

Elektronik für Elas' Traumstern

Bild 1: Elas' Traumstern



Einleitung

Um die Lämpchen (LEDs) im Traumstern nicht die ganze Nacht brennen zu lassen und damit die Akkus zu entlasten, sollen die LEDs ca. 16 Minuten nach dem Einschalten automatisch wieder ausgehen - auch ohne dass der Schalter auf AUS gestellt wird.

Erst nach dem Aus- und dann erneutem Einschalten können die LEDs wieder zum Leuchten gebracht werden - auch wieder für ca. 16 Minuten. Die Ausschaltung ist jederzeit durch Betätigung des Schalters möglich.

Die Spannungsversorgung erfolgt durch 4 NiMH-Akkus (Größe Mikro) - also mit ca. 4,8 Volt. Die Beleuchtung besteht aus 12 besonders lichtstarken LEDs, die je mit ca. 4 mA durchflossen werden.

Probleme bei der Realisierung

Die Schaltvorgänge beim Einschalten sind nicht definiert. Kontaktprellungen behindern einen eindeutigen Zeitpunkt der benötigten Impulsflanke. Darüber hinaus ist ein negativer Trigger-Impuls - also eine Impulsflanke von **High** nach **LOW** - zum Anstoß des Timers nötig. Diese kann natürlich erst erzeugt werden, wenn die Versorgungsspannung vorher ansteht.

Lösungsansatz

Dem Timer - bestehend aus einem **Programmable Timer HCF4536** - werden zwei monostabile Multivibratoren (Monoflops) vorgeschaltet. Der erste dient zum Entprellen und der zweite zur Erzeugung eines verzögerten Triggerimpulses. Idealerweise stehen mit dem Dual Type D-Flip-Flop **MC14013B** die benötigten beiden Multivibratoren in einem Gehäuse zur Verfügung.

Das Doppel-D-FlipFlop MC14013B (oder CD4013B oder MOS4013)

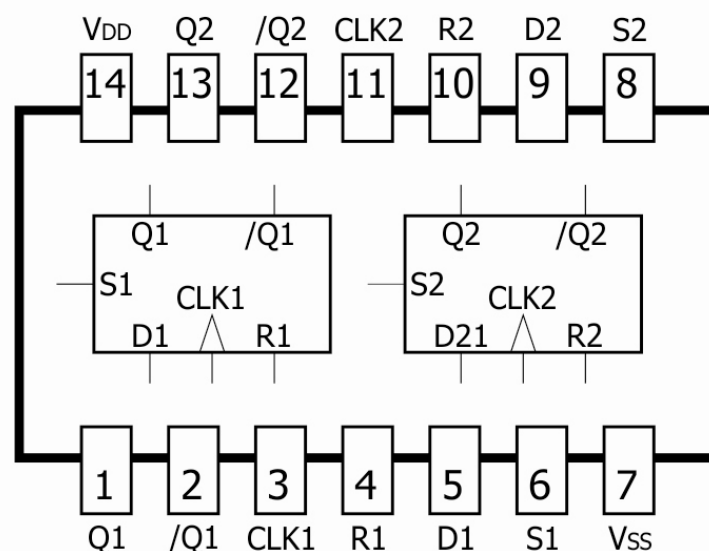


Bild 2: Doppel-FlipFlop - schematisch

Das **B** am Schluss der Typenbezeichnung bedeutet, dass die Ausgänge gepuffert sind. Man erkennt dies, wenn man sich im Datenblatt das Logik-Diagramm ansieht. Es hat zwischen der Schaltung und

den Ausgängen **Q** und **/Q** Inverterstufen. Der Vorteil ist, dass die Belastung von **Q** und **/Q** sich nicht auf die Funktionsfähigkeit der Schaltung auswirkt.

Die Anschlüsse je FlipFlop bedeuten: **D** (Data), **CLK** (Clock, Takt), **S** (Set) und **R** (Reset) sind Eingänge. **Q** ist der Ausgang, der nach einem Taktsignal an **CLK** das an **D** anstehende Logiksignal nicht invertierend übernimmt. Ebenso reagiert **Q** auf eine Triggerung an **R** (**Q** = **LOW**) bzw. an **S** (**Q** = **HIGH**) unabhängig vom Eingang **D** und sofern der Takteingang auf **LOW**-Pegel steht. **/Q** verhält sich invertierend zu **Q**.

Die Wahrheitstabelle des CD4013B und des MC14013B

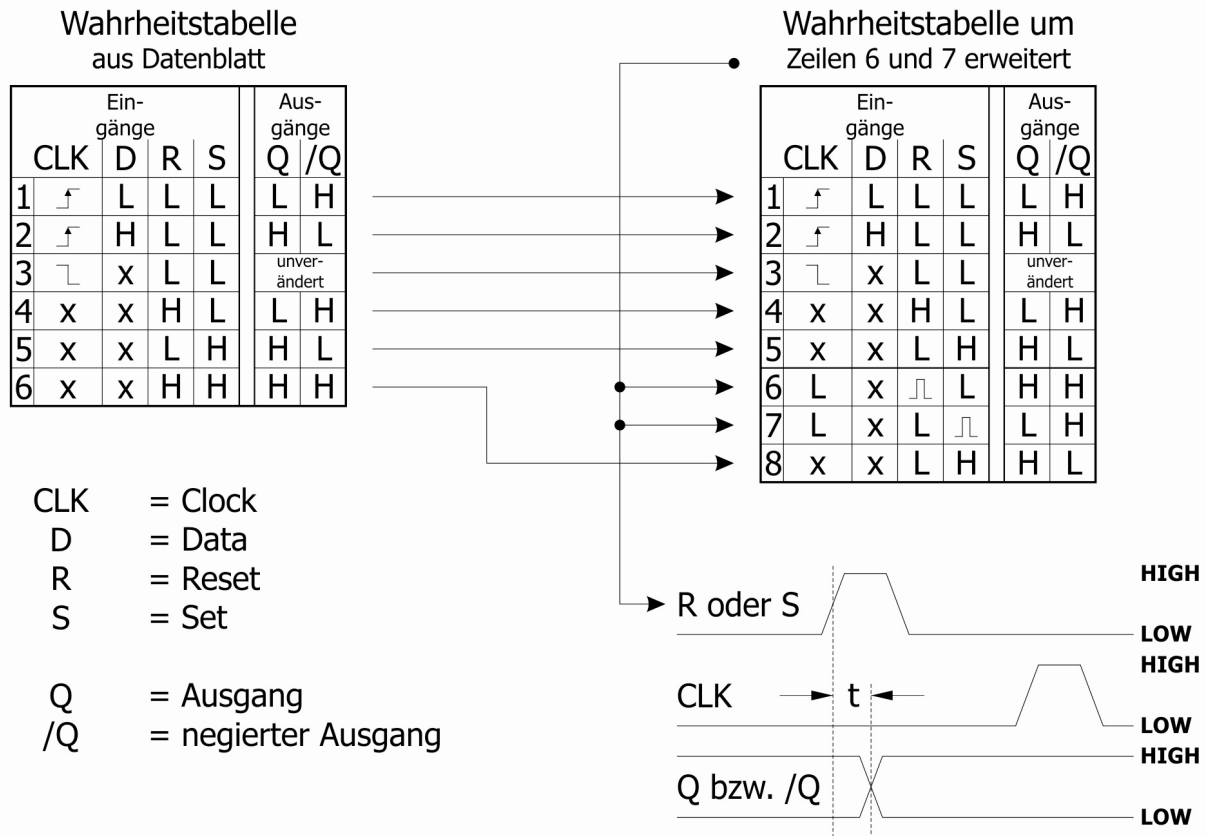


Bild 3: Wahrheitstabelle

Die Tabelle auf der linken Seite ist die Kopie aus dem Datenblatt des **MC14013B**. Die Tabelle auf der rechten Seite ist die erweiterte Wahrheitstabelle, die sich aus dem Impuls-Diagramm des Datenblattes von Motorola ergibt: Die **Figure 2. Dynamic-Signal-Waveforms (Set, Reset, Clock, and Output)** zeigt das Diagramm, welches im Bild 3 unterhalb der erweiterten Wahrheitstabelle vereinfacht wiedergegeben wird.

Erklärung zur erweiterten Wahrheitstabelle von Zeile zu Zeile

Zeilen 1 und 2: Die steigende Flanke des Taktsignals am **CLK**-Eingang überträgt den logischen Zustand am **D**-Eingang zum **Q**-Ausgang (und invertiert zum **/Q**-Ausgang). Mit diesen beiden Zeilen ist leicht erkennbar wie man aus einem D-Flipflop ein T-FlipFlop (Toggle-FlipFlop) realisiert: Man verbindet den Ausgang **/Q** mit dem Dateneingang **D**. Dadurch ändert sich nach jeder steigenden Taktfanke am **CLK**-Eingang der logische Zustand am **D**-Eingang. Die Folge davon ist, dass eine solche Schaltung die Frequenz an **CLK** halbiert. **Q** und **/Q** erzeugen die halbe Frequenz als zeitsymmetrisches Rechtecksignal (Tastverhältnis = 0,5). Das Funktionieren der Frequenzteilung setzt voraus, dass die beiden Eingänge **S** und **R** inaktiv, d.h. auf **LOW** gesetzt sind. Auch dies zeigen die Zeilen 1 und 2.

Zeile 3: Auch hier sind die beiden Eingänge **S** und **R** inaktiv, also ebenfalls auf **LOW** gesetzt. **D** ist mit einem **X** markiert. Dies bedeutet, dass es keine Rolle spielt ob hier ein **LOW**- oder ein **HIGH**-Pegel anliegt - mithin hat **CLK** eine fallenden Flanke keine Auswirkung auf die logischen Zustände der Ausgänge **Q** und **/Q**. Die Ausgänge werden, bezüglich des logischen Pegels, ebenfalls mit **Q** und **/Q** bezeichnet. Es gelten ganz einfach die Logikpegel welche gerade anliegen (no change!).

Zeile 4: Eingang **R** ist aktiv (**HIGH**) und **S** soll inaktiv (**LOW**) sein. **CLK** und **D** sind mit **X** ausgewiesen und das bedeutet, dass es egal ist, ob an **CLK** eine aktive steigende Flanke erfolgt und/oder an **D** ein **HIGH**- oder **LOW**-Pegel anliegt, es passiert einfach nichts. Der **LOW**-Pegel an **S** drückt dem D-Flipflop seinen Stempel auf, in dem dieser bestimmt, dass **Q** solange auf **LOW** bleibt, wie **R** auf **HIGH** liegt.

Zeile 5: Eingang **S** ist aktiv (**HIGH**) und **R** soll inaktiv (**LOW**) sein. **CLK** und **D** sind mit **X** ausgewiesen und das bedeutet, dass es egal ist, ob an **CLK** eine aktive steigende Flanke erfolgt und/oder an **D** ein **HIGH**- oder **LOW**-Pegel anliegt, es passiert ebenfalls nichts. Der **HIGH**-Pegel an **S** drückt dem D-Flipflop seinen Stempel auf, in dem dieser bestimmt, dass **Q** solange auf **HIGH** bleibt, wie **S** auf **HIGH** liegt.

Zeilen 6 und 7: Diese beiden Zeilen vervollständigen die beiden Zeilen 4 und 5, die im gewissen Sinne unvollständig sind. An den Eingängen **R** und **S** sieht man jetzt jeweils Impulse anstelle von Logikpegeln. Voraussetzung für die folgenden Funktionen ist, dass **CLK** auf **LOW** liegt.

Zeile 6: Ein ausreichender*) **HIGH**-Impuls an **R** setzt bei gleichzeitigem **LOW**-Pegel an **S** das FlipFlop auf inaktiv: **Q** wird **LOW** und **/Q** wird **HIGH**.

Zeile 7: Ein ausreichender*) **HIGH**-Impuls an **S** setzt bei gleichzeitigem **LOW**-Pegel an **R** das FlipFlop auf aktiv: **Q** wird **HIGH** und **/Q** wird **LOW**.

Wenn man das **LOGIC-DIAGRAM** im Datenblatt des **MC14013B** betrachtet, dann fällt auf, dass während des **LOW**-Pegels auf **CLK** je zwei NOR-Gatter mittels so genannter Transmission-Gates (MOS-Schalter) kreuzweise mitgekoppelt sind. Während dieses Zustandes arbeitet das D-FlipFlop mit den Eingängen **R** (Reset) und **S** (Set) als RS-FlipFlop. Im Impuls-Diagramm unterhalb der erweiterten Wahrheitstabelle sieht man, dass bei der RS-FlipFlop-Funktion der Eingang **CLK** auf **LOW** liegt.

*) Der Chip lässt auch langsame analoge Spannungsänderungen an den pegelsensitiven Eingängen **S** (Set) oder **R** (Reset) zu - im Gegensatz zu flankensensitiven Eingängen wie z.B. **CLK**, bei denen stets die jeweilige Impuls-Flanke eine vorgegebene Steilheit aufweisen muss!

Zeile 8: Wenn **R** und **S** auf **HIGH** sind, dann sind auch **Q** und **/Q** auf **HIGH** gesetzt sind. Es spielt dabei, wie in den Zeilen 4 und 5 keine Rolle was dabei an **CLK** und/oder **D** geschieht.

Die Eingangsschaltung

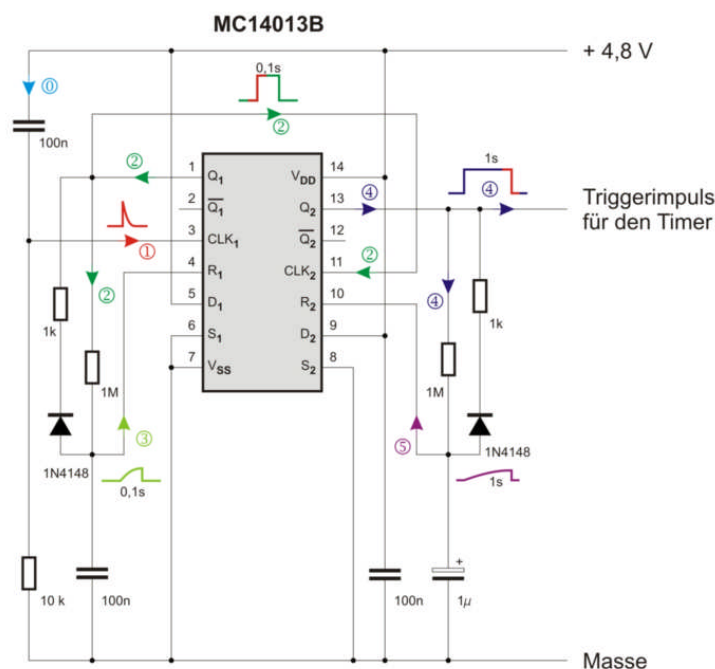


Bild 4: Eingangsschaltung

In der Eingangsschaltung zur Ausschaltverzögerung werden die FlipFlops als R-FlipFlops geschaltet, so dass sich die Wahrheitstabelle auf die Zeilen 2 und 6 beschränkt (**D** ist fest auf **HIGH** und **S** ist fest auf **LOW** geschaltet):

Zeilen-Nr.	CLK ₁	D ₁	R ₁	S ₁	Q ₁
2	①↑	H	L	L	②H
6	L	H	③↗	L	②L
Zeilen-Nr.	CLK ₂	D ₂	R ₂	S ₂	Q ₂
2	②↑	H	L	L	④H
6	L	H	⑤↗	L	④L

Bild 5: Wahrheitstabelle 2

1. Schritt - Schalter entprellen (1. Stufe)

Beim Einschalten (① Anlegen der Spannung von +4,8 V über einen Ein-/Aus-Schalter) wird durch das Differenzierglied (100nF/10kΩ) - ideal betrachtet - eine positive Flanke ①↑ auf **CLK₁** gegeben. Tatsächlich wirken alle **ausreichenden** positiven Spannungswechsel der Schalter-Prellungen als positive Flanken. Zeitgleich stehen an **D₁** die selben positiven Spannungswechsel an, so dass die erste wirk-same positive Flanke ①↑ aus den differenzierten Prellungen an **CLK₁**, den gleichzeitigen **HIGH**-Pegel aus den Prellungen von **D₁** nach **Q₁** durchgeschaltet. Das heißt, **Q₁** wird bereits nach dem ersten wirk-samen Impuls auf **HIGH** (②H) geschaltet und bleibt in diesem Zustand, da weitere Flanken an **CLK₁** unwirksam sind.

2. Schritt - Schalter entprellen (2- Stufe)

Anstelle einer Reset-Taste und Pulldown-Widerstand sieht man hier ein RCD-Netzwerk, bestehend aus den Widerständen 1MΩ, 1kΩ, einer Diode und einem Kondensator von 100nF zwischen **Q₁**-Ausgang und **R₁**-Eingang. Dieses Netzwerk erzeugt eine R-FlipFlop-Funktion.

Durch die Zeitkonstante RC (1MΩ * 100nF) wird die Impulsdauer von ③↗ definiert. Mit der steigenden Flanke ①↑ des Taktimpulses an **CLK₁** wird das FlipFlop gesetzt und der Ausgang **Q₁** geht auf **HIGH**. Nun beginnt der Kondensator 100nF sich über den Widerstand 1MΩ aufzuladen. Durch die zunehmende Ladung vom Kondensator steigt die Spannung an **R₁** (dargestellt durch den schrägen Pfeil ↗). Wenn diese etwa die halbe Betriebsspannung (+4,8V) übersteigt, dann wird das FlipFlop zurückge-setzt. **Q₁** geht auf **LOW**, der Ausgangsimpuls von etwa 0,1s Dauer ist beendet und der Kondensator 100nF entlädt sich sehr schnell über die Diode und dem Widerstand 1kΩ. Damit ist die Prellzeit in der ersten Hürde überwunden.

3. Schritt - Einleitung des Trigger-Impulses

Q₁ geht - wie in 1. Schritt beschrieben - bereits nach dem ersten wirksamen Impuls auf **HIGH**. Diese positive Flanke ②H wird auch zum Eingang **CLK₂** des 2. FlipFlops durchgeschaltet und leitet damit, da **D₂** fest mit dem Schalter zur Akkuspannung verbunden ist, den **HIGH**-Pegel an **Q₂** weiter: ④H.

4. Schritt - Trigger-Impuls

Über das 2. RCD-Netzwerk, bestehend aus den Widerständen 1MΩ, 1kΩ, einer Diode und einem Kondensator von 1µF zwischen **Q₂**-Ausgang und **R₂**-Eingang, wird die Verzögerung zum Triggern des Timers erzeugt. Das Signal ⑤↗ entsteht: Durch die zunehmende Ladung vom Kondensator steigt die Spannung an **R₂**. Übersteigt diese etwa die halbe Betriebsspannung, wird das 2. FlipFlop zurückge-setzt. **Q₂** geht auf **LOW**, was dem Triggersignal ④L entspricht. Der Ausgangsimpuls von etwa 1s Dauer ist beendet und der Kondensator 1µF entlädt sich sehr schnell über die Diode und dem Wider-stand 1kΩ.

Die Timerschaltung

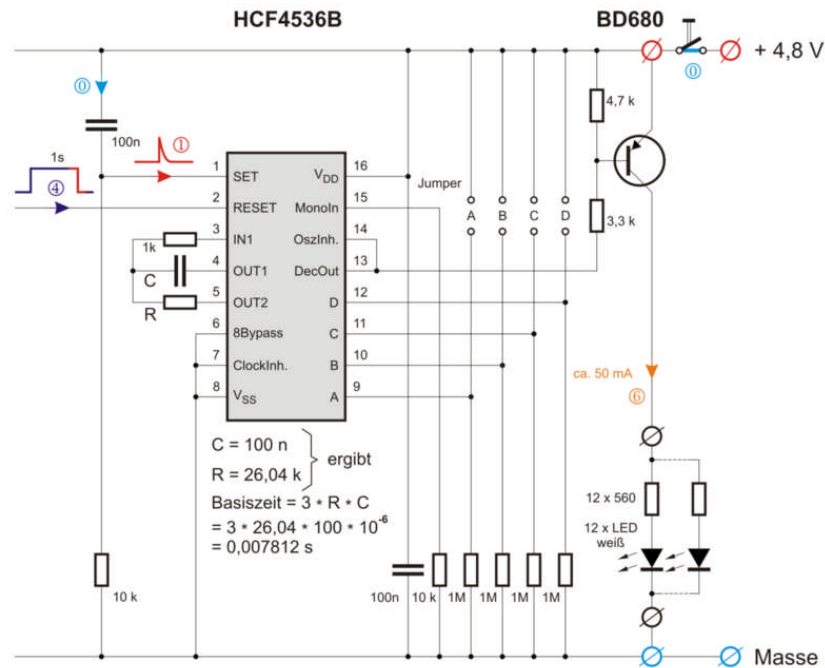


Bild 6: Timerschaltung

Beschreibung

Der **HCF4536B** (auch CD4536, MOS4536, MC14536) ist ein schon etwas betagter Schaltkreis, aber mit geradezu wundervollen Eigenschaften. Außerdem ist er spottbillig (ca. 30 Cent). Es lässt sich damit ein Timer für Millisekunden oder auch für 50 Tage Laufzeit oder länger aufbauen. Der Ruhestrom ist äußerst gering und kann auch bei Batteriebetrieb durch die sehr hohen Pulldown-Widerstände an **A** bis **D** von 1MΩ vernachlässigt werden. Er ist abhängig von den Schalterstellungen und beträgt bei 5 V max. 25 µA wenn **A** bis **D** geschlossen sind (geöffnet entsprechend weniger, bei höherer Spannung mehr). Bei unterschiedlichen Spannungen zwischen 3 und 20 Volt verhält er sich relativ konstant.

Etwas irritierend ist der Umstand, dass der **Eingangsimpuls** an Pin 2 (**RESET**) liegt, und umgekehrt die **Rückstellung** an Pin 1 (**SET**). Die Rückstellung wird in der verwendeten Schaltung als solche nicht benutzt. **SET** wird nach dem Einschaltvorgang fest auf **LOW** gesetzt. Statisch betrachtet setzt damit ein **HIGH** am **RESET** den Ausgang **DecOut/OszInh.** (Pin 13/14) auf **LOW**, was den Transistor leitend macht und die LEDs zum Leuchten bringt. Da der Timer re-triggerbar ist, was hier ebenfalls nicht zur Anwendung kommt, beginnt das Zeit-Intervall logischerweise erst mit dem **HIGH-LOW**-Übergang am **RESET**.

Die **Wahrheits-Tabelle** der allgemeinen Schaltungs-Konfiguration als **Zeit-Intervall-Schalter** mit RC-Oszillator stellt sich wie folgt dar:

SET Pin 1	RESET Pin 2	DecOut/OszInh. Pin 13/Pin 14	PNP- Transistor
L	L	H	gesperrt
L	H	L	leitend
L			
	H		
H	L	H	gesperrt
H			
H			

Bild 7: Trigger-Signale an SET bzw. RESET:

	Negativer Sprung von HIGH nach LOW (Trigger-Impuls)
	Kurzer positiver Impuls (Dauer < Zeit-Intervall)
	Langer positiver Impuls (Dauer > Zeit-Intervall)

Ausgangssignale an DecOut/OszInh.:

	Negativer Impuls mit der Länge des eingestellten Zeit-Intervalls plus des Vorlaufs, bedingt durch die vorherigen Pegel an RES und RESET
	Kurzer negativer Impuls (Dauer < Zeit-Intervall)
	Negativer Impuls (Dauer = Zeit-Intervall)

Wirkung (z.B. am PNP-Transistor):

	Transistor für die Dauer des Impulses (< Zeit-Intervall) leitend
	Transistor für die Dauer des Intervalls leitend
	Transistor für die Dauer des eingestellten Zeit-Intervalls plus des Vorlaufs, bedingt durch die vorherigen Pegel an RES und RESET , leitend

Bild 7: Trigger-Signale an SET bzw. RESET

In der verwendeten Schaltung wird beim Einschalten der Eingang **RESET** (Pin 2) der Timerschaltung unmittelbar mit dem Impuls ④ (**HIGH**) belegt. Das hat zur Folge, dass der Ausgang **DecOut/OszInh.** (Pin 13/14) sofort auf **LOW** geschaltet wird. Die Trigger-Impuls-Flanke ④L von der Eingangsschaltung kommt also erst mit einer Verzögerung von ca. 1s an, so dass - genau genommen - diese Vorlauf-Sekunde dem Zeit-Intervall hinzugefügt werden muss. Das ergibt die Timer-Laufzeit.

Einstellung des Zeit-Intervalls

Der Timer muss für den Bereich des Zeit-Intervalls codiert werden. Es werden dafür 4 Jumper für die einzelnen BCD-Eingänge **A** bis **D** vorgesehen, d.h. die IC-Eingänge **A**, **B**, **C** und **D** werden bei geschlossenem Kontakt auf **HIGH** gesetzt. Die Codierung wird nach Tabelle vorgenommen (die 16 verschiedenen Codes kann man sich wohl schlecht merken).

Am Ende des Zeit-Intervalls springt der Ausgang am **DecOut/OszInh.** (Pin 13/14) auf **HIGH** und sperrt den Transistor. Die LEDs erlöschen.

Dreh-Schalter	Codierung					Zeit Berechnung	Basiszeit $t_B = 0,0078125$ Sekunden	
							Zeit-Intervall	
	Stellung	A	B	C	D		8- Byp.	Sekunden
0	0	0	0	0	0	Basiszeit * $2^8 = t_B * 256$	"Null-Zeit" 2,0	00:00:02
1	1	0	0	0	0	Basiszeit * $2^9 = t_B * 512$	4,0	00:00:04
2	0	1	0	0	0	Basiszeit * $2^{10} = t_B * 1.024$	8,0	00:00:08
3	1	1	0	0	0	Basiszeit * $2^{11} = t_B * 2.048$	16,0	00:00:16
4	0	0	1	0	0	Basiszeit * $2^{12} = t_B * 4.096$	32,0	00:00:32
5	1	0	1	0	0	Basiszeit * $2^{13} = t_B * 8.192$	64,0	00:01:04
6	0	1	1	0	0	Basiszeit * $2^{14} = t_B * 16.384$	128,0	00:02:08
7	1	1	1	0	0	Basiszeit * $2^{15} = t_B * 32.768$	256,0	00:04:16
8	0	0	0	1	0	Basiszeit * $2^{16} = t_B * 65.536$	512,0	00:08:32
9	1	0	0	1	0	Basiszeit * $2^{17} = t_B * 131.072$	1.024,0	00:17:04
10	0	1	0	1	0	Basiszeit * $2^{18} = t_B * 262.144$	2.048,0	00:34:08

11	1	1	0	1	0	Basiszeit * $2^{19} = t_b * 524.288$	4.096,0	01:08:16
12	0	0	1	1	0	Basiszeit * $2^{20} = t_b * 1.048.576$	8.192,0	02:16:32
13	1	0	1	1	0	Basiszeit * $2^{21} = t_b * 2.097.152$	16.384,1	04:33:04
14	0	1	1	1	0	Basiszeit * $2^{22} = t_b * 4.194.304$	32.768,1	09:06:08
15	1	1	1	1	0	Basiszeit * $2^{23} = t_b * 8.388.608$	65.536,2	18:12:16
0	0	0	0	0	1	Basiszeit * $2^0 = t_b * 1$	0,00781	00:00:00
1	1	0	0	0	1	Basiszeit * $2^1 = t_b * 2$	0,01563	00:00:00
2	0	1	0	0	1	Basiszeit * $2^2 = t_b * 4$	0,03125	00:00:00
3	1	1	0	0	1	Basiszeit * $2^3 = t_b * 8$	0,06250	00:00:00
4	0	0	1	0	1	Basiszeit * $2^4 = t_b * 16$	0,12500	00:00:00
5	1	0	1	0	1	Basiszeit * $2^5 = t_b * 32$	0,25000	00:00:00
6	0	1	1	0	1	Basiszeit * $2^6 = t_b * 64$	0,50000	00:00:01
7	1	1	1	0	1	Basiszeit * $2^7 = t_b * 128$	1,00000	00:00:01
8	0	0	0	1	1	Basiszeit * $2^8 = t_b * 256$	2,00001	00:00:02
9	1	0	0	1	1	Basiszeit * $2^9 = t_b * 512$	4,00001	00:00:04
10	0	1	0	1	1	Basiszeit * $2^{10} = t_b * 1.024$	8,00003	00:00:08
11	1	1	0	1	1	Basiszeit * $2^{11} = t_b * 2.048$	16,00006	00:00:16
12	0	0	1	1	1	Basiszeit * $2^{12} = t_b * 4.096$	32,00011	00:00:32
13	1	0	1	1	1	Basiszeit * $2^{13} = t_b * 8.192$	64,00023	00:01:04
14	0	1	1	1	1	Basiszeit * $2^{14} = t_b * 16.384$	128,00046	00:02:08
15	1	1	1	1	1	Basiszeit * $2^{15} = t_b * 32.768$	256,00092	00:04:16

Bild 8: Tabelle für C = 100 nF und R = 26,042 kΩ

Nebenbei: Eine universelle Schaltungsvariante

Mit der nachfolgend beschriebenen universellen Schaltungsvariante lassen sich prinzipiell Timer-Laufzeiten zwischen Sekundenbruchteilen bis zu Tagen oder Monaten erzeugen. Das jeweilige Zeit-Intervall wird durch einen Code bestimmt, der mit einem DIP-Schalter oder Codier-Drehschalter realisiert werden kann. Vorweg muß lediglich die **Basiszeit** t_b in Sekunden einmalig bestimmt und justiert werden. Diese Zeit ist sehr kurz und basiert auf einer einstellbaren Frequenz (Clock) in Form von Rechteckschwingungen. Um sehr große Zeit-Intervalle zu erzeugen, wird mit Hilfe eines Dualzählers die Basiszeit vervielfacht. Das so erzeugte Intervall ist also ein ganzes Vielfaches (Faktor 2^0 bis 2^{23}) der Basiszeit. Die "Null-Zeit" (alle Jumper sind offen) wird in dieser universellen Schaltungsvariante per Stopp-Uhr mittels eines Potentiometers **P** fein eingestellt. Danach stimmen auch alle anderen Zeiten der anderen Codes automatisch mit der Tabelle überein.

Beachte: Das Zeit-Intervall läuft erst mit der negativen Impulsflanke (Triggerimpuls) am **RESET**-Eingang des **HCF4536B** los. Genau genommen muss der **RESET**-Eingang aber erst auf **HIGH** gehen, um dann die negative Impulsflanke erzeugen zu können. Daraus resultiert, dass die Zeit zwischen **HIGH** und **LOW** dem Zeit-Intervall hinzugefügt werden muss, um die echte Timer-Laufzeit zu erhalten. In der Praxis wird der Triggerimpuls durch eine Ein-Taste realisiert, so dass der Vorlauf vernachlässigt werden kann. Auch Prellungen dieser Taste haben keine Bedeutung - die letzte wirksame **HIGH/LOW**-Flanke triggert das Zeit-Intervall.

Die Ermittlung der Basiszeit

Die kürzeste einstellbare Zeit ist die **Basiszeit** t_b und errechnet sich aus der Oszillatorfrequenz. Diese wird durch die Beschaltung des Bausteins bestimmt. Da hier ein Impulszähler verwendet wird, sind die frequenzbestimmenden Bauteile **C** und **R** Basis der Berechnung. Wenn **R** aus einem Festwiderstand **R₁** und einem Potentiometer **P** besteht, kann man die Basis-Frequenz entsprechend der Basiszeit mit dem Potentiometer in Grenzen fein einstellen.

Es soll hier eine Basiszeit von **0,0078125 Sekunden** festgelegt werden, damit man mit den entsprechenden Codier-Einstellungen zu den angegebenen Werten der o.a. Tabelle kommt.

Nach der Formel für die Zeitkonstante t ergibt sich die bauteileabhängige Basiszeit in Sekunden:

$$t_b = 3 R C = 3 C (R_1 + P)$$

R_1 und P in Ohm sowie C in Farad

C wird auf 100 nF und R_1 auf 22 kΩ festgelegt. Folglich variiert man die Zeitkonstante nur noch durch Einstellung an P :

$$P = (t_b / 3C) - R_1$$

t_b in Sekunden, C in Farad sowie P und R_1 in Ohm

$$P = [0,0078125 / (3 * 100 * 10^{-9})] - (22 * 10^3) \quad \text{in Ohm}$$

$$P = 4.041,667 \, \Omega$$

Dieser Wert, plus/minus einer Korrektur, um die Streuung des Kapazitätswertes von C auszugleichen, wird an dem Poti P mit 10 kΩ eingestellt. Beim Feineinstellen muss P rechts herum höher oder links herum tiefer nachreguliert werden. Überprüft wird das Ergebnis durch Stoppen der **Timer-Laufzeit** unter Vernachlässigung des zeitlichen Vorlaufs zwischen **HIGH** und **LOW** am **RESET**-Eingang. Schaltet man den mit **8Bypass** bezeichneten Pin 6 auf **LOW**, ergibt die Stellung **0** der oberen Tabelle das 256-fache der Basiszeit von 0,0078125 Sekunden also genau 2 Sekunden. Da dieser Wert noch in der Größenordnung des Vorlaufs liegt und relativ kurz ist, kann zur genaueren Einstellung ein Vielfaches dieser Zeit gewählt und diese dann mit der Stopp-Uhr an P abgeglichen werden (z.B. Schalterstellung 5 mit 64 Sekunden). Je höher die Schaltstellung, umso exakter lässt sich die Einstellung der Basiszeit durchführen.

Die "krumme" Basiszeit von 0,0078125 Sekunden ist notwendig, weil diese bei der Einstellung des ersten Codes der Tabelle, d.h. bei geöffneten Schaltern **A** bis **D** (**LOW**-Pegel an allen Impulszähler-Eingängen) exakt ein Zeit-Intervall von 2 Sekunden ergibt. Beim Dreh-Codierschalter entspricht das auch der Grund-Schalterstellung. Bei jeder folgenden Schalterstellung verdoppelt sich das Zeit-Intervall, so dass insgesamt ein Bereich von 2 Sekunden (Stellung 0) bis über 18 Stunden (Stellung 15) abgedeckt wird. Mit anderen Werten für C und R lassen sich auch gänzlich andere Bereiche erzielen, kürzere oder erheblich längere.

Für die Einschalt-Elektronik für Elas' Traumstern wurde lediglich die Zeitkonstante per $C = 100 \text{ nF}$ und $R = 27 \text{ k}\Omega$ festgelegt, was in etwa Timer-Laufzeiten gem. der o.a. Tabelle ergibt. Es wurden Jumper bei den IC-Eingängen **A** und **D** gesteckt, so dass der Traumstern ca. 16 Minuten nach dem Einschalten sein Leuchten einstellt.

Realisierte Timerschaltung - Platine: Bestückung und Layout

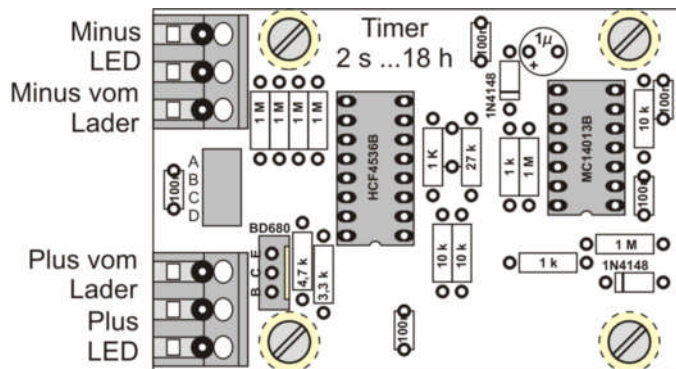


Bild 9: Bestückungsseite

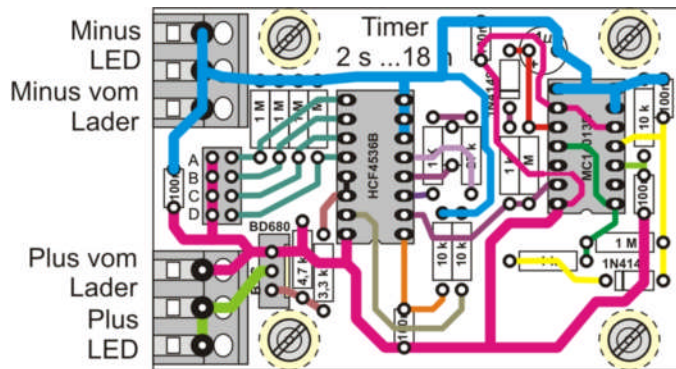


Bild 10: Leiterbahnen-Kontrolle

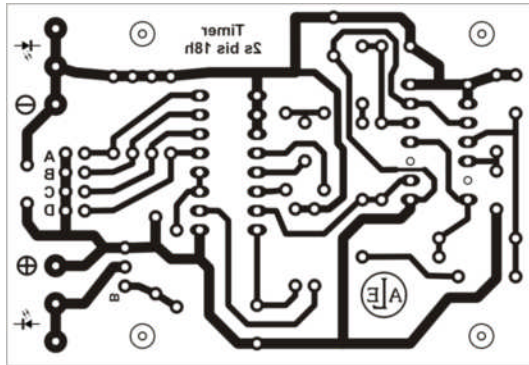


Bild 11: Layout 48 mm x 70 mm

NiMH-Akku-Lader

Ein häufiger Akku-Wechsel auf Grund der geringen Ladekapazität der 4 Mikro-Zellen (AAA-Akkus mit je 800 mAh) wird durch den festen Einbau der Zellen einschließlich einer Ladeschaltung vermieden. Die Aufladung geschieht über einen fertigen Netzteiladapter, der 8 bis 12 V und einen Strom bis zu 0,5 A liefert.

Vorsicht: Bei zu hoher Spannung des Netzteiladapters wird der Transistor **2N6109** sehr warm. Auch wenn er laut Datenblatt bis zu 150 Grad aushält, sollte man wegen der bedrängten und geschlossenen Bauweise diesen Zustand vermeiden. Ideal ist ein Netzteiladapter, der bei einer Spannung von 9 Volt genau 250 mA fließen lässt.

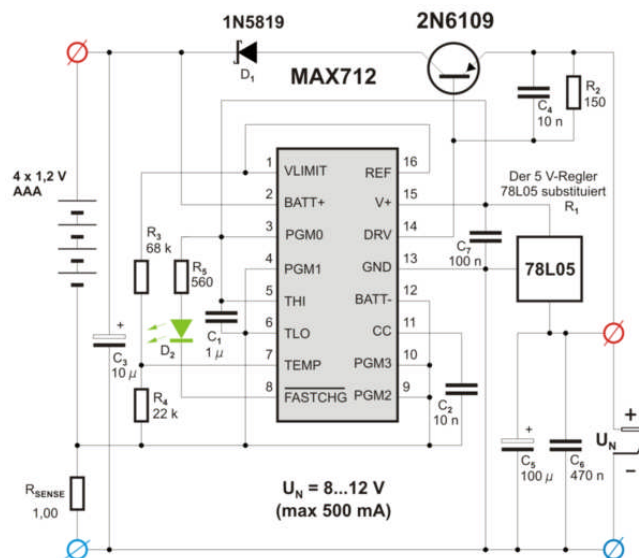


Bild 12: Ladeschaltung

Die Ladeschaltung basiert auf dem Ladecontroller **MAX712** von Maxim. Dieser Baustein stellt eine Konstantstromquelle mit Spannungsbegrenzung und automatischer Abschaltung zur Verfügung. Über die Pins **PGM0** bis **PGM3** lässt sich der Controller "programmieren". Die Schaltung wird für den Betrieb an vier in Serie geschaltete 1,2-V-NiMH-Zellen eingestellt (**PGM0** an **V+** und **PGM1** an **BATT-**). Die Akku-Zellen des Traumsterns befinden sich zwischen der Timerschaltung und der hier dargestellten Ladeschaltung. Zu beachten ist, dass der Minuspol der Akku-Batterie nicht direkt an Masse liegt,

sondern es wurde zur Erfassung der Ladestromstärke der Shunt R_{sense} zwischengeschaltet. Der Transistor **2N6109**, den der Ladecontroller über **DRV** ansteuert, bestimmt den Ladestrom durch **D₁**, die Akku-Batterie und durch R_{sense} . Der **MAX712** misst ständig die am Shunt-Widerstand R_{sense} abfallende Spannung und steuert den Transistor so, dass diese immer 250 mV beträgt. Bei einem Widerstand von 1,00 Ohm fließt ein Ladestrom von genau 250 mA. Bei anderen Werten für R_{sense} ändert sich der Ladestrom entsprechend. Ein Strom von 250 mA eignet sich gut für Zellen mit einer Kapazität von 800 mAh (das entspricht etwa C/3) und erlaubt eine Komplettladung aller Zellen in weniger als vier Stunden.

Von diesem Wert ausgehend setzt die Schaltung über die Programmierpins **PGM2** und **PGM3** den Timeout auf 4,4 Stunden (**PGM2** an **BATT-** und **PGM3** an **BATT-**). Auch wenn der MAX712 nach dieser Zeit keine vollen Akkus erkannt haben sollte, bricht er den Ladevorgang ab und schaltet auf Erhaltungsladung. Diese Maßnahme verhindert ein Überladen der Akkus. Im Normalfall stellt der Ladecontroller aber schon vorher fest, dass die Batterien voll sind. Der MAX712 misst dazu permanent an den Pins **BATT+** und **BATT-** die über den Akku anstehende Spannung. Sie steigt während des Ladens stetig an. Sind die Akkus voll, dann bleibt sie eine Zeitlang konstant, um danach wieder leicht abzufallen. Das Absinken der Spannung am Ende des Ladezyklus ist bei NiMH-Akkus weniger ausgeprägt als bei NiCd-Zellen, so dass der für NiMH-Batterien entworfene MAX712 am Gleichbleiben der Spannung das Ende des Ladevorganges erkennt.

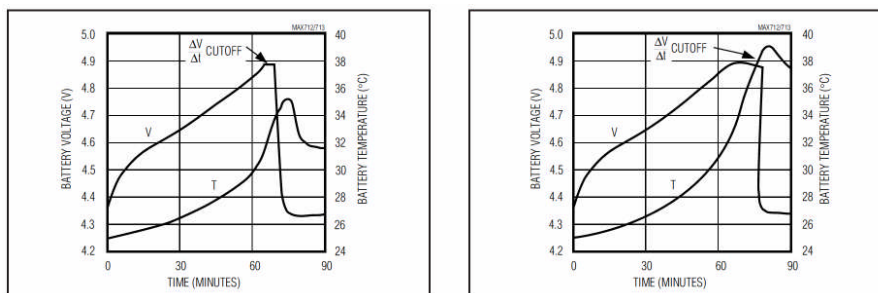


Bild 13: NiMH-Ladekurven von MAX712 (links) und MAX713 (rechts)
am Beispiel von 3 AA-Akkus je 1000 mAh; Timeout bei 90 min
Beim MAX712 wird der CUTOFF bereits im Scheitelpunkt der Ladekurve ausgelöst

Alternativ lässt sich die Schaltung also auch mit dem pinkompatiblen MAX713 betreiben. Dieser Chip reagiert auf den NiCd-typischen Spannungsknick und sollte für diese Akkusorte verwendet werden. Auf Grund ihrer deutlich größeren Kapazität und vor allem wegen des kaum vorhandenen Memory-Effekts werden NiMH-Akkus eingesetzt.

Die hier vorgestellte Schaltung setzt den Laderegler über die beiden Pins **PGM0** und **PGM1** fest auf die Ladung von vier Zellen. Zusätzlich begrenzt der Chip die maximale Ladespannung (**BATT+** gleich **VLIMIT** mal der verwendeten Zellenzahl). Da **VLIMIT** mit dem Pin **REF** - hier liegt eine Referenzspannung von 2 V an - verbunden ist, ergibt sich für die Ladespannung ein Wert von maximal 8 V.

Die grüne LED **D₂** zeigt den Ladevorgang während der "Schnellladung" an. Pin **FASTCHG** schaltet sie beim Übergang zur Erhaltungsladung wieder aus.

Realisierter Akkulader - Platine: Bestückung und Layout

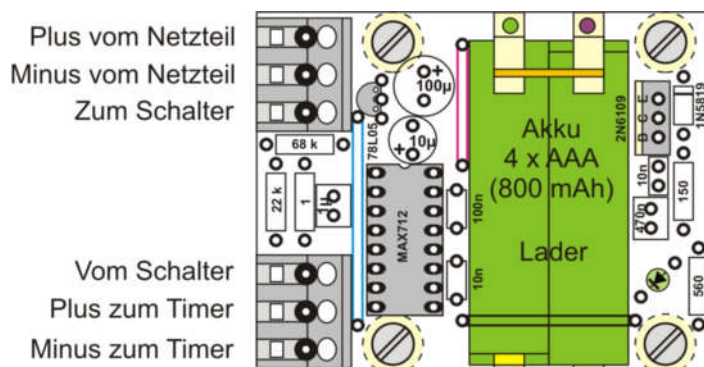


Bild 14: Bestückungsseite

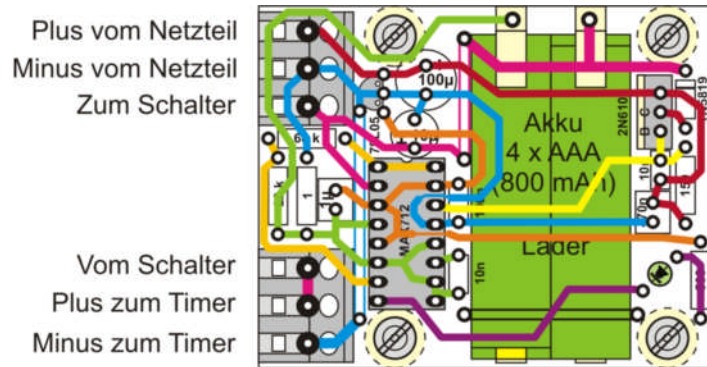


Bild 15: Leiterbahnen-Kontrolle

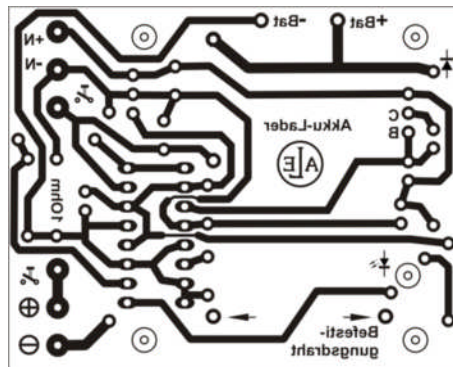


Bild 16: Layout 48 mm x 60 mm

