465138)

[4.1.2 Schema 4](#_Toc508465139)

[4.1.3 Berechnungen 4](#_Toc508465140)

[4.2 Ansteuerung der Anzeige 5](#_Toc508465141)

[4.3 Anschlüsse 6](#_Toc508465142)

[4.4 Gesamtschema 7](#_Toc508465143)

[4.5 Bauteilliste 8](#_Toc508465144)

[**5. Layout** 8](#_Toc508465145)

[5.1 PCB Layout 9](#_Toc508465146)

[5.2 3D Ansicht 10](#_Toc508465147)

[5.2.1 Ansicht von Oben 10](#_Toc508465148)

[5.2.2 Ansicht von Unten 11](#_Toc508465149)

[5.3 Bemassung der LEDs 12](#_Toc508465150)

[**6. Spezifikationen** 13](#_Toc508465151)

[**7. Einsatzmöglichkeiten** 13](#_Toc508465152)

[**8. Anpassungen** 13](#_Toc508465153)

[**9. RTC Modul (Real Time Clock)** 14](#_Toc508465154)

[9.1 Installation RTCLib 14](#_Toc508465155)

[9.2 Schema von RTC 15](#_Toc508465156)

[9.3 Layout 15](#_Toc508465157)

[9.4 Bestückungsliste 16](#_Toc508465158)

# **1. Sinn und Zweck**

Es soll eine Digitaluhr entwickelt werden, bei der die Lötfähigkeit trainiert und angewendet werden können. Der Umfang der Aufgabe sollte dabei in einem angemessenen Masse liegen, welche für das erste Lehrjahr realisierbar ist.

# **2. Grobkonzept**

Das Ziel ist es, eine 7-Segment-Anzeige mit einzelnen LEDs zu entwickeln. Die Segmente sollen einzeln und zusätzlich über einen Decoder angesteuert werden können. Das Ganze besteht aus den sieben Einzelsegmenten, einem Dezimalpunkt und zwei LEDs, welche einen Doppelpunkt bilden.

# **3. Skizze**

# **4. Schaltungsentwicklung**

## 4.1 LEDs Ansteuerung

### 4.1.1 Beschreibung

Die einzelnen Segmente der Anzeige bestehen aus jeweils vier LEDs, welche einzeln mit einem Vorwiderstand beschaltet werden. Als Treiber wird ein einfacher BC547 Transistor verwendet. Das Steuersignal um das Segment anzusteuern liegt bei 5VDC. Die Speisung der LEDs erfolgt ebenfalls mit 5VDC.

### 4.1.2 Schema

### 4.1.3 Berechnungen

#### 4.1.3.1 Basiswiderstand

gegeben:

gesucht:

**19710 Ω** (gewählt 2kΩ)

Die LEDs benötigen einen Kollektorstrom von 40 mA, deshalb muss mit dem Treiber garantiert werden, dass ein solcher Strom auch fliessen kann. Damit noch eine gewisse Reserve vorhanden ist, wird ein Kollektorstrom von bis zu 80mA gewählt. Dieser wird natürlich nicht erreicht, da die Vorwiderstände der LEDs den Strom auf die 40 mA begrenzen (10 mA pro LED).

#### 4.1.3.2 Vorwiderstand

gegeben:

gesucht:

**180 Ω**

## 4.2 Ansteuerung der Anzeige

Zur Ansteuerung der Anzeige wird ein einfacher BCD zu 7-Segment-Decoder (74HC4511) verwendet.

Der Decoder besitzt 4 BCD-Eingänge, wodurch die einzelnen Segmentausgänge von A bis F angesteuert werden. Die Segmentausgänge müssen dadurch nur noch mit den Treibern verbunden werden, welche die jeweiligen Segmente schalten. Durch die Übersteuerung der Transistoren ist garantiert, dass die LEDs immer die 10 mA habe, auch wenn der Basisstrom kleiner ist als geplant.

|  |  |  |
| --- | --- | --- |
|  | **74HC4511** |  |
| Bildergebnis für 74HC4511 |
| BCD B Input | Power supply |
| BCD C Input | Segment F |
| Lamp Test | Segment G |
| Blank | Segment A |
| Latch/Store | Segment B |
| BCD D Input | Segment C |
| BCD A Input | Segment D |
| Power ground | Segment E |
|  |  |

## 4.3 Anschlüsse

Die Anschlüsse des Decoders sowie die Anschlüsse für die separate Ansteuerung der Segmente werden auf zwei Stiftleisten geführt. Somit kann einfach gewählt werden, ob die Anzeige über den Decoder oder einzeln Angesteuert werden soll.

|  |  |
| --- | --- |
| Pin 1 | ENL (Latch) |
| Pin 2 | BL (Blanking Input) |
| Pin 3 | LT (Lamp Test) |
| Pin 4 | D0 |
| Pin 5 | D1 |
| Pin 6 | D2 |
| Pin 7 | D3 |
| Pin 8 | LED 29 (Dez. Punkt) |
| Pin 9 | GND |
| Pin 10 | Vcc |

|  |  |
| --- | --- |
| Pin 1 | Segment a |
| Pin 2 | Segment b |
| Pin 3 | Segment c |
| Pin 4 | Segment d |
| Pin 5 | Segment e |
| Pin 6 | Segment f |
| Pin 7 | Segment g |
| Pin 8 | LED 29 (Dez. Punkt) |
| Pin 9 | GND |
| Pin 10 | Vcc |
| Pin 11 | LED 31/32 (Doppelpunkt) |
| Pin 12 | unbenutzter Pin |

## 4.4 Gesamtschema

## 4.5 Bauteilliste

|  |  |  |  |
| --- | --- | --- | --- |
| **Bauteil** | **Bezeichnung** | **Wert** | **Kosten** |
| Leuchtdiode | LED1 bis LED31 | Blau, 3 V, 20 mA | \* |
| Vorwiderstand | R1 bis R31 | 390 Ω | \* |
| Basiswiderstand | RB1 bis RB7 | 2 kΩ | \* |
| Transistor (Treiber) | T1 bis T7 | BC547C | \* |
| BCD-Decoder | IC1 | 74HC4511 | \* |
| IC-Sockel | Sockel | 16 Pin / DIL 16 | \* |
| Stiftleiste | Stiftleiste 1 | K1 10 Pin | \* |
| Stiftleiste | Stiftleiste 2 | K2 12 Pin | \* |

\*noch zu ermitteln

# **5. Layout**

Das Layout wird mit einem Zeichnungsprogramm erstellt. Zur Verfügung stehen dabei Eagle, TINA und Target T3001. Das erste Layout wurde mithilfe von TINA erstellt, dies erwies sich jedoch als recht kompliziert, da das Programm schnell Mühe hat mit komplexeren Layouts. Mit Eagle kann sehr schnell und einfach ein anständiges Layout erstellt werden aber das Programm bietet keine 3D Ansicht des fertigen Boards. Somit wurde das ganze Schema mit Hilfe von Target aufgezeichnet und danach ein Layout erstellt. Mit Target ist es dann auch möglich eine 3D-Ansicht der fertigen Leiterplatte zu erstellen.

## 5.1 PCB Layout

## 5.2 3D Ansicht

### 5.2.1 Ansicht von Oben

### 5.2.2 Ansicht von Unten

## 5.3 Bemassung der LEDs

# **6. Spezifikationen**

Speisespannung: 5VDC

Stromverbrauch: pro Segment ca. 40 mA

max: 320 mA

**Bemerkung:**

Wenn die 7-Segmentanzeige an einen Mikrocontroller angeschlossen wird, muss darauf geachtet werden, ob der Mikrocontroller den entsprechenden Strom liefern kann. Ansonsten sollte eine externe Speisung verwendet werden.

# **7. Einsatzmöglichkeiten**

Die 7-Segmentanzeige kann mit einem Mikrocontroller angesteuert (auf externe Speisung achten) oder für digitale Schaltungen (z.B. Reaktionstester) verwendet werden. Dabei kann zwischen dem Ansteuern über einen Decoder oder dem direkten betreiben einzelner Segmente ausgewählt werden.

# **8. Anpassungen**

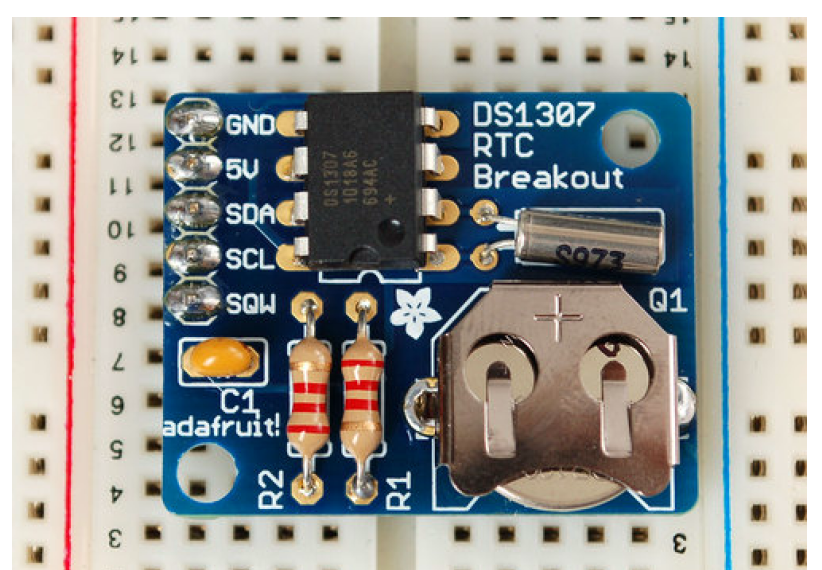
Da die Helligkeit der LEDs auch bei einem Strom von 10 mA noch extrem hoch ist, werden die Vorwiderstände neu dimensioniert. Durch einen Versuchsaufbau, welcher aus der LED und einem Potentiometer besteht, wird durch verändern des Widerstandswertes eine passende Helligkeit eingestellt. Der ermittelte Widerstandswert liegt bei 15 kΩ.

Dies führt zu einem Vorwärtsstrom IF von:

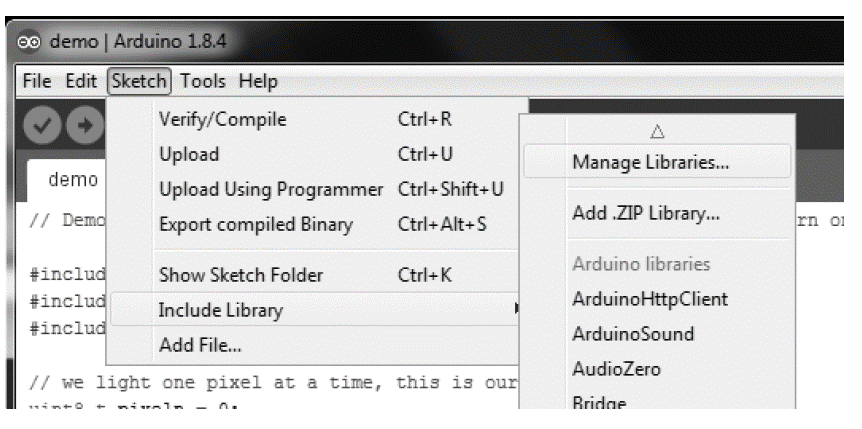
**120 µA**

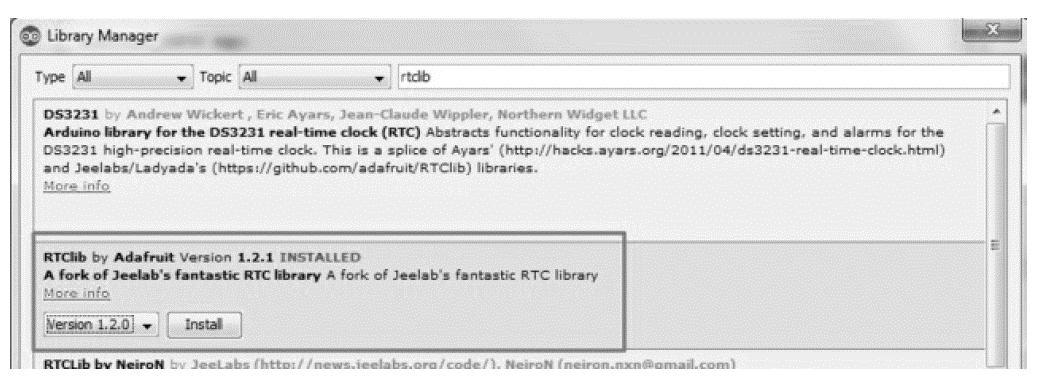
Max. Stromverbrauch: 31 mal 120 µA = **4 mA**

# **9. RTC Modul (Real Time Clock)**

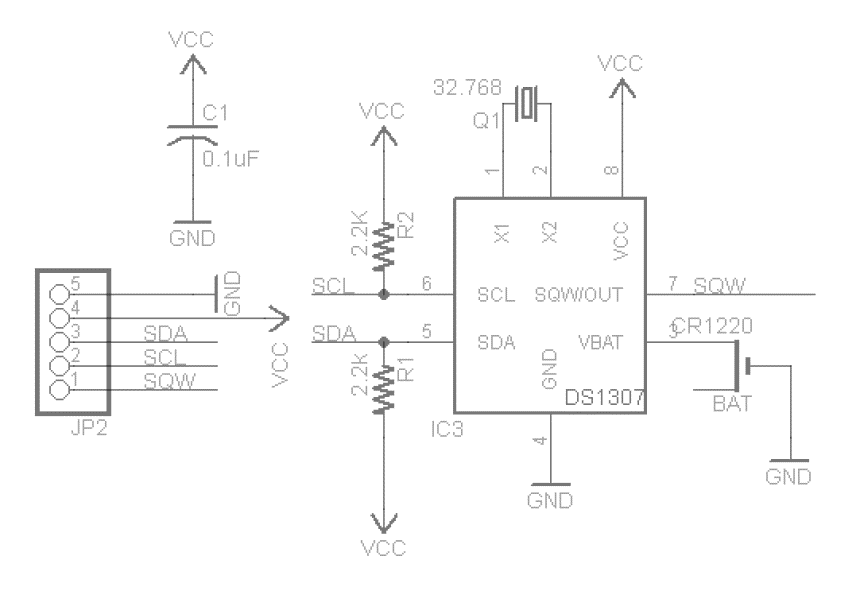


## 9.1 Installation RTCLib

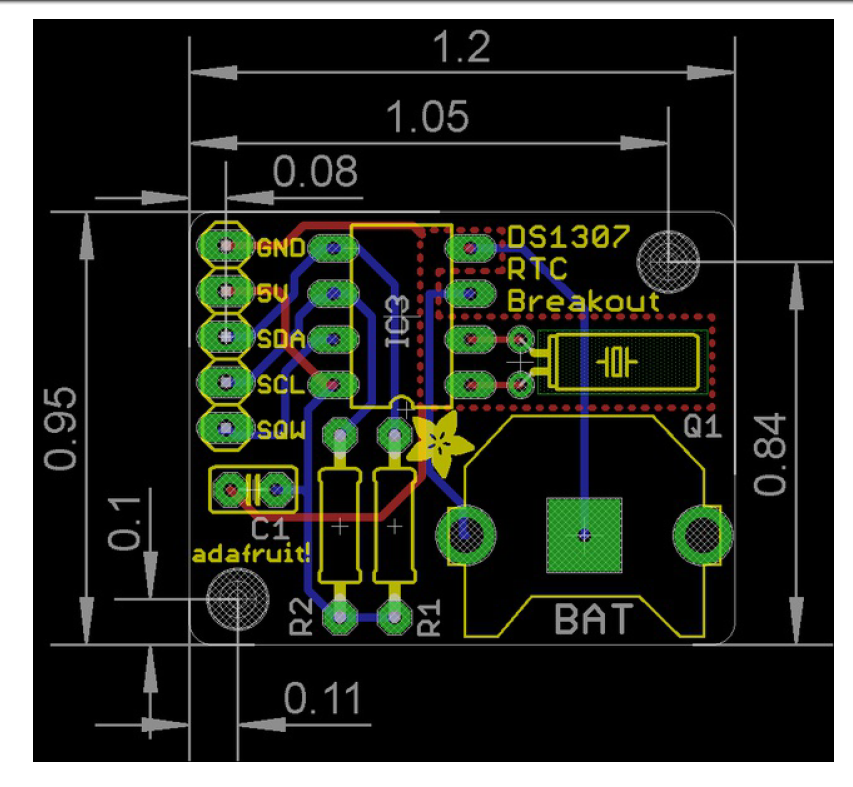




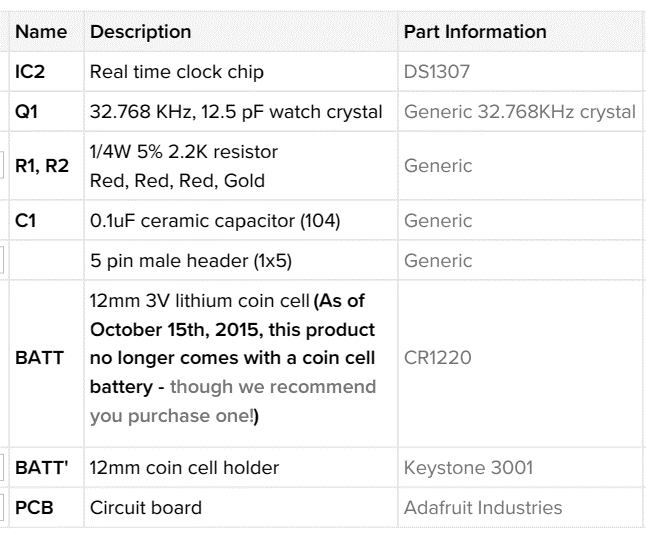
## 9.2 Schema von RTC



## 9.3 Layout

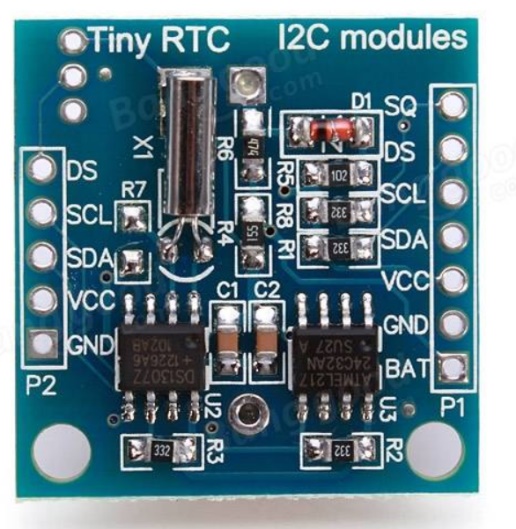


## 9.4 Bestückungsliste



**Aliexpress:**

<https://de.aliexpress.com/item/Tiny-RTC-I2C-modules-24C32-memory-DS1307-clock-RTC-module-without-battery-good-quality-low-price/2020927349.html?spm=a2g0x.search0104.3.43.78d44067BPaeOa&ws_ab_test=searchweb0_0,searchweb201602_4_10152_10151_10065_10344_10068_10342_10343_10340_10341_10084_10083_10618_10307_10302_10134_5711211_10313_10059_10534_100031_10103_10627_10626_10624_10623_10622_5711315_10621_10620_5722413,searchweb201603_25,ppcSwitch_7&algo_expid=82d28a95-5eb5-467a-a95d-cba468e33b86-6&algo_pvid=82d28a95-5eb5-467a-a95d-cba468e33b86&priceBeautifyAB=0>



**Preis: 0,34 €**

**Code:**

*// Bibliotheken für RTC-Modul*

#include <Wire.h>

#include "RTClib.h" *// Sketch --> Include Library*

RTC\_DS1307 rtc;

int UHR[4];

void setup (){

while (!Serial);

Serial.begin(9600);

if (! rtc.begin()){ Serial.println("Couldn't find RTC"); while (1);}*//if*

if (! rtc.isrunning()){ Serial.println("RTC is NOT running!");

rtc.adjust(DateTime(2018, 3, 05, 11, 57, 0)); *//Einstellung für die Zeit RTC*

}*//if*

}*// setup*

void loop (){

DateTime now = rtc.now();

UHR[0] = now.hour(), DEC;

UHR[0] = UHR[0] / 10; *// Erste Ziffer für Stunden: Zehner*

UHR[1] = now.hour(), DEC;

UHR[1] = UHR[1] % 10; *// Zweite Ziffer für Stunden: Einer*

UHR[2] = now.minute(), DEC;

UHR[2] = UHR[2] / 10; *// Dritte Ziffer für Minuten: Zehner*

UHR[3] = now.minute(), DEC;

UHR[3] = UHR[3] % 10; *// Vierte Ziffer für Minuten: Einer*

Serial.print("Uhrzeit: ");

Serial.print(now.hour());

Serial.print(":");

Serial.println(now.minute());

}*//loop*