

Pretzelboard trifft Nextion-Display NTP-Multifunktionsdisplay-Uhr

Welches Potenzial das im ELV Journal 5/2016 vorgestellte Pretzelboard in sich trägt, zeigt die Lösung unseres Lesers Peter Tschulik. Er hat dieses mit einem Nextion-Touchdisplay der Firma ITEAD kombiniert und damit eine faszinierende Uhr mit verschiedenen Anzeigemöglichkeiten gebaut, die sich über WLAN und den NTP-Dienst immer mit der aktuellen Uhrzeit versorgt.



Schicke Uhr

Uhren üben seit langer Zeit eine Faszination auf viele Menschen aus. Es gibt sie in unzähligen Ausführungen, Preisklassen und Größen. Inspiriert von dem Artikel im ELV Journal 5/2016 zum hochinteressanten Pretzelboard habe ich mir ein solches Board besorgt und überlegt, wie ich dessen Leistungsfähigkeit ausreizen könnte. Schon länger war ich im Besitz eines intelligenten Nextion-Touchdisplays der Firma ITEAD [1], das ich über ein Kickstarterprojekt erworben habe.

So entstand die Idee, eine Uhr zu realisieren, die sich mit dem WLAN verbindet, über NTP die Zeit synchronisiert und auf dem hochauflösenden Display anzeigt. Gesagt, getan! Als Display kam ein NX487T043-011R [2] zum Einsatz, das recht preiswert ist und über eine Größe von 4,3 Zoll bei einer Auflösung von 480 x 272 Pixeln sowie einen resistiven Touchscreen verfügt. Statt wie früher die Grafiken kompliziert direkt über einen Mikrocontroller zu generieren, geht dieses Display einen anderen Weg: ITEAD stellt für



Bild 1: Der umfangreich ausgestattete Nextion-Editor

das Nextion-Display einen vollständigen Grafikeditor für den PC zur Verfügung, den man kostenlos von der Homepage [3] herunterladen kann. In diesem Editor platziert man alle Funktionselemente und Grafiken nach dem "What you see is what you get"-Prinzip. Ganz einfach können so Bedienelemente, Hintergrundbilder, Slider, Texte und verschiedene Screens definiert werden. Das verkürzt die Entwicklungszeit enorm. Die Funktion lässt sich sogar ohne Hardware komfortabel am PC simulieren. Sind die einzelnen Bildschirminhalte mit Ihren Bedienelementen einmal fertiggestellt, so werden diese in ein *.tft File kompiliert. Dieses File wird dann auf eine leere microSD-Karte kopiert. Danach schiebt man die microSD-Karte in den entsprechenden Slot am ausgeschalteten Display. Nachdem das Display an die Versorgungsspannung von 5 V angeschlossen ist, startet das Update automatisch. Ist dieses abgeschlossen, wird die Versorgungsspannung wieder abgeschaltet und die microSD-Karte entfernt. Wird das Display wieder an die Versorgungsspannung ange-



Bild 2: Die in ein dekoriertes Gehäuse eingebaute Uhr wird von einem externen 5-V-Netzteil versorgt.

schlossen, so wird der programmierte Inhalt sofort angezeigt.

Die Grafiken selbst habe ich mit der freien Software "Inkscape" erstellt.

Bild 1 zeigt die Entwicklungsumgebung mit den umfangreichen Funktionen.

Bild 3: Ein Tablet-Halter dient als Tischaufsteller für die Uhr.

Der Clou an der ganzen Sache ist, dass dieses Display über nur vier Anschlüsse verfügt: GND = schwarzes Kabel, +5 V = rotes Kabel, TX=blaues Kabel und RX= gelbes Kabel. Über die beiden serielle Leitungen RX und TX kann damit das Display mit dem Pretzelboard über einfache serielle Kommandos gesteuert werden.

Grundfunktionen

Bild 2 zeigt die Uhr, eingebaut in ein formschönes Gehäuse und über ein 5-V-Netzteil mit Strom versorgt. Beim Einbau des Displays ist zu beachten, dass das Display nicht ganz am Gehäuse anliegt, da sonst die Funktion des Touchscreens beeinträchtigt ist.

Die Uhr selbst steht auf einem Tablet-Halter, der im Elektronikfachhandel in verschiedensten Ausführungen verfügbar ist (Bild 3).

Die NTP-Multifunktionsdisplay-Uhr verfügt über die folgenden Funktionen:

- Einen Begrüßungs-Bildschirm, der für 5 Sekunden angezeigt wird und leicht durch Austausch der Grafik angepasst werden kann (Bild 4).
- Eine grafische Anzeige, wie man sie von der bekannten ClockTwo her kennt, inklusive einer Datumsanzeige sowie Statusinformationen, ob die Uhr erfolgreich mit dem WLAN-Netz verbunden ist und die NTP-Synchronisation geklappt hat (Bild 5).
- Tippt man einmal auf den Bildschirm, so ändert die Uhr ihr Aussehen und zeigt die Uhrzeit auf einer Anzeige, die den alten Nixie-Röhren nachempfunden ist (Bild 6). Die Statusinformationen, ob die Uhr erfolgreich mit dem WLAN-Netz verbunden ist und die NTP-Synchronisation geklappt hat, sind ebenfalls wieder vorhanden.
- Ein weiterer Touch auf das Display zeigt ein Einstellungsmenü (Bild 7), in dem die Anzeigenhelligkeit sowie eine Zeit eingestellt werden kann, nach der das Display abgeschaltet wird, um Strom zu sparen.

Wenn der Bildschirm abgeschaltet ist, reicht ein Touch, um den zuletzt angezeigten Inhalt wieder zum Vorschein zu bringen. Es ist natürlich auch möglich, die Anzeige dauerhaft eingeschaltet zu lassen. Die Hometaste speichert die Einstellungen dauerhaft und führt zurück zur ersten Uhrenanzeige.

Hardware

Die gesamte Schaltung besteht nur aus dem Pretzelboard sowie dem Nextion-Display. Da das Display vom Pretzelboard über eine software-emulierte serielle Schnittstelle (SoftwareSerial) angesprochen wird, ergibt sich die folgende Verdrahtung:

- GND (schwarzes Kabel) des Nextion-Displays an GND des Pretzelboards (Pin GND)
- +5 V (rotes Kabel) des Nextion-Displays an 5 V des Pretzelboards (Pin +5 V)
- TX (blaues Kabel) des Nextion-Displays an Pin D2
- · RX (gelbes Kabel) des Nextion-Displays an Pin D3

Damit ist auch schon der Hardwareaufbau erledigt, wenn man die Schaltung direkt von der Micro-USB-Buchse des Pretzelboards aus mit Strom versorgen



Bild 4: Der Begrüßungsbildschirm der Uhr nach dem Einschalten



Bild 5: Die Uhrenanzeige 1 ist in Form der bekannten WordClock gestaltet.



Bild 6: Die Uhrenanzeige 2 in der Form einer klassischen Nixie-Anzeige



Bild 7: Das Einstellungsmenü für das Display



Bild 8: Der Hardware-Aufbau ist sehr einfach – gerade einmal vier Leitungen zwischen Pretzelboard und Display sowie eine zusätzliche Stromversorgungsbuchse stellen die ganze Verdrahtung dar. Oben im Bild ist die Schirmabdeckung für die Displayrückseite zu sehen.

will. Ich habe mich für eine stabilere Variante mit einem Hohlstecker entschieden, den ich noch mit einem Elektrolyt- und einem Abblockkondensator versehen habe. Da das Pretzelboard leider über keine Befestigungsbohrungen verfügt, habe ich dieses im Gehäuse mit Heißkleber befestigt. Das Display ist im Gehäuse über schraubbare Kunststoffabstandshalter montiert.

Während der Inbetriebnahme ergab sich leider ein anderes Problem: Durch die Enge des Gehäuses haben die Komponenten des Nextion-Displays (Schaltregler?) den WLAN-Empfang gestört. Deshalb habe ich aus einem Stück kupferkaschierter Platine eine Abschirmung angefertigt, die mit Masse verbunden ist und mittels einer Selbstklebefolie über der Kupferfläche Kurzschlüsse verhindert.

Bild 8 zeigt das geöffnete Gerät inklusive Verdrahtung, Befestigung und der Abschirmplatine.

Software

So einfach der Aufbau der Hardware auch ist, die Softwareerstellung gestaltete sich entsprechend komplex, um alle Funktionen in den begrenzten Speicher des Pretzelboards unterzubringen. Folgende Grundfunktionen sind implementiert:

- · Initialisierung der Hardware
- · Anzeige des Startbildschirms
- · Anzeige des ersten Uhrenbildschirms
- Verbindung mit dem WLAN
- Verbindung mit dem NTP-Server und Synchronisierung der Uhrzeit
- Laufende Aktualisierung der Anzeige in einer Schleife jede Minute (ClockTwo-Display) oder jede Sekunde (Nixie-Display)
- Überwachung des Touchscreens auf Berührung und Umschaltung auf andere Anzeigen
- NTP-Resynchronisierung ein Mal pro Tag in der Nacht

Die Software wurde mit der Arduino IDE in der Version 1.6.5 erstellt und kompiliert. Für die Ansteuerung des Nextion-Displays steht neben einer recht komplexen Bibliothek auch eine einfache Bibliothek zur Verfügung [4], die in das "libraries"-Verzeichnis kopiert werden muss. Weiterhin habe ich die Time-Bibliothek [5] verwendet, die umfangreiche und nützliche Funktionen für die Zeitsynchronisation und die Uhrzeit bietet. Da es nicht auszuschließen ist, dass es inzwischen Updates gibt, die zu einer fehlerfreien Kompilation führen, sind die von mir verwendeten Bibliotheken unter [6] ebenfalls verfügbar.

Nach Aufruf des Programms Nextion_Clock.ino [6], der Auswahl des Arduino-Nano-Boards und der korrekten Schnittstelle kann dieses kompiliert und auf das Pretzelboard übertragen werden.

Sowohl das Nextion-Display als auch der auf dem Pretzelboard verbaute WLAN-Chip ESP8266 werden über jeweils eine software-emulierte serielle Schnittstelle (SoftwareSerial) mit den Namen "nextion" und "esp8266" angesprochen. Damit ist es möglich, die serielle Hardwareschnittstelle für die Ausgabe von Debuginformationen zu nutzen. Stellt man die Baudrate im "Serial Monitor" der Arduino IDE auf 19.200 Baud, kann man schön sehen, wie sich das Pretzelboard zuerst mit einem WLAN verbindet und dann die NTP-Zeit synchronisiert. Die Ausgabe kann mit dem Schalter #define DEBUG false am Anfang des Programms abgeschaltet werden. Damit sich das Pretzelboard mit dem WLAN in der jeweiligen Umgebung richtig verbindet, sind die Informationen unter #define SSID "xxx" und #define PASSWORD "xxxxx" richtig einzutragen!

Das größte Problem war der dynamische Speicher, der während der Entwicklung dauernd auszugehen drohte. Der Grund liegt darin, dass die Arduino-Umgebung alle Strings, die über die serielle Schnittstelle auszugeben sind, im Arbeitsspeicher ablegt. Daher wurde zu Beginn des Programms eine geänderte Print-Routine *SUB_STREAM_PRINT_PROGMEM* implementiert, die die Meldungen aus dem Flash-Speicher heraus ausgibt. Das Gleiche gilt für die Ausgabe von Strings auf dem Nextion-Display über die Routine *SUB_NEXTION_SEND*, die eine recht obskure Funktion mit dem Namen "FlashStringHelper" verwendet. Informationen dazu findet man im Internet.

In der "setup"-Routine werden zuerst die Nextion-, ESP8266- und seriellen Schnittstellen initialisiert. Danach wird die gespeicherte Helligkeit sowie das Timeout für die Abschaltung des Displays aus dem EEPROM ausgelesen und eingestellt. Nach 5 Sekunden wird statt des Begrüßungs-Bildschirms die Uhrenanzeige 1 am Nextion-Display aufgerufen. Danach versucht die Software in einer Dauerschleife, sich mit dem WLAN zu verbinden. War dies erfolgreich, wird in einer weiteren Dauerschleife versucht, die NTP-Zeit zu ermitteln. Ist dies gelungen, wird die Uhrzeit am Nextion-Display eingestellt und über die serielle Schnittstelle ausgegeben. Danach startet das Timeout für die Abschaltung des Displays.

In der Hauptschleife fragt das Programm zuerst den Touchscreen ab, ob dieser gedrückt wurde. Je nach Zustand wird das Display aus dem Sleep-Modus aufgeweckt oder eine neue Anzeige aufgerufen. Bei jeder neuen Sekunde (Uhrenanzeige 2) oder Minute (Uhrenanzeige 1) erfolgt eine Aktualisierung der Anzeige am Nextion-Display. Jeden Tag um 04:02 Uhr versucht die Software, in maximal drei Versuchen die Uhr-



- [1] Nextion Touchdisplays: https://www.itead.cc/wiki/Nextion_HMI_Solution
 [2] Nextion NX4827T043 011R:
- https://www.itead.cc/wiki/NX4827T043
- [3] Nextion-Editor: Download unter https://nextion.itead.cc/download.html Nextion-Editor Quick Start Guide: Download unter https://www.itead.cc/wiki/Nextion_Editor_Quick_Start_Guide
- [4] A simple arduino library for the Itead Nextion HMI: https://github.com/bborncr/nextion
- [5] Arduino-Time-Bibliothek: http://playground.arduino.cc/Code/Time http://www.pjrc.com/teensy/td_libs_Time.html https://github.com/PaulStoffregen/Time
- [6] Download der Projektdateien: www.elv.de: Webcode #10062

zeit erneut mit der NTP-Zeit zu synchronisieren. Alle 10 Minuten erfolgt eine Überprüfung, ob die WLAN-Verbindung nach wie vor besteht. Falls dies nicht der Fall ist, wird ein neuer Verbindungsversuch unternommen.

Die Unterroutinen sind in die Rubriken Basisroutinen, ESP8266 WLAN, NTP und Display unterteilt. Die Rubrik Basisroutinen enthält Routinen zum Senden und Empfangen von Informationen zum/vom WLAN-Chip ESP8266 und zum Senden von Informationen an das Nextion-Display. Die Rubrik ESP8266 WLAN enthält die Routine zum Verbinden mit einem WLAN, bei der auch die SW-Version des ESP8266, der WLAN-Name sowie die zugeteilte IP-Adresse über die serielle Schnittstelle ausgegeben werden. Die Rubrik NTP enthält alle Routinen, um die NTP-Zeit abzufragen: Dazu wird zuerst ein NTP-Server über eine DNS-Abfrage ermittelt, danach dieser NTP-Server abgefragt, um die korrekte Zeitzone zu ermitteln und danach nochmals zur Ermittlung der korrekten Zeit. Die Rubrik Display dient dazu, die aktuelle Zeit- und Datumsinformation auszugeben. Die Routine SUB_CLOCK_ PRINT gibt die formatierte Information über die serielle Schnittstelle aus. Die Routine SUB_CLOCK_ CLEAR löscht alle Segmente der Uhrenanzeige 1. Die Routine SUB_CLOCK_1_DISPLAY zeigt die aktuelle Zeit- und Datumsinformation für die Uhrenanzeige 1 an. Im Prinzip werden immer einzelne Grafikelemente ein- oder ausgeblendet. Die Routine SUB_CLOCK_2_ DISPLAY erledigt das Gleiche für die Uhrenanzeige 2. Die Routine SUB_CONFIG_MENU schließlich ist für das Einstellungsmenü zuständig.

Der ausführlich dokumentierte Code sollte es möglich machen, eigene Ideen umzusetzen oder Änderungen durchzuführen und sich mit der Ansteuerung des Nextion-Displays vertraut zu machen.

Unter [6] findet man folgende Informationen zum Projekt:

- Arduino-Software "Nextion_Clock.ino"
- Arduino-Nextion-Bibliothek
- · Arduino-Time-Bibliothek
- Grafikfile f
 ür den Nextion Editor "Uhr.HMI" und kompilierte Version "Uhr.tft"

Ich wünsche viel Spaß beim Nachvollziehen des Projekts!



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 verhelfen 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 ELV Journal mit Nennung des Namens vorgestellt.





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.