

Sailor's Clock

Ralf Beesner

17.9.2011



1 Einleitung

Ein Freund bekam vor vielen Jahren eine Schiffsuhr (Yachtzubehör) geschenkt, die seither in seinem Wohnzimmer die Stunden schlägt. Das Schlagwerk ist jedoch für Landratten ungewohnt, denn es orientiert sich am Schiffsbetrieb.

In der christlichen Seefahrt dauer(t)en die Wachen 4 Stunden in einem Zweischicht-

oder Dreischicht- System. Auf den Windjammern wurde die Zeit mit der Schiffsglocke geschlagen:

- Einzelschlag zur halben Stunde
- Doppelschlag für jede volle Stunde

Beispiel: um 12.00 Uhr endete die alte Wache mit 4 Doppelschlägen. Um 12:30 Uhr ertönte ein Einzelschlag, um 13:00 Uhr ein Doppelschlag; um 13:30 ein Einzelschlag und (mit etwas Pause dazwischen) ein Doppelschlag; um 14:00 dann zwei Doppelschläge.

Eigentlich ist das viel praktischer als das übliche 12- Stunden- Schema von Turmuhren und Wohnzimmeruhren - ob es Vor- oder Nachmittag, morgens oder abends ist, weiß man ja, und während man z.B. um 11:00 Uhr bei einer Kirchturmuhr die 4 Viertelstunden- Schläge und 11 Stunden- Schläge mitzählen muss, sind es bei der Schiffsuhr nur 3 Doppelschläge.

2 AtTiny13 ausreizen

Das Thema „Morseuhren“ ist ja schon ziemlich „ausgelutscht“, aber mich wurmte immer, dass meine „Morseuhr 1“ (<http://elektroniklabor.de/AVR/Morseuhr.html>) in der AtTiny 13- Version einen externen Sekudentakt aus einer abgewrackten Analoguhr-Elektronik benötigte, da der AtTiny13 keinen Quarzoszillator und nur einen Timer hat.

Er lässt sich jedoch recht einfach an einem externen Quarzoszillator betreiben, und Töne kann man (mit Abstrichen bei der Ton- Konstanz) auch ohne Timer mit dem Bascom- Befehl „Tone“ erzeugen, so dass der Timer für die Erzeugung der Sekundenimpulse frei wird.

Herausgekommen sind 3 Uhrenvarianten:

- Sailor’s Clock
- eine Uhr mit konventionellem 12- Stunden- Schlagwerk
- eine „standalone“- AtTiny13- Morseuhr

3 Hardware

Im Beitrag „AtTiny13- Mutteruhr“ (<http://www.elektroniklabor.de/AVR/Mutteruhr.html>) hatte ich über Probleme berichtet, einen 3,6864 MHz- Pierce- Quarzoszillator mit ge-

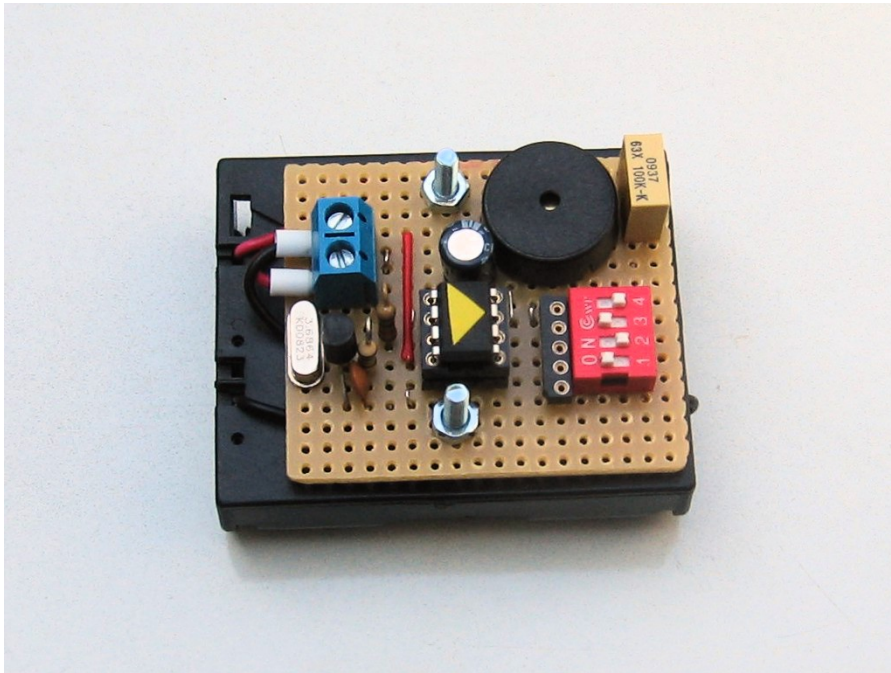


Abbildung 1: Sailor's Clock

ringem Stromverbrauch und großer Amplitude zu bauen (ich hatte dann stattdessen einen 32-kHz- Quarz in einer Multivibrator- Schaltung verwendet).

Wie ich später feststellte, saß das Problem vor dem Oszilloskop. Ich hatte den Tastkopf des Oszilloskops auf 1:1 gestellt und die Last, die er für den Oszillator darstellte, völlig unterschätzt. Nachdem ich bei einem neuen Versuch den Tastkopf auf 1:10 stehen hatte, stellte sich heraus, dass ein Pierce- Oszillator auch bei 1V schon wunderbar schwingt, nur etwas über 100 μA verbraucht und eine sichere Taktquelle für den AtTiny 13 darstellt.

Der externe Takteingang des AtTiny13 ist PB3. Der Buzzer kann an einem beliebigen Port liegen; ich habe PB4 gewählt. Dadurch liegen die 3 Mäuseklavier- Schalter für die Zeitvorwahl an PB0 - PB2. Das ermöglicht einfaches Rechnen und ist auch beim Flashen des Controllers praktisch, da die drei Pins bei offenen Schaltern lastfrei sind.

Da der Controller sich ohne externe Taktquelle nicht programmieren lässt, habe ich eine Single-in-Line- Fassung vorgesehen (für Masse, MOSI, MISO, SCK); man kann 4 Drähte einstecken, mit der Platine des LP Mikrocontroller verbinden und darüber das Flashen durchführen.

Die Schaltung läuft mit ca. 1,5V - 5V. Die Stromversorgung kann durch 2 oder 3

Mignonzellen bzw. eine Lithiumzelle erfolgen (mein Prototyp ist huckepack auf einen Batteriehälter für 3 Mignonzellen geschraubt).

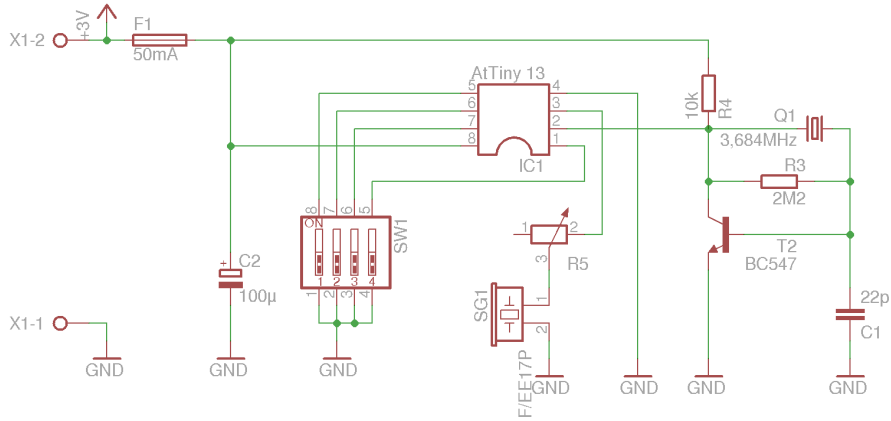


Abbildung 2: Schaltbild

4 Software

Von den 3 Uhrenvarianten gibt es neben der Quarztakt- Version je eine Testversion für den internen 1,2 MHz- Oszillator. Mit diesen ist die Programmentwicklung praktischer (und weil man mit ihnen die Programme ausprobieren kann, ohne den Controller auf externen Takt umzufusen, sind sie beigelegt).

Die Programme sind ähnlich aufgebaut - über das CLKPR- Register wird zunächst der Takt heruntergeschaltet, um Strom zu sparen. Der Timer wird so konfiguriert, dass er jede Sekunde (bzw. jede zweite Sekunde) einen Interrupt wirft. Die Interrupts werden gezählt und lösen alle 15 min (Morseuhr) bzw. alle 30 min das Schlagwerk aus.

Die Morseuhr ist aufwendiger als die beiden anderen Varianten und passt gerade noch in 1 KByte Flashspeicher.

Da die anderen beiden Programme recht kurz sind, habe ich herumprobiert, um den Ton prägnanter zu machen. Aber ohne PWM lässt sich nur wenig ausrichten, und übriggeblieben ist daher ein schlichter Doppelton, der ein wenig an PC- Spiele der achziger Jahre erinnert.

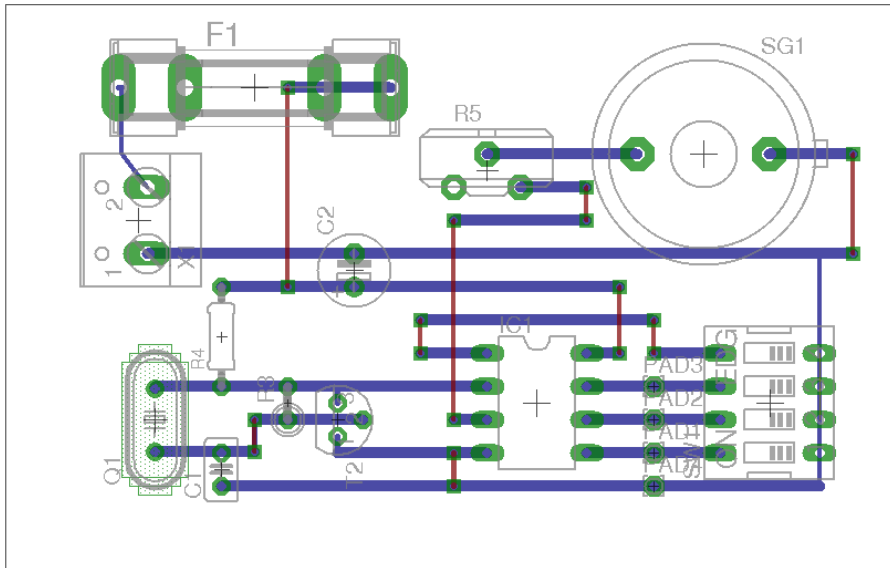


Abbildung 3: Streifenraster-Layout

5 Inbetriebnahme

Nach Aufbau der Schaltung sollte man sie zunächst ohne Controller in Betrieb nehmen und prüfen, ob der Quarzoszillator tatsächlich schwingt (vorzugsweise mit einem Oszilloskop oder durch Abhören mit einem Kurzwellenempfänger, der auf 3,686 MHz eingestellt ist).

Notfalls kann man den Stromverbrauch messen und mit einem Kondensator (ca. 100 nF), den man an Masse und Basis des Transistors klemmt, dafür sorgen, dass die Schwingungen abreißen. Die Stromaufnahme müsste sich dann ändern.

Dann sollte man das Programm in den AtTiny flashen und als letztes den Controller auf externen Takt umfusen.

Da dieses mit Burkhard Kainkas MikroISP.exe nicht möglich ist, muss man ein anderes Tool verwenden - z.B. avrdude.

Die erforderlichen Fusebytes sind: lfuse:0x68 und eventuell hfuse:0xef (letzteres schaltet den Brownout ab, falls man mit geringer Betriebsspannung arbeiten möchte).

Danach ist der Controller ohne externen Takt nicht mehr ansprechbar.

Wer die Programme nur mal kurz ausprobieren möchte, sollte die Versionen mit „1200000“ im Namen verwenden, die mit dem internen RC- Oszillator des AtTiny 13 arbeiten

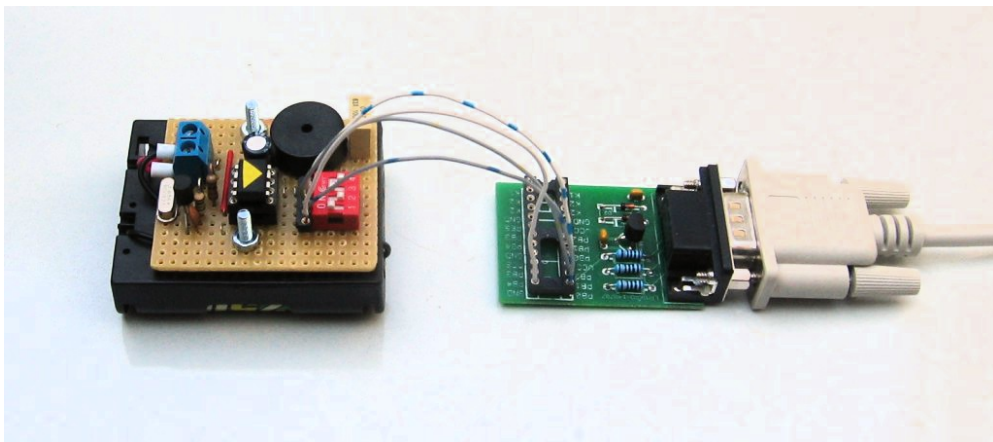


Abbildung 4: alles bereit zum Flashen