

Küchentimer mit ESP32

Der Küchentimer ist den allseits bekannten Kurzzeitweckern nachempfunden, wie sie in vielen Küchen zu Hause zum Einsatz kommen.

Ziel im Projekt war es, ein für den täglichen Einsatz brauchbares Gerät zu konstruieren, um die Gar- oder Backzeit im Griff zu haben, oder an andere Dinge im Minutenbereich zuverlässig erinnern zu lassen, keine Fallstudie, sondern an den gängigen Vorbildern aus dem Handel orientiert und noch etwas mehr auf die eigenen Bedürfnisse abgestimmt.



Bild01 – Fertiges Gerät (b01.jpg)

Es soll gezeigt werden, was ein Eigenbau mit einem Mikroprozessor-Baustein wie dem ESP32 von Espressif im praktischen Gebrauch leisten kann. Daneben soll natürlich auch etwas experimentiert werden. Deshalb bietet der Entwurf zusätzlich noch einen HTTP-Server, so dass es möglich wird, alles auch über ein Smartphone oder Notebook zu kontrollieren.

Somit wird unser Timer ins lokale Web integriert und kann zu Hause divers kommunizieren, wie es die Zukunft der Geräte im „Smarthome“ verspricht.

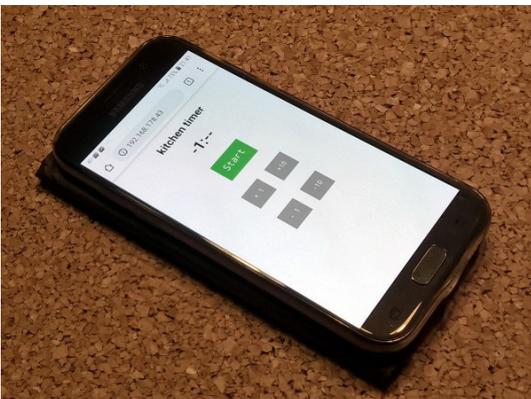


Bild02 – Bedienoberfläche auf dem Smartphone (b02.jpg)

Die Software steht auf GitHub bereit, das Sketch „kitchentimer.ino“ wurde für den ESP32 mit WiFi geschrieben. Im Quelltext sieht man, es braucht die „Adafruit LEDBackpack“- und die „Adafruit GFX“-Library. Beide Bibliotheken können in der Arduino-IDE recht einfach geladen werden. Als Board wählt man dort ein „ESP WiFi Kit“ oder Ähnliches aus.

Der Github-Link ist <https://github.com/swenae/kitchentimer> , wo neben den aktuellen Quellen auch die Verdrahtung und Bildmaterial zu finden sind.

Für den Schritt-für-Schritt-Aufbau ist es bei kleineren Projekten wie immer eine gute Idee, zuerst mit einem Steckboard zu arbeiten. Da lassen sich schnell Anpassungen und Modultests vornehmen.

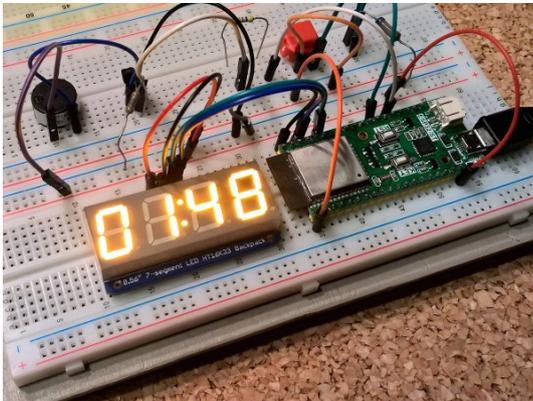


Bild03 – Breadboard (b03.jpg)

Die Schaltung braucht LED-Segmentanzeige mit I2C-Backpack, Taster, Piezo-Summer, Transistor und ein paar passive Bauteile. Die wenigen Komponenten kann man sich recht preiswert beschaffen.

Da diverse ESP-Boards unterschiedliche Außenbeschaltungen haben, also bei der Pinbelegung abweichen können, muss mitunter auf den eigenen Baustein angeglichen werden. Das Wiring passt bei uns auf einen „ESP32-ST wroom“ von eboxmaker. Aber man kann auch einen anderen Baustein aus dem eigenen Fundus nehmen.

Ist alles getestet, kann der schematische Aufbau fixiert werden. Manche verwenden „Fritzing“ als Software, um das Ganze zu dokumentieren.

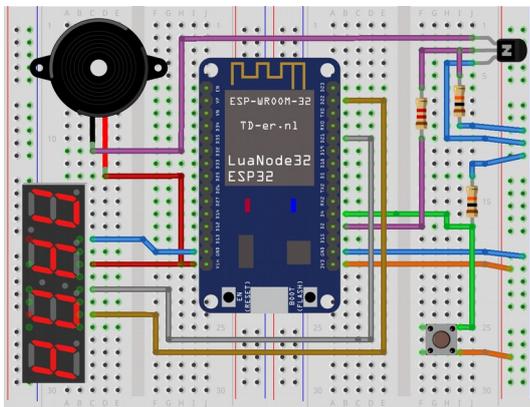


Bild04 – Schematischer Aufbau mit „Fritzing“ (b04.png)

Bei einer so übersichtlichen Beschaltung bot sich an, alles auf einer Leiterplatte unterzubringen, welche unter dem ESP-Modul angeordnet wurde. Externe Bauelemente wurden verdrahtet.

Zusätzlich sollte noch ein aufladbarer Akku ins Gehäuse. Der ist hier nicht im Bild. Ein 600mAh-Lipo wurde extra angeschafft, damit auch alles in unser kompaktes Gehäuse passt.

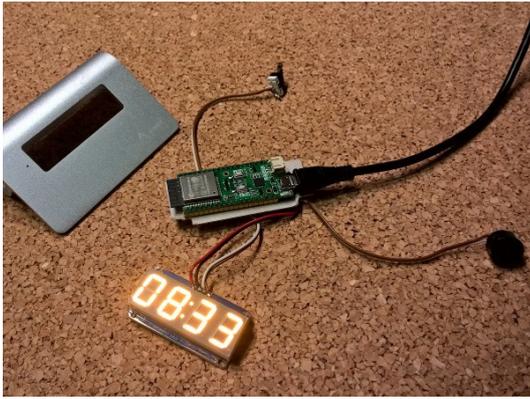


Bild05 – Bestückte Leiterplatte (b05.jpg)

Daneben ist noch eine kleine Platine mit der Ladeelektronik für den Lipo verbaut worden, die außerdem einen DC-DC-Wandler enthält und nach dem Akku 5V für LED-Segment-Anzeige und Piezo-Signalgeber abgibt. Die funktionieren sonst bei niedrigeren Spannungen leider nicht.

Zum Aufladen des Akkus wird nicht direkt am ESP, sondern an der Ladeelektronik angesteckt. Das geht im fertigen Aufbau am Unterboden vom Gehäuse. Links am Gehäuse befindet sich noch der Ein/Aus-Schalter, rechts der Signalgeber. Im Frontpanel hat rechts neben dem Display der Bedientaster Platz genommen.

Nun zur Web-Funktionalität. Um die Oberfläche aufzurufen, kann man sich ein Lesezeichen im Browser machen. Das Ganze lohnt sich, wenn eine lange Zeit eingestellt ist, wo man etwa nicht mehr in der Küche oder in der Nähe vom Timer ist.

Dann kann man auch von Ferne die Dinge unter Kontrolle halten. Das Webinterface aktualisiert nicht sekundengenau, sondern nur über ein paar Sekunden, gibt aber über das Ende sichere Auskunft.

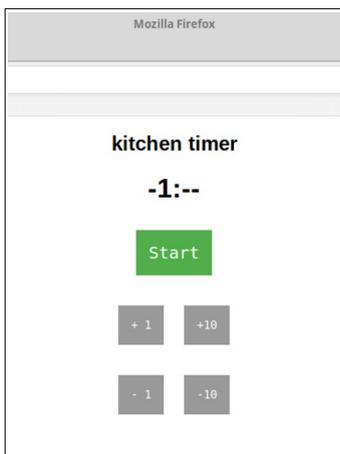


Bild06 – Startscreen - Webansicht (b06.png)

Im Unterschied zur Hardware gibt es hier mehrere Schaltflächen zur Eingabe. Da sich beim Druck auf einen Button einer Website schlecht feststellen lässt, wie lange gedrückt wurde, sind wir hier vom „Ein-Button-Prinzip“ abgewichen. Schließlich will man ja schnell an der gewünschten Voreinstellung für die Startzeit angelangt sein.

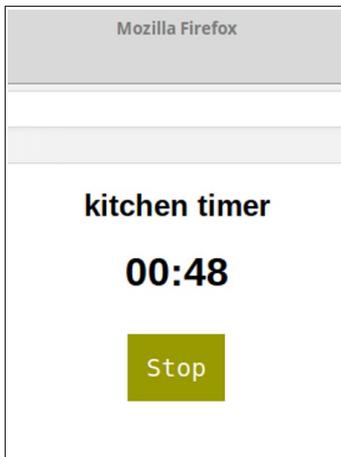


Bild07 – Running - Webansicht (b07.png)

Unser Kurzzeitwecker rennt los, ganz egal, ob man ihn am Gehäusetaster oder per „Start“-Button im Webinterface dazu angewiesen hat. Alles, was online festgelegt wurde, wird auch auf das Gerät übertragen.

Umgekehrt ändert auch eine Eingabe am Gerät die Anzeige auf der Website, so dass alles immer synchron ist. Ist der Countdown beendet, wird das Ganze im Webinterface und auf dem Gerät angezeigt.

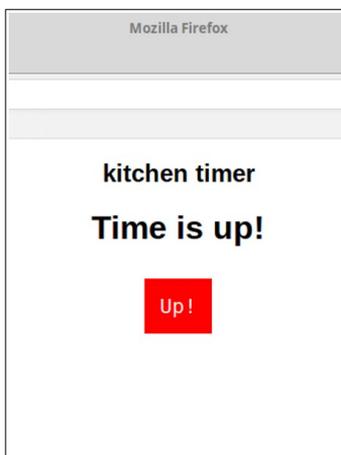


Bild08 – Zeit abgelaufen - Webansicht (b08.png)

Damit zurück zum nun fertigen Timer. Wichtig war mir eine gut hörbare akustische Meldung, weil man ja nicht immer in der Nähe sein möchte, um von der Anzeige abzulesen. Unser Timer meldet sich da wie eine klassische Küchenuhr. Und es gibt noch weitere Beepcodes:

Timer abgelaufen, Alarmmeldung	-	40 Beeps (500ms-Intervall)
Nach Einschalten, in WLAN eingeloggt	-	3 Beeps (200ms-Intervall)
Login fehlgeschlagen, Offline-Betrieb	-	2 Beeps (200ms-Intervall)
Abbruch des laufenden Countdown	-	1 Beep für eine Sekunde

Ein besonderes Ziel war die „Ein-Taster-Bedienung“ des Geräts. Ähnlich wie beim Aufziehen von mechanischen Kurzzeitweckern, gibt es bei unserem elektronischen Timer (neben dem Einschalter) nur eine Eingabemöglichkeit in Form eines Tasters rechts neben der

Segmentanzeige. Denn wer mag schon komplizierte Küchengeräte, bei denen man erst überlegen muss, wie denn die Bedienung der vielen Knöpfe eigentlich war.



Bild09 – Bedienung (b09.jpg)

Daneben erschien eine gute Bedienerführung mittels diverser Anzeigen auf dem Display genauso wichtig. Bei unserem Timer sieht das so aus:

Nach dem Einschalten per WLAN eingeloggt	-	Anzeige	--	C0
kein Netzwerk gefunden	-	Anzeige	--	EE
Timer steht danach auf Preset von 1min	-	Anzeige	-1:--	
gehaltene Taste rückt die Timerminuten vor	-	Anzeige	2, 3 usw.	

Um die gewünschte Voreinstellung schnell zu erhalten, wird in immer kleineren Zeitintervallen hochgezählt. Nach Erreichen der gewünschten Countdown-Zeit von bspw. 15min lässt man den Eingabetaster los und der Timer zählt sekundenweise abwärts. Man kann anschließend mit der gleichen Taste abbrechen und wie oben wieder neu stellen:

Ioslassen, der Countdown beginnt	-	Anzeige	15:00
Tastendruck während Laufzeit - Abbruch	-	Anzeige	EE EE

Läuft der Countdown regulär ab, wird per Anzeige und Signalton alarmiert und der Timer behält anschließend seine letzte Voreinstellung. Wird jedoch während der Laufzeit angehalten, springt der Timer auf den Preset von 1min zurück. So ist es möglich, eventuelle Fehleingaben schnell zu korrigieren.

Timer regulär abgelaufen, Anzeige blinkt	-	AA AA / -- --
--	---	----------------------

Soviel zur kurzen Bedienungsanleitung. Der Ablauf ist gerade im Praxistest, da wird sich herausstellen, wo etwa noch angepasst werden muss.

Ein guter Einsatzfall für eine klassische LED-Segment-Anzeige, die kann zwar weniger als ein Vollgrafik-Display darstellen, ist aber gut ablesbar und ein minimales User-Interface reicht uns aus. Alle Bauteile wurden in einem ausgedienten USB-Hub-Gehäuse untergebracht.

Unser Küchentimer hat sich in den ersten Tagen nach der Fertigstellung schon gut bewährt und wird bestimmt im täglichen Gebrauch Freude machen...