

Vorgeschichte

Es war im letzten Jahr. Kurz vor Weihnachten. Der Trafo einer Halogenlampe hatte den Geist aufgegeben, Ersatz musste her, und so war ich mit meiner Frau beim Baumarkt in der Lampenabteilung gelandet.

Wir standen schon an der Kasse, da fiel ihr plötzlich ein, dass sie ja noch „ein paar Schaltuhren“ für die alljährliche Lichterketten-Orgie bräuchte...

Es gab zwei Sorten, mechanische und elektronische. Meine Frau griff sich eine handvoll von den ersteren, und gerade wollte ich energisch einwenden „wenn schon, dann aber die anderen“, da erinnerte ich mich daran, dass ich vor Jahrzehnten schon einmal eine elektronische Schaltuhr im Einsatz, und eine enorme Abneigung gegen die Art ihrer Programmierung entwickelt hatte.

Da konnte man irgendwelche Ein- und Ausschaltzeiten programmieren, Wochentage einstellen, das ganze für 6 unabhängige Timer – es war ziemlich verwirrend, und ohne die Möglichkeit der Zufallsprogrammierung wäre ich aufgeschmissen gewesen...

So waren es halt drei oder vier **mechanische** Schaltuhren. Als ich meiner Frau erklärte, wie man die „programmiert“, war ich von der super-simplen Bedienung selbst überrascht:



Schaltuhr mit "Nippel-Rad"

48 im Kreis angeordnete Plastik-Nippel, die man runterdrücken und hochziehen kann, bestimmen für jede halbe Stunde den Schaltzustand – Nippel oben bedeutet „AUS“, Nippel unten „EIN“. In die Steckdose stecken, die richtige Uhrzeit an der mit einem Pfeil gekennzeichneten Position durch Drehen des „Nippel-Rads“ einstellen – fertig. Dann gibt es noch einen „Zwangs-EIN-Schiebeschalter“, damit man leicht die Funktion des angeschlossenen Verbrauchers testen kann.

Na gut, die 30-Minuten-Auflösung ist nicht berauschend und das ist auch keine Wochen- sondern eine Tages-Schaltuhr, was aber für den Betrieb von Lichterketten so gut wie ohne Bedeutung war...

Von Bedeutung hingegen war dieses Erlebnis für mich, denn es war die „Initial-Zündung“ für das Projekt

Komfortable elektronische mehrfach Wochen-Schaltuhr

Die elektronischen Schaltuhren haben ein Mini-Display, auf dem die Ein- und Ausschaltzeiten der Timer bequem dargestellt werden können. Eine dem „Nippel-Mechanismus“ nachempfundene Programmiermethode ist auf so einem Mini-Display schwer zu machen. Aber - wer redet von einem Mini-Display? Die hier beschriebene Schaltuhr hat gar kein Display!

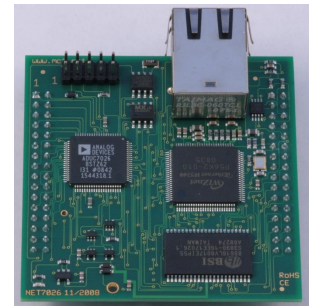
Wie sie programmiert wird? Über's LAN!

Das kann man mit einem Computer machen. Aber - wer redet von einem Computer? Im (w)LAN tummeln sich heutzutage schon längst nicht mehr nur „normale“ Computer. Da wimmelt's von Netbooks, Tablets und Smartphones – alle mit „richtigen“ Displays ausgestattet. Und mit Browsern, die sogar Spezial-Software überflüssig machen.

„Now is the cat out of the sack“: Ein Main-Feature dieser Schaltuhr ist, dass sie eine IP-Adresse hat, und per Browser programmiert wird!

Hardware

Das Herzstück der Schaltuhr ist – wen wundert's – der Netzwerk-fähige Einplatinencomputer NET7026 von MCT Paul & Scherer (www.mct.de/product/net7026.html). Ein bisschen „Overkill“, OK. Überhaupt, die hier beschriebene Schaltuhr ist was für Technik-Freaks, und vom Kosten- und Aufbau-Aufwand her nicht konkurrenzfähig. Da gibt's Preiswerteres von Reichelt & Co. Das muss man allerdings „nehmen wie's kommt“, während die grenzenlose Flexibilität dieser Lösung zweifellos ihr Haupt-Vorteil ist.



NET7026



RELAIS8

Ebenfalls ein Standard-Produkt von MCT ist „RELAIS8“ (www.mct.de/product/relais8.html), eine 8-fach Relais-Karte, wie der Name schon sagt. Jedes Relais betätigt einen Einfach-Umschalt-Kontakt mit bis zu 12A. Ein Laboraufbau für 8 beliebige Schaltvorgänge ist damit ebenso machbar, wie die zweipolige Schaltung von vier 230V Verbrauchern mit je zwei Relais.

Software

Das Tolle an Netzwerk-basierten Lösungen ist, dass man die Software nicht für beide Seiten schreiben muss. Dafür wird in die Hardware einfach ein HTTP-Server eingebaut, so dass die andere Seite (PC, Smartphone usw.) mit Standard-Software – sprich einem Browser – auskommt. „HTTP-Server“ hört sich ja beeindruckend an, ist aber furchtbar simpel. Hier jedenfalls - mit dem NET7026 - denn da erledigt ein WIZnet-Ethernetcontroller eine Menge Arbeit „in Hardware“. Der hat nämlich den TCP/IP Stack „fest-verdrahtet“ im Bauch. Ein paar Zusatz-Funktionen, wie z.B. net_init(), net_read() und net_write() - fertig ist der Web-Server. Der liefert dem Browser eine HTML-Seite mit einem „Formular“, und der Browser schickt die – vom User eingegebenen – Formulardaten zurück.

Die Schaltuhr wird also mit dem Formular programmiert. Jetzt bietet sich die „Nippel-Methode“ an. Im übertragenen Sinn, denn für jede halbe Stunde gibt es ein Formular-Feld, in dem der Schaltzustand steht. Wenn ein Feld leer ist, entspricht das einem hochgezogenen Nippel, also „AUS“. Ein Nippel wird runtergedrückt, indem etwas in das Feld geschrieben wird, z.B. „Ein“.

Aber es soll ja eine Wochen-Schaltuhr sein, und außerdem gibt es nicht nur einen, sondern 8 Schaltzustände. Das ist nur mit „Super-Nippeln“ zu bewältigen, die so aussehen:

Jeder Nippel wird zunächst dreigeteilt, je ein Teil für „Montag bis Freitag“, „Samstag“ und „Sonntag“. Dann wird jeder der drei Teile noch einmal 8-fach unterteilt, für Relais1 bis Relais8. Man stelle sich eine solche Schaltuhr bildlich vor - pro halbe Stunde ein Nippel-Tripel mit 24 Sub-Nippeln...

Zum Glück haben wir es hier aber mit „Software-Nippeln“ zu tun. Auf das Formular übertragen heißt das nämlich:

Es gibt pro halbe Stunde drei Felder (eins für „Montag bis Freitag“, eins für „Samstag“, eins für „Sonntag“), und wir schreiben einfach die Nummern (von 1 bis 8) der Relais, die „EIN“ sein sollen, in das jeweilige Feld.

So sieht die HTML-Seite aus, die der Web-Server liefert...

Außerdem kann man das aktuelle Datum der Schaltuhr sehen (RTC) und bei Bedarf korrigieren (set), jedes der 8 Relais „Zwangs-EIN-Schalten“ (force 'on'), Änderungen seit dem letzten Absenden rückgängig machen (Undo), und alle Felder löschen (Reset). Die Zeit-Auflösung (delta) kann in drei Stufen gewählt werden (60, 30, 15 [Minuten]), und die Zeit-Tabelle lässt sich ein- oder zweispaltig anzeigen (display).



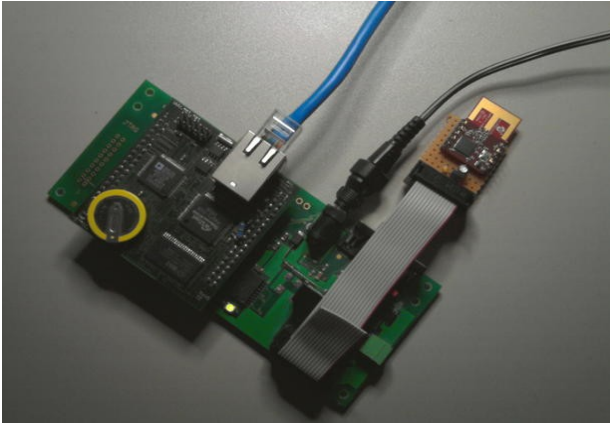
.. und so z.B. auf einem iPod

„Verlängerung“ per Funk

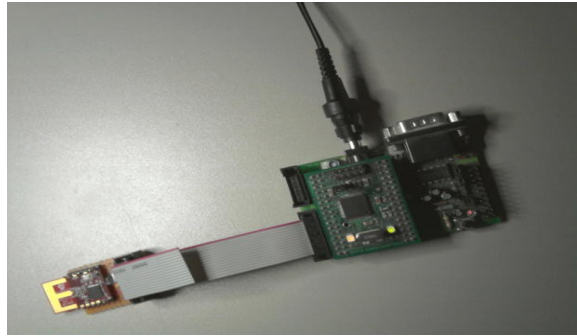
Die Schaltuhr kann nur per Kabel an das Netzwerk angeschlossen werden, muss sich also irgendwo in der Nähe des Routers befinden. Das gilt auch für die zu schaltenden Verbraucher, will man nicht über unhandlich lange Kabel stolpern. Eine wLAN-Anbindung würde der Schaltuhr zu mehr „Bewegungsfreiheit“ verhelfen und vielleicht gibt es von MCT ja bald ein Standard-Produkt dafür...

In der Zwischenzeit kann man die Kabel-Verbindung zwischen NET7026 und RELAIS8 durch eine ZigBee-Funkstrecke ersetzen, und dadurch auch Verbraucher schalten, die sich – je nach HF-Power – weiter weg vom NET7026 befinden, das in diesem Fall als LAN/ZigBee-Bridge fungiert. Denkbar wären auch mehrere ZigBee-Empfänger, die alle an einer IP-Adresse „hängen“.

Der ZigBee-Empfänger ist mit einem LC2138 aufgebaut (www.mct.de/product/lc2138.html) Er empfängt über das ZigBee-Modul ZB1 Daten vom NET7026 und steuert RELAIS8 entsprechend. Daten werden übertragen „wenn es nötig ist“, mindestens aber alle drei Sekunden (Heartbeat). Eine LED zeigt den Normalbetrieb an. Wird drei Sekunden lang nichts empfangen, fängt die LED an zu blinken und signalisiert so, dass die Datenverbindung aus irgendeinem Grund unterbrochen ist, z.B. durch Überschreitung der Reichweite.



Laboraufbau der LAN/ZigBee-Bridge...



... und des Empfängers ohne RELAIS8

Warum so ein Aufwand?

Wie gesagt. Etwas für Technik-Freaks.

Aber auch etwas für die, die für ihre konkrete Anwendung keine Lösung „von der Stange“ finden. Sonderwünsche? Vielleicht mehr als nur Schalten, z.B. die Rückmeldung von Schaltzuständen, oder auch mal die Übertragung von Analogwerten?

Die Hard- und Software kann leicht an spezielle Erfordernisse angepasst werden. Und das muss man nicht einmal selbst machen. MCT liefert nicht nur Standard-Produkte, die bieten auch maßgeschneiderte Individual-Lösungen an. Einfach mal mit denen reden...