

Countdown zum fertigen Kuchen

Der ultimative Küchen-Timer

Uhren und Timer stehen schon immer in der ersten Reihe, wenn es um die Realisierung ganz individueller Selbstbauprojekte geht – hier gehen sowohl Anlässe als auch Themen niemals aus. Das beweist auch das Projekt unseres Lesers Hans-Joachim Beck, der sich einen Küchen-Timer genau nach Maß gebaut hat. Wir finden, der pfiffige Timer eignet sich durchaus auch für den Nachbau als Mikroprozessor-Einsteigerprojekt, da hier gleich mehrere grundlegende Teilbereiche wie I²C, Anzeigen, Eingaben und Tonausgabe zugleich behandelt werden und gezeigt wird, dass Mikroprozessor-Bausteine sehr viel flexibler sind als Standard-Bausteine. Selbstverständlich liegt die Software offen.

Vielen Dank!

Hans-Joachim Beck

hat für seinen Beitrag zum Leserwettbewerb einen 200-Euro-Warengutschein erhalten!
Sie wollen auch mitmachen? Infos auf Seite 93.

Ultimativer Küchen-Timer

Alles begann damit, dass wieder einmal unser Küchenwecker stehen blieb, während ein Kuchen im Ofen war. Daraus reifte die Idee, sich so etwas doch selbst zu bauen, und zwar genau mit den Funktionen, die man ansonsten schon immer an käuflichen Küchenweckern vermisst hatte. So entstand ein in meinen Augen **Ultimativer Küchen-Timer: UTK**.

Ursprünglich wollte ich zunächst nur einen Küchenwecker bauen, der sich schnell auf eine Zeit einstellen lässt und dann herunterzählt. Nach Ablauf der Zeit sollte er sich akustisch melden und dies auch mehrfach wiederholen, falls man das Ende-Signal nicht sofort hört.

Beim Nachdenken über die hauptsächlichen Nutzungen fielen mir zunächst zwei Hauptanwendungen ein: das Kuchenbacken mit der damit verbundenen Timer-Laufzeit von 30 bis 90 Minuten – hier ist eine Auflösung in Minuten ausreichend. Die zweite Hauptanwendung ist das Kochen von Eiern, hier benötigt man eine deutlich kürzere Laufzeit, aber praktischerweise in Sekunden-Teilung.

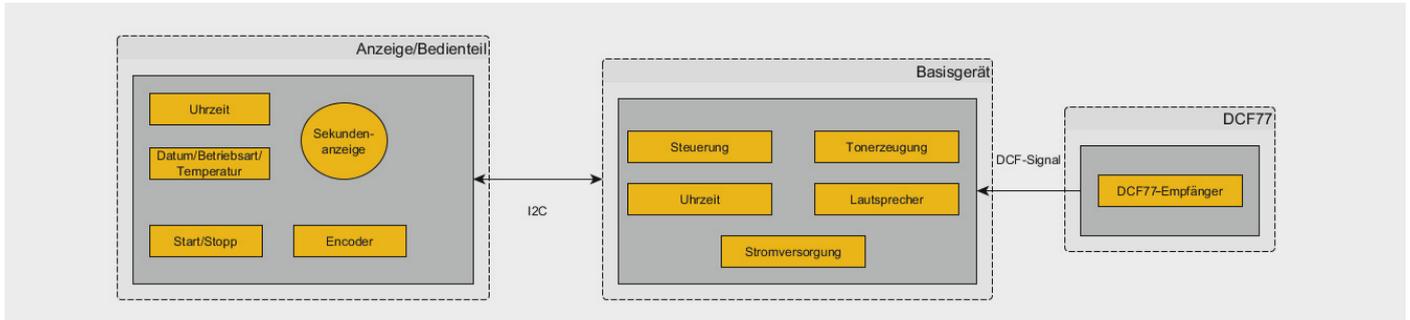


Bild 1: Die Baugruppen des Küchen-Timers

Wenn man das alles schon kann, was macht dieser Küchen-Timer, wenn er eigentlich nichts zu tun hat? Die Uhrzeit anzeigen – so sollte es sein. Wie bei solch einer Ideenfindung üblich, kamen nach und nach weitere Funktionen hinzu. So eine Stoppuhr, und da dann schon alles enthalten ist, was noch weitere Funktionen erfüllen könnte, wurde zunächst auch noch eine wechselseitige Anzeige der Uhrzeit und des Temperaturwerts zweier DS1820-Temperatursensoren für Innen- und Außentemperatur geplant. Da allerdings ein Standort für einen Außenfühler nicht ohne höheren Aufwand zu erreichen war, habe ich diesen Part vorerst wieder fortgelassen. Er kann aber, da es dafür leicht implementierbare Standard-Softwarebausteine gibt, jederzeit wieder eingebunden werden. Genau das habe ich dann auch mit dem zunächst letzten Feature gemacht: Die Uhrzeit sollte nicht nur manuell eingebbar sein, sondern über einen DCF77-Empfänger empfangen und eingestellt werden.

I²C für grünen WAF

Um dies alles, inklusive des benötigten Lautsprechers in einem Gehäuse unterzubringen, hätte es eine Gehäusegröße im Bereich der Anzeige erfordert, die am Ende den WAF (Woman Acceptance Factor) in Richtung Rot getrieben hätte, also mussten Anzeige inklusive Bedienteil vom Lautsprecher und vom Grundgerät getrennt werden. Als ich die Bedienelemente zusammen hatte, waren es viel zu viele Kabel – 2 Taster und ein Encoder sind sicher schon 5 Leitungen, dann die Spannungsversorgung und die Ansteuerung der Displays. Daher wurde das Bedienteil mit dem Basisteil über I²C gekoppelt. Bild 1 zeigt die Geräteanordnung inklusive des abgesetzten DCF77-Empfängers in der Übersicht.

Anzeige- und Bedienteil

Als Anzeige für die Uhrzeit (HH:MM) habe ich ein vierstelliges 7-Segment-Modul gewählt, das über den I²C-Bus angesteuert werden kann. Als Multifunktionsanzeige setzte ich eine vierstellige alphanumerische Anzeige ein, die ebenfalls über den I²C-Bus ansteuerbar ist.

Als ich eigentlich schon alle Teile zusammen hatte, stieß ich zufällig auf einen schönen LED-Ring aus 60 WS2812-LEDs, deren Ansteuerung seriell über eine einzige Busleitung erfolgt [1]. Wenn man so etwas wie diesen Timer selber baut, dann baut man eben das alles ein, was man sowieso immer schon mal gern ausprobieren wollte, also wurde dieser Ring als Sekundenanzeige eingesetzt. Bild 2 zeigt die Anordnung der Anzeige- und Bedienelemente. Dazu habe ich ein etwas dickeres Brett gewählt und in diesem den LED-Ring bündig eingelassen und auch die zwei Anzeigen. Da die Leiterplatte bei den beiden Anzeigen in der Mitte größer war als die Anzeigen, haben sich zunächst unschöne Lücken ergeben, die habe ich inzwischen durch eine schwarze Blende unter meiner Plexiglasfront verdeckt.

Unten links im Bild ist die Start- und rechts die Stopp-Taste zu sehen. Rechts oben ist seitlich der Encoder angebracht.

Um, wie oben erwähnt, die Anzahl der Verbindungsleitungen zum Basisgerät zu verringern, habe ich in der Anzeige einen kleinen Mikroprozessor eingesetzt, der als I²C-Slave fungiert und die Bedienelemente erfasst. Hier kam ein ATtiny44A zum Einsatz, der folgende Aufgaben hat:

- **Auslesen der Tasten inkl. Entprellung**

Diese Aufgabe ist einfach, hier werden die Tasten einfach entprellt. Ich habe hier eine Prellzeit von 100 ms unterdrückt, obwohl ich diese recht lange Zeit bisher bei keiner Taste festgestellt habe.

- **Auswertung des Encoders**

Hier werden die beiden Kontakte des Encoders ausgewertet und natürlich auch der Taster, der im Encoder eingebaut ist.

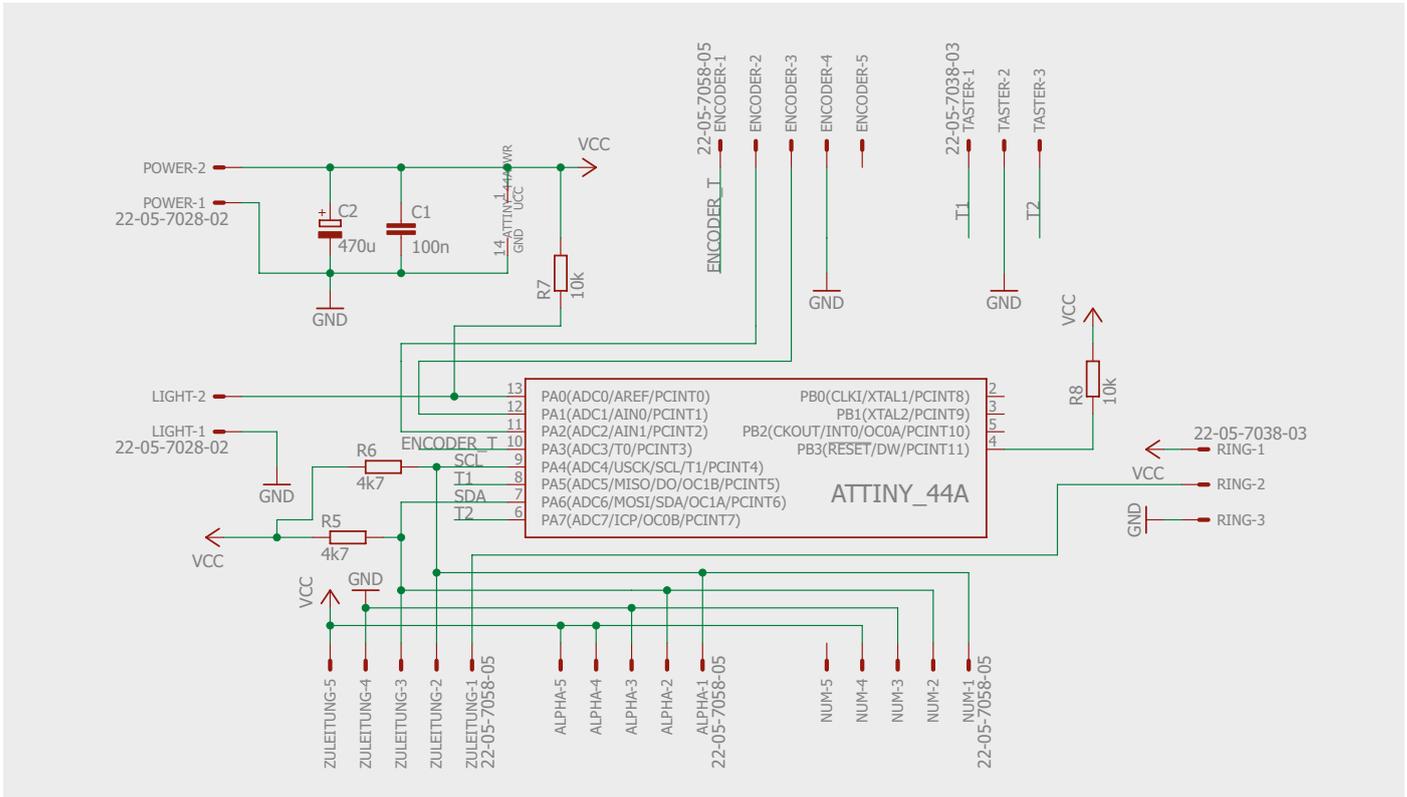
Ein solcher Encoder erzeugt innerhalb kurzer Zeit viele Änderungen, häufig schneller, als diese Informationen über den I²C-Bus gelesen werden können. Daher puffert dieser Prozessor die Zählimpulse zwischen und liefert dann über den I²C-Bus die Anzahl der Impulse seit der letzten Abfrage. Das hat in meinem Fall die Reaktion auf Änderungen am Encoder auf die Anzeige deutlich verbessert.

- **Auswertung eines LDR**

Und weil es immer eine Verbesserungsmöglichkeit gibt, habe ich noch einen Fotowiderstand (LDR, in Bild 2 zwischen den beiden vierstelligen Anzeigen zu sehen) eingesetzt, um die Umgebungshelligkeit zu erfassen. Dadurch kann sich die Helligkeit der Anzeige dem Umgebungslicht anpassen.



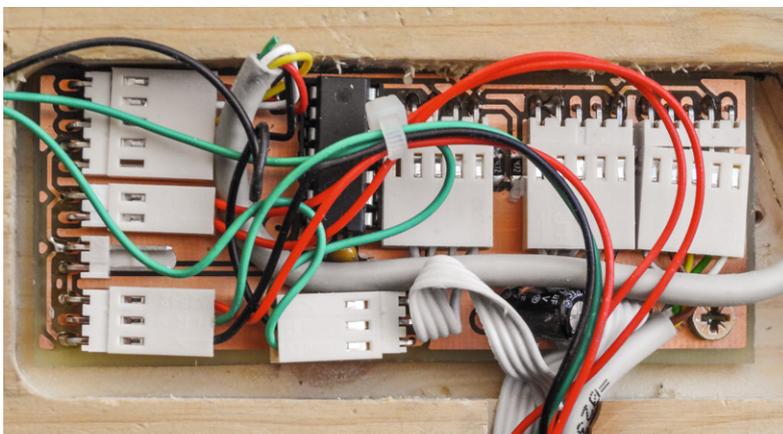
Bild 2: So sind Anzeigen und Bedienelemente angeordnet.

Bild 3: Die Schaltung des I²C-Slaves

Diese Informationen werden über den I²C-Bus als Datenblock übermittelt, der folgenden Aufbau hat:

Tasten	Byte
Encoder	Signed Integer
Helligkeit	Unsigned Integer

Die Schaltung des I²C-Slaves ist in Bild 3 zu sehen. Da nun doch einiges an einzelnen Anschlüssen und der Prozessor im Bedienteil anzuschließen waren, habe ich dazu eine kleine Leiterplatte erstellt, über die die einzelnen Komponenten miteinander verbunden werden. Wie dies eingebaut und verkabelt aussieht, zeigt Bild 4, der hier leer gebliebene Stecker ermöglicht eine zusätzliche Stromversorgung für das Anzeige- und Bedienteil, sie erwies sich jedoch als nicht nötig. In Bild 5 sind

Bild 4: Die I²C-Slave-Platine mit voller Anschlussbestückung. Hier geht es recht gedrängt zu.

die in die Rückseite der Trägerplatte eingelassene Platine und die Gesamtverdrahtung der Anzeige- und Bedienelemente zu sehen.

Das Basisgerät

Das Basisgerät besteht aus dem steuernden Mikroprozessor, hier habe ich zu einem gerade vorhandenen Arduino Pro Mini gegriffen, der auf dem ATmega 328 basiert, einem RTC als stabile und netzausfallsichere Uhrzeitbasis und einem weiteren kleinen Mikroprozessor, der die Programmierung von Quittungsmelodien einfach macht.

Wie gesagt, als Hauptprozessor kommt der Einfachheit halber ein Arduino Pro Mini zum Einsatz, die Software ist nicht so komplex, sodass man sie

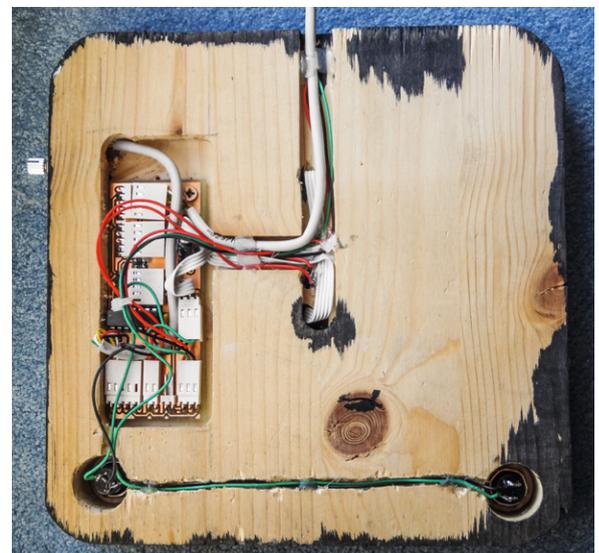


Bild 5: Die Gesamtverkabelung in der Trägerplatte der Bedien- und Anzeigeeinheit



bequem per Arduino IDE entwickeln und übertragen kann. So steht sie auch als Arduino-Sketch zum Download bereit.

Als Uhrzeitbasis habe ich einen DS1307 eingesetzt. Um den auch bei einem Spannungsausfall weiterlaufen lassen zu können, ist ein GoldCap zur Notstromversorgung eingebaut.

Jetzt fehlte neben der Stromversorgung eigentlich nur noch der Soundgenerator. Zunächst hatte ich hier vor, einen bewährten 3-Klang-Gong-Schaltkreis vom Typ SAE800 einzusetzen, aber da war mir die Gestaltungsmöglichkeit der Töne zu eingeschränkt. Töne sind ja auch mit dem eigentlichen Steuerprozessor zu erzeugen – hier ist ohnehin kein Hi-Fi gefragt –, aber als ich das auch noch mit meinem Steuerprozessor machen wollte, da musste ich mich entscheiden:

- Genaue Anzeige der Sekunden, dann muss ich viel implementieren hinsichtlich der Tonlängen
- Tonerzeugung vereinfachen, dann habe ich aber zusätzlichen Aufwand, um die Sekundenanzeige exakt ausgeben zu können.

Also habe ich die Melodierzeugung ausgelagert, und da sowieso schon ein I²C-Slave implementiert ist, war der Schritt zum insgesamt dritten Prozessor nicht weit. Hier habe ich einen ATTINY13 eingesetzt. Dieser reicht mehr als aus, um ein paar einfache Melodien zu speichern, die dann über einen I²C-Befehl abgerufen werden können. Einmal ausgewählt, wird die Melodie unabhängig von den Aktivitäten des Hauptprozessors abgespielt. Die Steuerung für die Uhrzeitanzeige läuft unbeeinträchtigt weiter und eine Bedienung ist jederzeit möglich.

Der trotzdem noch sehr überschaubare Stromlaufplan ist in Bild 6 zu sehen. Wie gesagt, eine Schnittstelle für die Temperatursensoren ist bereits vorhanden, wird aber derzeit softwaremäßig nicht genutzt.

Bild 7 zeigt die hierfür erstellte, zum verwendeten Gehäuse passende Hauptplatine.

Die DCF77-Synchronisation

Zunächst wollte ich mir das ersparen – aber eigentlich wollte ich schon immer einmal eine DCF77-Decodierung bauen, also habe ich den ELV-DCF77-Empfänger in ein kleines, abgesetztes Gehäuse eingebaut und hier angeschlossen. Ich habe diesen Empfänger von meiner Basisplatine getrennt, um im Falle von Störungen seitens der Mikroprozessorschaltungen etwas unabhängiger zu sein. Aber mittlerweile weiß ich, dass dies unnötig war, ich empfangen eigentlich fast immer ein sauberes Signal.

Zur Verifikation der empfangenen Zeit: Ich synchronisiere meine interne Zeit mit dem DCF-Empfänger immer um 03:00 Uhr. Ich werte dazu die Parity-Informationen im DCF77-Protokoll aus und benutze diese empfangene Zeitinformation nur dann, wenn ich zwei gültige Telegramme bekommen habe, bei denen Jahr, Monat, Tag und Stunde gleich sind, und die Minuteninformation nur eine Differenz von 1 aufweist.

Die Bedienung

Wie bereits zu Beginn erwähnt, sollte die Bedienung möglichst einfach erfolgen, nämlich mit den drei Bedienelementen Start-/Stopp-Taster und dem Encoder zur Zeitanwahl.

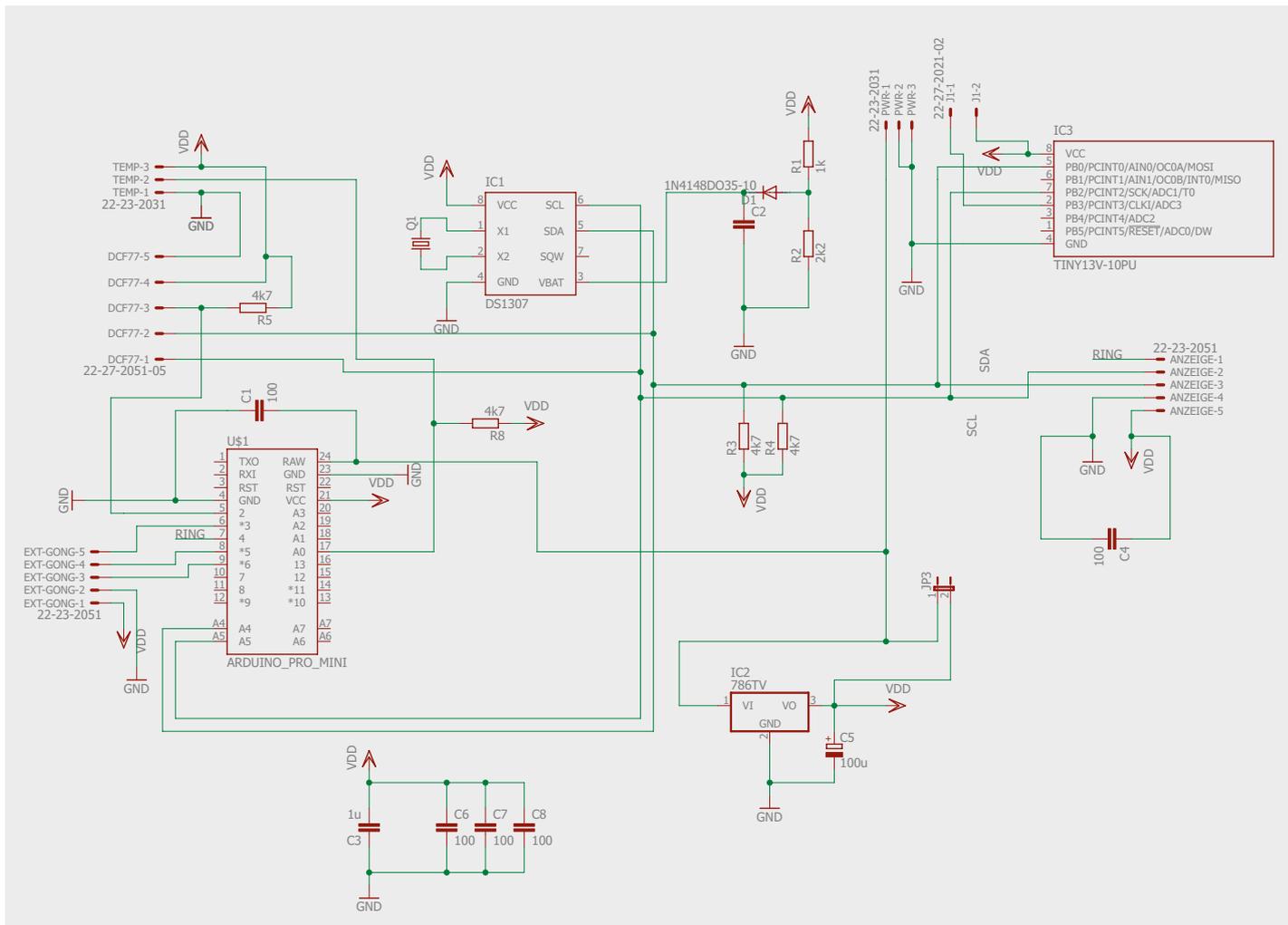


Bild 6: Das Schaltbild der Hauptplatine mit Steuerprozessor, RTC, Stromversorgung und Tonerzeugung

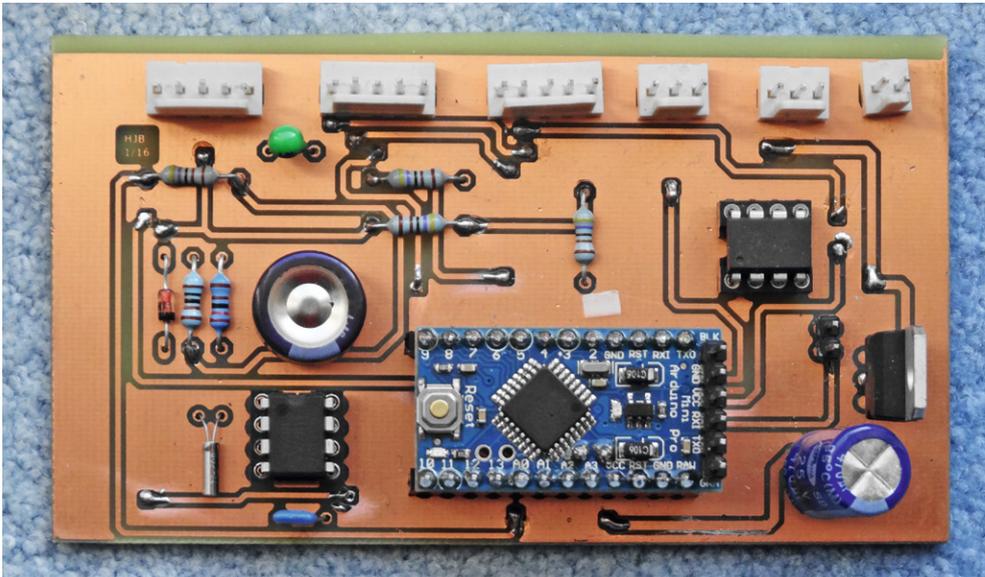


Bild 7: Die zum verwendeten Gehäuse passende Hauptplatine

Erste Priorität hat die schnelle Erreichbarkeit des Rückwärtszählers: Bestätigung der Start-Taste links, 30 Min sind voreingestellt. Mit dem Encoder kann mit einer einfachen Drehung schnell im Minutenraster die Zeit geändert werden, noch einmal Start, und der Rückwärtszähler läuft (der „Sekundenzeiger“ läuft links herum, also gegen den Uhrzeigersinn). Wenn der Rückwärtszähler abgelaufen ist, kann man mittels der Stopp-Taste den weiteren Ablauf stoppen und schnell eine neue Zeit eingeben und damit noch eine Nachlaufzeit einstellen.

In Bild 8 ist das gesamte Bedienkonzept abgebildet. Wie man sieht, habe ich das Ganze in verschiedene selbsterklärende Betriebszustände gegliedert, von denen wir einige näher betrachten wollen:

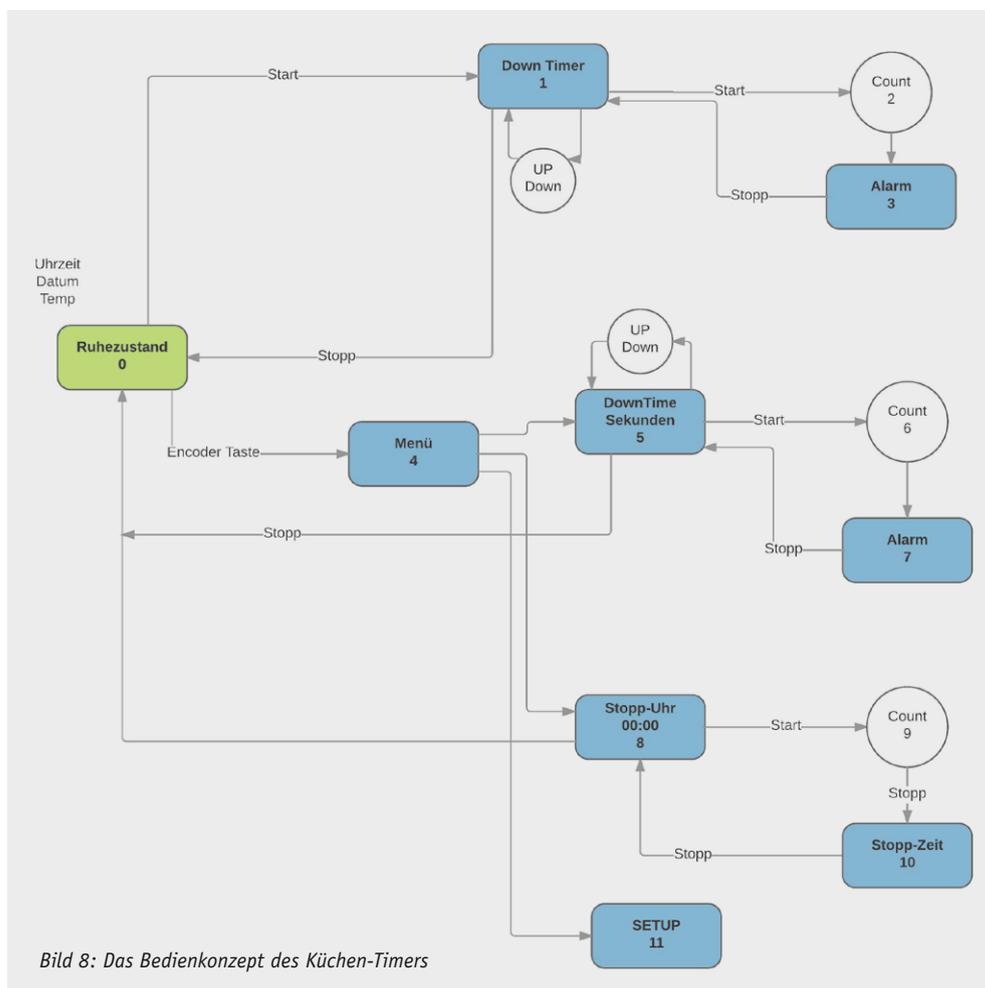


Bild 8: Das Bedienkonzept des Küchen-Timers

Betriebszustand 0 (Ruhezustand)

In diesem Zustand wird die Uhrzeit (Bild 9 zeigt die DCF77-Synchronisierung) angezeigt und dann dazwischen (wie gesagt, derzeit nicht realisiert) sporadisch die Innen- und Außentemperatur.

Diesen Betriebszustand kann ich durch den Start-Taster oder durch den Taster im Encoder jederzeit ganz einfach verlassen.

Betriebszustand 1

Durch Betätigung des Start-Tasters kommt man in den Betriebszustand 1 (Bild 10). Hier kann mit dem Encoder die Zeitvorgabe inkrementiert (heraufgezählt) oder dekrementiert (herabgezählt) werden. Hier nutze ich die Stunden/Minuten-Anzeige zur Anzeige der Laufzeit des Timers. Zusätzlich sind die LEDs im Ring angeschaltet, um eine schnelle Übersicht zu haben. Hier entspricht jede LED einer Minute.

Durch eine wiederholte Betätigung des Start-Tasters kommt man in den Betriebszustand, in dem die Zeit heruntergezählt wird. Durch den Stopp-Taster gelangt man sofort wieder zurück in den Bereitschaftszustand 0. Zusätzlich ist zu beachten: Nach 20 s im Betriebszustand 1 ohne Tastenbetätigung wechselt das Programm automatisch wieder in den BZ 0.

Betriebszustand 2 (Bild 11)

Jetzt läuft der Timer ab. In der Stunden/Minuten-Anzeige werden die verbleibenden Stunden und Minuten angezeigt, im Ring laufen die Sekunden – wie es sich gehört – gegen den Uhrzeigersinn. Ist der Timer abgelaufen, ertönt das Alarmsignal, die Stunden/Minuten-Anzeige blinkt und zeigt die Stunden/Minuten, die seit dem Ablauf des Timers vergangen sind. Im Ring werden die Sekunden angezeigt, nun im Uhrzeigersinn. Alle 10 s erfolgt die Ausgabe eines kurzen Alarmtons, zu jeder vollen Minute ertönt ein längerer Alarmtone. Mit dem Stopp-Taster kommt man zurück in den BZ 2 und kann bei Bedarf schnell noch einmal eine andere Laufzeit eingeben.

Betriebszustand 9 (Bild 12)

Hier ist die Stoppuhrfunktion erreichbar. Nach dem Start mit dem



Bild 9: Die Uhrzeitanzeige – hier läuft die DCF77-Synchronisierung.



Bild 10: Anzeigebeispiel für den Rückwärtszähler, hier 45 Minuten

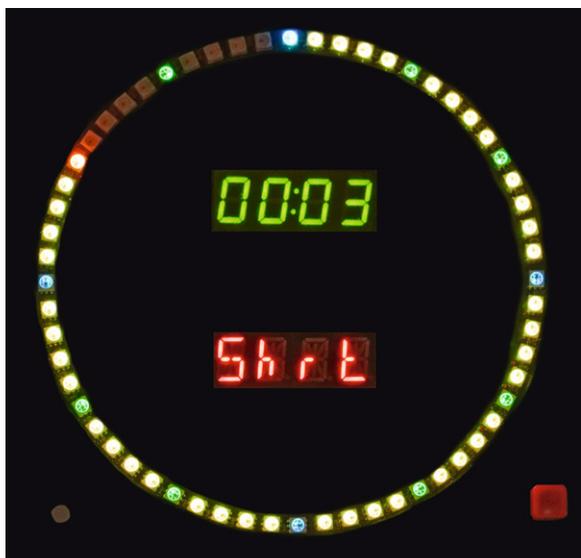


Bild 11: Der Timerlauf ist mit 3 Minuten und 50 Sekunden eingestellt.



Bild 12: Betriebszustand 9 – die Stoppuhr läuft seit 12 Sekunden.

Start-Taster wird die Zeit mit dem Stopp-Taster angehalten, erneutes Drücken führt wieder auf Null zurück.

Die Software für alle Abläufe ist hier aus Platzgründen nicht abgedruckt, sie wird daher im Onlinebereich des ELVjournal [2] kostenfrei zur Verfügung gestellt. **ELV**



Weitere Infos:

- [1] LEDs mit Intelligenz, ELVjournal 4/2015, S. 54
- [2] Software zum Projekt: Webcode #1466



Machen Sie mit!

Jede veröffentlichte Anwendung wird mit einem Warengutschein in Höhe von 200 Euro belohnt.

Wir wollen es wissen – Ihre Anwendungen und Applikationen!

Welche eigenen kreativen Anwendungen und Applikationen haben Sie mit den ELV-Haustechnik-Systemen, aber auch mit anderen Produkten und Bausätzen realisiert – ob mit Standard-Bausteinen oder eingebunden in eigene Applikationen? Alles, was nicht gegen Gesetze oder z. B. VDE-Vorschriften verstößt, ist interessant. Denn viele Applikationen verhalfen sicher anderen zum Aha-Erlebnis und zur eigenen Lösung.

Schreiben Sie uns, fotografieren Sie Ihre Applikation, berichten Sie von Ihren Erfahrungen und Lösungen. Die interessantesten Anwendungen werden redaktionell bearbeitet und im ELVjournal mit Nennung des Namens vorgestellt.



Per E-Mail
leserwettbewerb@elv.de



Per Post
ELV Elektronik AG, Leserwettbewerb, 26787 Leer

Die Auswahl der Veröffentlichungen wird allein durch die ELV-Redaktion ausschließlich nach Originalität, praktischem Nutzen und realisierter bzw. dokumentierter Ausführung vorgenommen, es besteht kein Anspruch auf Veröffentlichung, auch bei themengleichen Lösungen. **Der Rechtsweg ist ausgeschlossen.** Für Ansprüche Dritter, Beschädigung und Verlust der Einsendungen wird keine Haftung übernommen. Alle Rechte an Fotos, Unterlagen usw. müssen beim Einsender liegen. Die eingesandten Unterlagen und Aufnahmen verbleiben bei der ELV Elektronik AG und können von dieser für Veröffentlichungen und zu Werbezwecken genutzt werden.