

Bits of Time



Das Projekt wurde eingestellt. Bausätze sind nicht mehr verfügbar. Diese Seiten dienen nur als Referenz.

Die gute alte Sanduhr in moderner, digitaler Ausführung

Wie das klassische Original wird sie durch Umdrehen gestartet. Eine Sandkorn-Simulation visualisiert auf originelle Art das Verstreichen der Zeit. Natürlich kann man die Bits-of-Time auch zwischendrin umdrehen. Nach Ablauf der Zeit schlägt sie optisch Alarm. Je nach Programmierung können auch weitere Aktionen ausgelöst werden (z. B. den Teebeutel aus der Tasse ziehen). Im Gegensatz zu der traditionellen Variante lässt sich bei der Sanduhr 2.0 die Zeit in 15-Sekunden-Schritten einstellen.



Video

Aufbau

Man benötigt die üblichen Werkzeuge zum Lötten von Elektronikschaltungen. Der Mikrocontroller ist ein SMD-Bauteil im SO14-Gehäuse. Entsprechende Erfahrung im Lötten von SMD-Bauteilen sollte vorhanden sein. Zum Auflöten des Controllers empfiehlt sich die Verwendung von dünnem Lötzinn (0,5 mm) und einer Pinzette.

Für die Programmierung des Mikrocontrollers wird außerdem ein passendes Programmiergerät (z. B. [ARVisp MKII](#), [USBasp](#) o. ä.) benötigt. Mit dem Flashen eines Mikrocontrollers sollte man vertraut sein. Hinweise hierzu, finden sich beim Hersteller des jeweiligen Programmiergeräts.

Material



- 2 [PixBlocks](#)
- 1 Neigungsschalter (SW200D oder SW520D)
- 1 Mikrocontroller ATtiny84A (SMD)
- 1 Wannenstecker mit 2 x 3 Pins
- 3 Drucktaster 6 x 6 mm
- ca. 8 cm Flachbandkabel 5-adrig (nicht im Teilesatz enthalten)

Es gibt einen Teilesatz, der außer dem Flachbandkabel alle aufgeführten Teile enthält.

Bauanleitung



1. Auf dem ersten PixBlock werden die Löt pads für den Prozessor mit Entlötlitze von überschüssigem Lötzinn befreit. Danach wird der Prozessor aufgelötet.
2. Anschließend werden die drei Taster und der 6-polige Wannenstecker bestückt. Der Wannenstecker ist der ISP-Anschluss zur Programmierung des Prozessors. Beim Wannenstecker auf die richtige Orientierung achten.
3. Der Neigungsschalter wird an CON5 zwischen GND (Pin 1) und PA3 (Pin 6) geschaltet. Dabei ist er so anzubringen, dass er offen ist, wenn die Sanduhr aufrecht steht, und schließt, wenn die Sanduhr auf dem Kopf steht. Um Störungen durch Vibrationen zu unterdrücken, kann es sinnvoll sein, noch einen Kondensator von z. B. 1 μF parallel zum Neigungsschalter zu löten.



4. Die beiden PixBlocks werden mit dem Flachbandkabel verbunden.
OUT von PixBlock 1 geht an IN von PixBlock 2.
5. Zuletzt lötet man die LED-Matrizen auf.

Bedienung



Zeiteinstellung auf 3 1/2 Minuten

Die Bedienung ist sehr einfach.

- Taster S1 stellt die Minuten ein.
- Mit Taster S2 kann man die Viertelminuten wählen.
- Taster S3 startet die Sanduhr neu.

Wenn die Sanduhr einmal mit einer Zeiteinstellung komplett durchgelaufen ist, wird diese Einstellung im EEPROM gemerkt, so dass sie auch nach einem Stromausfall wieder zur Verfügung steht.

Software



Die Software steht auf [Github](#) zur Verfügung.

Das obere Glas der Sanduhr wird mit 54 Sandkörnern gefüllt. Entsprechend der eingestellten Zeit fallen diese in schnellerer oder langsamerer Folge in das untere Glas.

Die maximal einstellbare Anzahl von Minuten kann im Programmcode über MAX_MINUTES eingestellt werden.

Flashen des Controllers

Der Mikrocontroller muss noch mit der „Bits of Time“-Software programmiert werden. Dazu benötigt man ein passendes Programmiergerät (z. B. [ARVisp MKII](#), [USBasp](#) o. ä.). Zuerst müssen die Fuses im Controller richtig gesetzt werden (siehe Datei „fuses.txt“). Anschließend wird die Datei „Bits_of_Time.hex“ auf den Mikrocontroller übertragen.

Kalibration

Die Sanduhr verwendet den internen RC-Oszillator des Prozessors als Zeitbasis. Dieser hat werksseitig eine Genauigkeit von 10 % oder besser. Wer eine höhere Präzision braucht, hat zwei Möglichkeiten.

1. Man kalibriert den RC-Oszillator indem man zu Beginn des Programms einen geeigneten OSCAL-Wert lädt. Wie dies genau vor sich geht und wie man den passenden Wert findet, kann man in den Application Notes von [Atmel](#) nachlesen.
2. Über den Parameter „CALIBRATION“ lassen sich in der Software die Zeitdauern anpassen. Sollte die Sanduhr bei Einstellung auf 1 Minute, z. B. tatsächlich nur 53 Sekunden zum Ablaufen benötigen, dann setzt man CALIBRATION auf (60.0 / 53.0).

PWM-Ausgang



Anschluss für Servo

An Ausgang PA7 (CON5, Pin 10) kann ein PWM-Signal erzeugt werden. PWM_MODE wählt zwischen normalem und invertiertem Modus. Die Einstellung des Prescalers (PWM_PRESCALER) legt die PWM-Frequenz fest. Bei PWM_PRESCALER = 4 beträgt die Frequenz 61,3 Hz. Damit eignet sich das Signal zur Ansteuerung von Modellbauservos. Auf der Platine sind neben CON5 drei Löt pads mit den Signalen GND, Vdisplay und PA7 vorhanden, an denen man den Servo anschließen kann.

Gehäuse

Das oben gezeigte Holzgehäuse ist natürlich nur eine von vielen Möglichkeiten.

„HorstKevinObst“ hat zum Beispiel ein sehr schickes Designergehäuse entworfen, das durch Leichtigkeit und Eleganz besticht. Er hat seinen Entwurf veröffentlicht und stellt ihn auf [Thingiverse](#) auch anderen zur Verfügung.

Download

Datei	Lizenz
Bits of Time	siehe license.md
Bauanleitung	CC-BY-SA

CC-BY-SA = [Creative-Commons-Lizenz](#), Namensnennung, Weitergabe unter gleichen Bedingungen
CC0 = [Creative-Commons-Lizenz](#), ohne Einschränkungen

[PixBlock](#), [Sanduhr](#), [Timer](#), [Elektronik](#), [Bausatz](#), [Led](#), [Dot-Matrix](#), [Anzeige](#), [Display](#)

From:

<http://www.doku.fab4u.de/> - **fab4U**

Permanent link:

http://www.doku.fab4u.de/de/projects/bits_of_time/start

Last update: **2018/07/03 21:12**

