

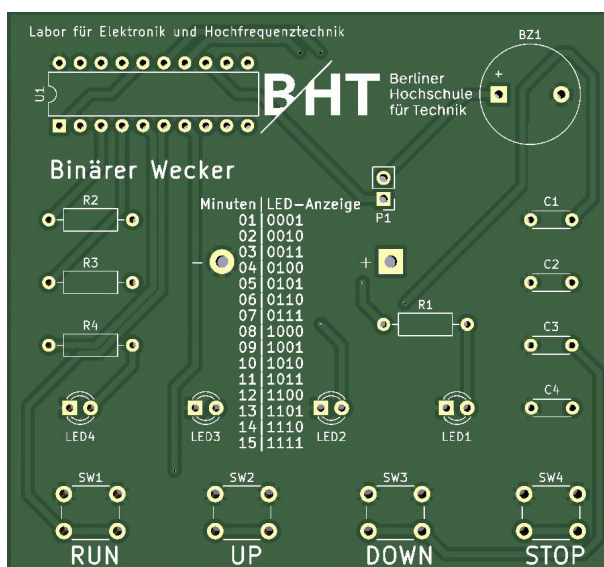


Binär Wecker

1 Konzept des binären Weckers

Statt Dezimalzahlen trifft man in der Programmierung häufig Binärcode an. Über die LEDs wird die verbleibende Zeit des Weckers in diesem Binärcode angezeigt. Ist die LED eingeschaltet, so stellt sie eine 1 da. Bei ausgeschalteter LED ist es eine 0. Die Codierung ist auf der Vorderseite der Platine zu sehen. Insgesamt kann der Wecker auf bis zu 15 Minuten, Binär 1111, eingestellt werden. Durch Tastendruck wird der Timer gestartet, pausiert, eingestellt oder ausgeschaltet. Das Projekt wurde durchgeführt im Rahmen einer Praxisphase in Zusammenarbeit mit Prof. Dr.-Ing. J. Carstens.

Minuten	LED4	LED3	LED2	LED1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1



2 Schaltung

Abbildung 1 zeigt die Schaltung der Platine an. Gesteuert wird der Wecker über den Microcontroller MPS430G2313. Dabei handelt es sich um einen 16-bit Mikrocontroller, welcher mit 1 MHz arbeitet. An ihm sind 4 Taster als Eingänge zur Bedienung angeschlossen. Die 4 LEDs sind als Ausgänge geschaltet. Die Stromquelle ist eine 3 V Knopfzelle. Um die LEDs vor zu hohen Strömen zu schützen ist jeweils ein 300 Ω Widerstand vorgeschaltet. Je höher der Widerstand, desto geringer fallen Helligkeit und Stromverbrauch der LED aus. Die Taster werden durch je einen 100 µF auf Ground gezogen um im Ausgang eine logische 0 zu erzeugen, solange der Taster nicht betätigt wird. Wird der Taster betätigt, gibt der Ausgang eine logische 1 aus. Der Buzzer ist über zusätzliche Pins an den MC angeschlossen. Der Stromkreis zum Buzzer wird erst über die Brücke geschlossen. Dadurch kann der Ton vom Wecker mit Entfernen der Brücke ausgeschaltet werden.

3 Software

Die Software auf dem Microcontroller realisiert einen Zustandsautomaten, zu sehen in Abbildung 2. Nach dem Einlegen der Batterie kann die Zeit mit den mittleren Tastern eingestellt werden. Der linke Taster startet und pausiert den Timer, der rechte Taster schaltet den Wecker in den Ruhemodus. Ein beliebiger Taster kann den Wecker anschließend wieder aufwecken. Sowohl der Pause- als auch der Alarm-modus haben ein Time Out von einer Minute.

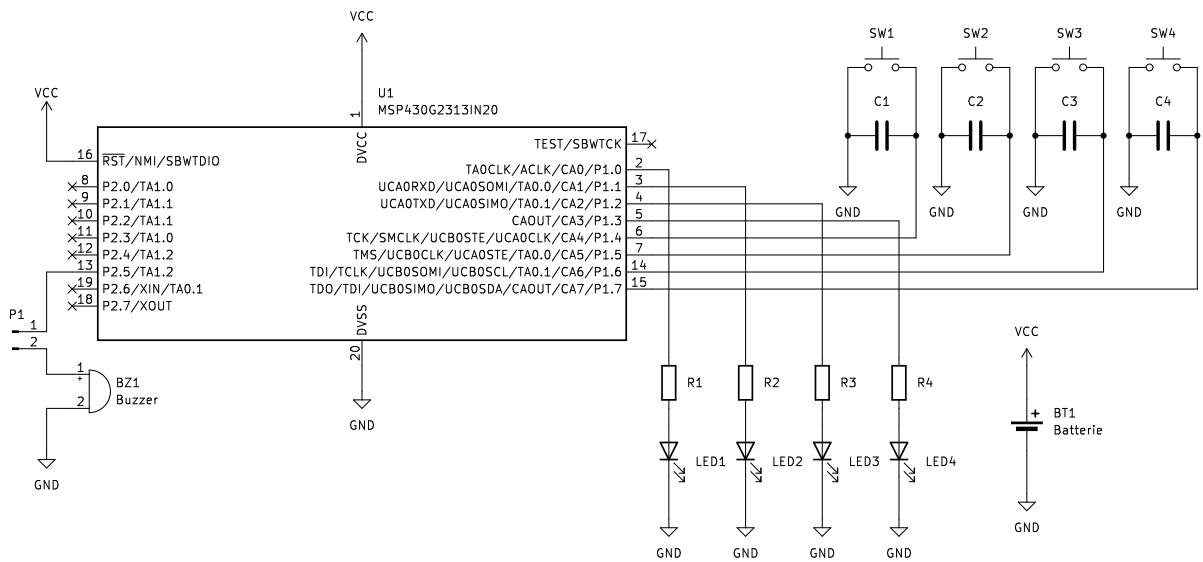


Abbildung 1: Schaltplan des Projekts

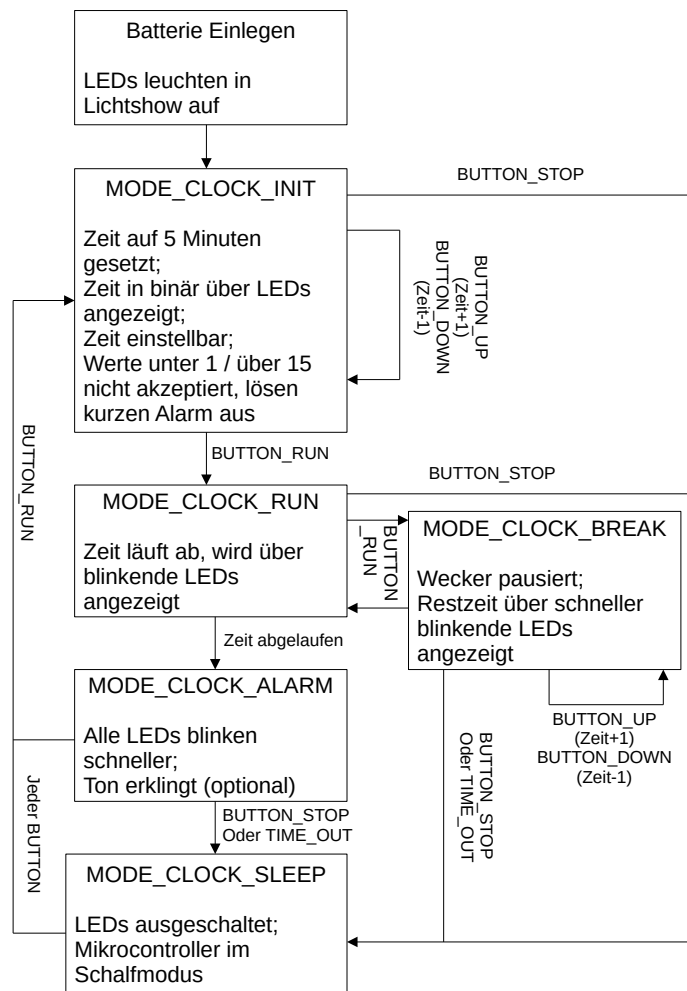


Abbildung 2: Zustandsautomat



Die Software wird ausgeführt auf dem Mikrocontroller MSP430G2313. Programmiert wurde dieser mit dem LaunchPad der Firma Texas Instruments in der Programmiersprache C.

Über die Funktion Lightshow(mode) können verschiedene Abläufe der LED Beleuchtungen wie etwa der Alarm oder die erste Lichtershow programmiert werden. Über IlluminateLED(color, duration) können ausgewählte LEDs für eine angegebene Dauer eingeschaltet werden. Sollen die LEDs blinken kann ebenfalls die Funktion delay_100ms(duration) verwendet werden um die Länge der Pausen zwischen dem Blinken zu bestimmen. Der Buzzer wird aktiviert über P2OUT &= ~SUMM und deaktiviert über P2OUT |= SUMM. Der Time Out wird jeweils im Modus ALARM und BREAK über eine If Abfrage geregelt, welche die Variable t_aus bis zu einem gegebenem Wert hochzählt. Dieser Wert hängt mit der in dem Modus verwendeten Delays zusammen, welche pro Durchlauf verwendet werden.

Soll die Zeit verändert werden, so kann dieser wie folgt berechnet werden:

In BREAK: $t_{aus} = \text{Zeit[s]} * 5$.

In ALARM: $t_{aus} = \text{Zeit[s]} * 0,5$.

Der gesamte Code kann hier gefunden werden: [Website](#).

4 Bestückung der Leiterplatte

Auf der Platine sind die Plätze für die Bauteile aufgedruckt. Wo die Bauteile verlötet werden müssen, kannst du am Schaltplan sehen. LEDs und IC-Sockel müssen in einer bestimmten Ausrichtung angebracht werden. Wenn etwas unklar ist, frage ruhig nach. Dir wird dann dabei geholfen. Bitte lass den Aufkleber auf dem Buzzer, damit andere nicht mit zu viel Lärm belästigt werden.

Tipp: Flache Bauelemente sollten zuerst angebracht werden. Die IC-Fassung und der Mikrocontroller haben jeweils eine Kerbe, welche für die Ausrichtung wichtig ist. Diese ist auch auf dem Umriss auf der Platine zu sehen.

Schaltplan und Platine des Weckers wurden mit KiCad, einer kostenlosen ECAD Software, erstellt.

Inbetriebnahme

Zum Starten der Schaltung gehen wir wie folgt vor:

- Lege die Batterie ein und warte die Lichtershow ab.
- Stelle eine Zeit zwischen 1 und 15 Minuten über die Taster UP und DOWN ein. Die Binärcodierung ist auf der Platine abgebildet. 5 Minuten sind z.B. 0101. Wenn die Begrenzung überschritten wird, ertönt ein kurzer Alarm.
- Starte den Wecker mit dem Taster RUN. Die verbleibenden angefangenen Minuten werden über die blinkenden LEDs angezeigt.
- Willst du den Wecker pausieren oder die Zeit ändern, betätige erneut RUN. Die LEDs blinken schneller und die Zeit kann wie zuvor eingestellt werden. bleibt der Wecker eine Minute im Pause-Modus, schaltet er sich automatisch in den Ruhe-Modus.
- Ist die Zeit abgelaufen, wird der Alarm aktiv. RUN bringt den Wecker wieder in den Anfangszustand, nach einer Minute oder dem drücken des Tasters STOP geht der Wecker in den Ruhe-Modus.
- Jeder Taster bringt dich aus dem Ruhemodus wieder in den Anfangszustand.

Tipp: Nervt dich der Alarmton? Dann zieh einfach die Brücke ab.

5 Nutzdauer

Wenn der Wecker zwei Mal in der Woche für volle 15 Minuten aktiviert ist, hält die Batterie etwa 4.408,8 Stunden oder etwas über 6 Monate. Würde der Wecker im Ruhemodus bleiben, so könnte die Batterie für etwa 1002 Monate oder 83,7 Jahre durch den sehr geringen Verbrauch ausreichen. Der Stromverbrauch des aktiven Weckers hängt von der Helligkeit der LEDs (abhängig vom Vorwiderstand) und der Nutzung des Buzzers ab.

Die Lebenserwartung der Batterie wird wie folgt berechnet:

$$\text{Lebensdauer [h]} = \text{Kapazität [mAh]} / \text{Verbrauch [mA]}$$

Wird der Wecker lange nicht genutzt wird ist es besser, die Batterie zu entfernen. Dadurch wird die Batterie nicht entladen



und hält länger an. In diesem Fall ist dies vielleicht nicht so wichtig, es gilt jedoch nicht nur für diesen Wecker, sondern alle Batterie betriebenen Geräte.

Viel Spaß! Komm gerne auch zur nächsten Langen Nacht der Wissenschaften an der Berliner Hochschule für Technik.

Schöne Grüße vom Studiengang Elektrotechnik

