

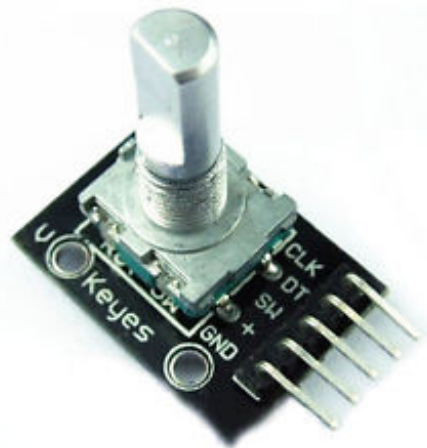
Die **Astro-Clock** ist eine kleine GPS-synchronisierte Uhr mit Weckfunktion, Anzeige von Innen- und Außentemperatur, sowie Darstellung einiger astronomischer Daten (Position und Phase von Sonne und Mond, Auf- und Untergangszeiten). Sie ist eine Ergänzung zur GPS-Außeneinheit, um deren Nutzung nicht ausschließlich auf die Synchronisierung und Positionsbestimmung der WSPR-Bake zu beschränken (siehe WSPR-Projekt auf meiner Webseite). Ursprünglich als kleine Spielerei und Geburtstagsgeschenk für meine Frau gedacht, hat sich die Programmierung nun doch auf beinahe 2000 Zeilen Code aufgebläht und den verwendeten Arduino Nano fast an seine Grenzen gebracht. Das Ergebnis ist ein, wie ich finde, ansprechendes Selbstbauprojekt, welches für eine nur einmalige Umsetzung einfach zu schade wäre. Großes Augenmerk wurde auf die Verwendung vieler im Onlinehandel preiswert verfügbarer Standardmodule gelegt, so dass trotz der Komplexität auch dem noch nicht so geübten Selbstbauer ein sicherer Aufbau gelingen sollte.

### **Software:**

Das Hauptprogramm sowie alle notwendigen Bibliotheken stehen in diesem Paket zum Download bereit. Ein Großteil der Software wurde von mir geschrieben. Wo angebracht, wurde auf fertige open-Source Bibliotheken zurückgegriffen. Teilweise enthalten diese auch anwendungsspezifische Anpassungen.

### **Benötigte Hardware:**

- ein Drehencoder mit integriertem Taster  
(Quelle: eBay)



- ein DS18B20 Temperatursensor



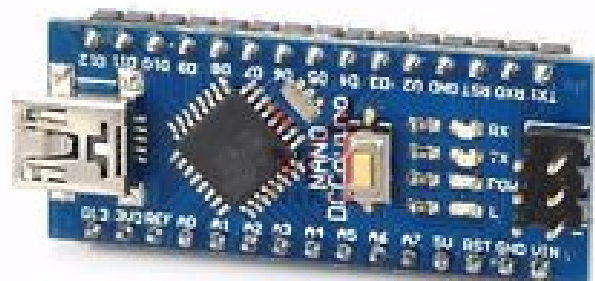
- ein 433MHz ISM Empfänger (wird oft im Paket mit Sender verkauft, welcher für die GPS-Außeneinheit benötigt wird / Quelle: eBay)



- ein 1,8“ TFT Display, basierend auf ST7735 (128x160 Pixel, Typ KMR-1.8 / Quelle: eBay)



- ein Arduino Nano oder entspr. Derivat (Quelle: eBay)



- ein Kleinstlautsprecher (in meinem Fall aus einer alten Wechselsprechanlage)
- diverse Kleinteile (Widerstände, Kondensatoren, ...)
- einen Photowiderstand (meine Quelle war: <https://www.ebay.de/itm/20-PCS-GL5528-5528-Fotowiderstand-ltr-Erfassung-Widerstande-Licht-abhangig-CCYE/401176239861?hash=item5d67f79ef5:g:zj0AAOSwHoFXvYl6>)

GL5528 5528 Photoresistor  
 Spectral peak: 540nm  
 Bright resistance: 10-20K  
 Dark resistance: 1M

gab's leider nur im 20er-Pack; gegen Erstattung des Portos kann ich gern noch welche abgeben)

- eine Spannungsversorgung (in meinem Fall ein altes Handyladegerät / 5V Stecker-Schaltnetzteil)

### **Schaltungsdetails:**

- Drehencoder:

„CLK“ und „DT“ sind die Ausgänge des Encoders, hinter „SW“ verbirgt sich der integrierte Taster. Auf meiner Platine sind für den Encoder 10k Pull-up Widerstände gegen + vorhanden. Sollte das nicht der Fall sein, sind diese extern vorzusehen, damit die restliche Schaltung (Entprellung) korrekt funktioniert. Die Werte der besagten Entprellschaltung sind relativ kritisch und sollten eingehalten werden (20% Abweichung sind sicher OK).

- DS18B20 Temperatursensor:

Auch hier ist auf den integrierten Pull-up zu achten. Ist der nicht vorhanden, bitte extern vorsehen. Werte von 4,7 bis 10k sind OK.

- Helligkeitssensor:

Die Größe des Serienwiderstandes hängt vom verwendeten Photowiderstand ab. Die Uhr hat eine Funktion zum Abgleich und der Einstellung des Tag/Nacht Schwellwerts integriert. Diese zeigt u.A. den Sensorwert in % an (0...99). Mit ihrer Hilfe kann man den Serienwiderstand so auslegen, dass an der Schwelle von Tag- zu Nachtmodus ca. 30...70% detektiert werden.

Der Wert des Kondensators ist unkritisch (10n...100n).

#### - Lautsprecher:

Wie erwähnt, wurde hier ein dynamischer Kleinstlautsprecher verwendet. Soll ein Piezo zum Einsatz kommen, so ist eine Ansteuerung in Kollektorschaltung natürlich ungeeignet, da der Piezo keine Möglichkeit zum entladen hätte. Für diesen Fall findet sich ein alternativer Schaltungsvorschlag auf Seite 3 des Schaltplans.

Ebenfalls untypisch für die Kollektorschaltung ist der Basisvorwiderstand von 1k. Durch negative Erfahrungen vorsichtig geworden, begrenzt dieser in meinen Schaltungen im Fehlerfall den Ausgangsstrom des Arduino und sichert sein Überleben. Werte von 470 Ohm bis 2k sind OK. Als Transistor eignet sich jeder Kleinstleistungstyp mit hinreichender Stromfestigkeit und Verstärkung ( $>200$ ). Da die Ausgänge des Arduino mit bis zu 40mA belastbar sind, kann man grundsätzlich den Lautsprecher (mit einem Vorwiderstand) auch direkt anschließen. Allerdings wollte ich diese Möglichkeit nicht austesten. Zumindest für einen Piezo stellt dies aber sicher eine weitere mögliche Schaltungsalternative dar.

#### - 433MHz Empfänger:

Auch hier ist auf den 10k Pull-up zu achten. Ohne diesen zeigte zumindest mein Modul etwas undefinierte Ausgangspegel.

#### - TFT-Display:

Das Widerstandsnetzwerk ist notwendig, um von den 5V des Arduino auf 3,3V zu konvertieren. Jede Kombination, die ein solches Teilungsverhältnis ergibt, ist geeignet. Die absoluten Werte sind unkritisch, allerdings sollten die Widerstände gegen Masse sich im Bereich 1k ... 10k bewegen. Im Internet kursieren auch Schaltungen, welche auf eine Anpassung völlig verzichten bzw. lediglich Reihenwiderstände vorsehen. Mag sein, dass einige Module ein solches Vorgehen tolerieren, aber ich habe dies weder getestet noch möchte ich es empfehlen. Die Ansteuerung der Hintergrundbeleuchtung erfolgt, wie beim Lautsprecher, mit einer Kollektorschaltung und es gelten die gleichen Randbedingungen.

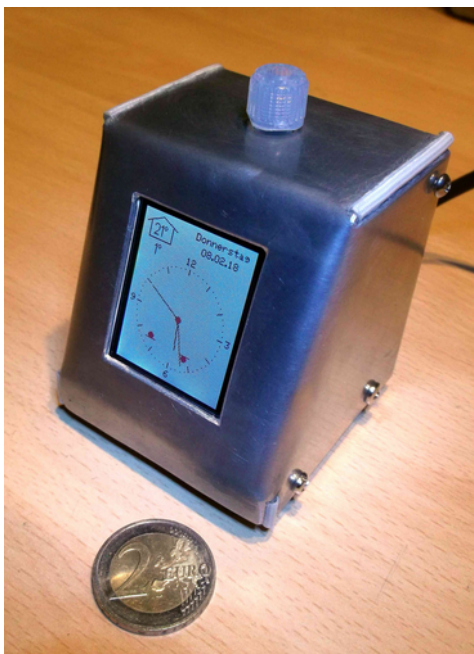
#### - Stromversorgung:

Grundsätzlich kann die Schaltung den auf der Arduino-Platine integrierten Längsregler nutzen. Dann erfolgt die Speisung über den Anschluß „Vin“. Um den Regler nicht zu überlasten, sollte die Eingangsspannung zwischen 7 und 8V liegen. Alternativ kann man auch direkt auf die 5V-Ebene speisen, wobei dann eine sauber stabilisierte Spannung vorausgesetzt werden muss. Oft finden sich noch alte Stecker-Schaltnetzteile mit stabiler 5V Ausgangsspannung, welche zum laden von Mobiltelefonen verwendet wurden.

Der 220 $\mu$ F Elko ist in jedem Fall zu empfehlen, um Stromspitzen sauber abzapfen. Die Entstörung mittels der 2x100 $\mu$ H Drossel mag etwas überzogen erscheinen, allerdings bin ich als Funkamateur auch diesbezüglich recht sensibel (und der

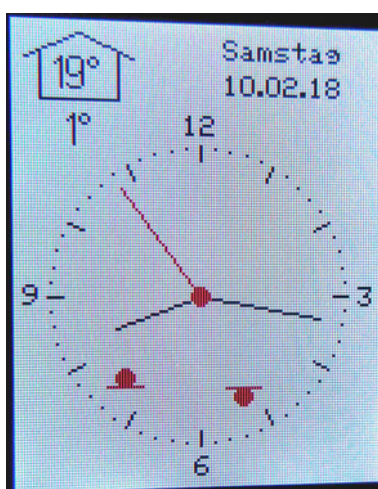
Arduino erzeugt eine durchaus nennenswertes Störspektrum). Von Fall zu Fall kann man hier auch etwas kleinere Geschütze auffahren. **Apropos Störnebel, eine Montage des ISM Empfängers direkt neben dem Arduino/Display schmälert in jedem Fall dessen Empfindlichkeit. Sollte der Sender nicht weiter als 10m entfernt sein, geht das in Ordnung. Ich habe mich aber dafür entschieden, sowohl Empfänger als auch Temperatursensor ca. 20cm entfernt am Stromversorgungskabel zu befestigen (Schrumpfschlauch) und die Datenleitungen zusammen mit der Betriebsspannung ins Gehäuse zu führen (altes USB- oder Telefonkabel). Dies kommt auch einer korrekten Temperaturmessung entgegen. Die Verlustwärme der Schaltung ist gering, reicht aber im geschlossenen Gehäuse aus, um die Temperaturmessung nennenswert zu verfälschen.**

### **Bedienung:**

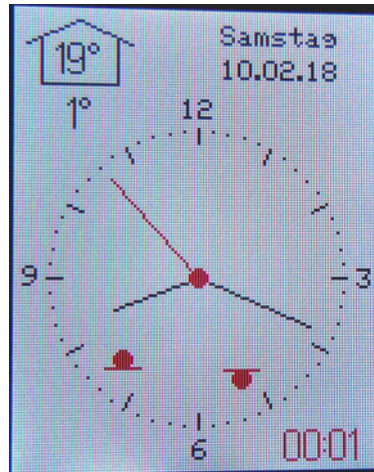
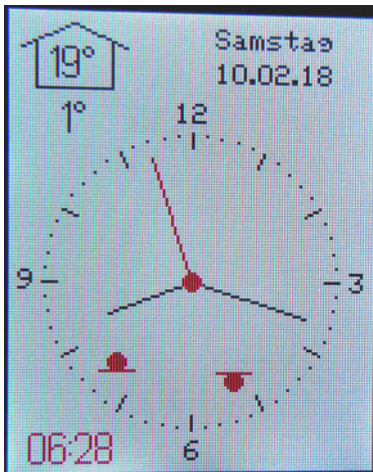


Hier ist die fertige Uhr zu sehen. Der Bedienknopf mit Encoder und Taster befindet sich oben. Wird er während des Startens der Uhr gedrückt gehalten, so werden im Display die gemessene Umgebungshelligkeit sowie der Schwellwert für die Tag/Nacht – Umschaltung angezeigt. Mit dem Encoder kann dieser Grenzwert verändert und durch Druck des Tasters gespeichert werden. Danach gelangt man zur Eingabe der Helligkeit im Nacht- bzw. Tagmodus, der Zeitzone sowie der Sommer-/Winterzeitumschaltung, wo entsprechend zu verfahren ist.

Ist die Uhr hochgefahren, so kann durch drehen des Knopfes zwischen mehreren Menüs gewechselt werden.



Der Hauptbildschirm in Tag- und Nachtdarstellung. Zu sehen sind Datum, Uhrzeit, Außen- und Innentemperatur, sowie die Zeiten von Sonnenauf- und Untergang (im Ziffernblatt).

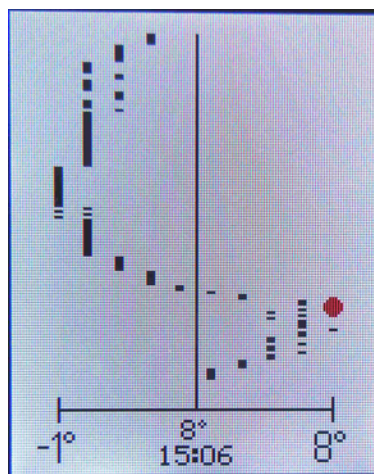
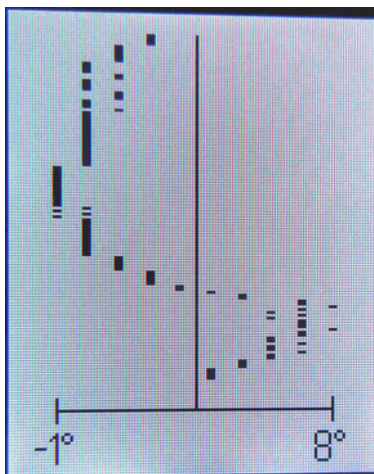


Der Hauptbildschirm mit aktiviertem Wecker (links) bzw. Kurzzeitwecker (rechts). Zur Auswahl des jeweiligen Timers ist der Bedienknopf kurz zu drücken. Wird er gedrückt gehalten, so kann der Timerwert (Weckzeit bzw. Countdown-Zeit) verändert werden (Ziffern blinken). Ein nochmaliges Drücken speichert den gewählten Wert

und aktiviert den Alarm. Die Darstellung der Ziffern wechselt dabei zu Rot. Die Weckzeit wird nicht flüchtig gespeichert und bleibt auch nach ausschalten der Uhr erhalten.

Die Deaktivierung erfolgt sinngemäß, also durch gedrückt halten des Kopfes nach vorheriger Timerauswahl (kurzer Knopfdruck).

Wurde der Alarm ausgelöst, so kann durch drehen des Bedienknopfes ein Stummschalten mit erneutem Alarm nach 5 Minuten bewirkt werden. Das Drücken des Bedienknopfes schaltet den Alarm ab. Im Falle des Wecker würde dieser dann erst in ca. 24h, also bei erneutem Erreichen der Weckzeit wieder auslösen.



In diesem Menü wird der Verlauf der Außentemperatur über 24h dargestellt. Der oberste Wert entspricht dabei dem Aktuellen. Durch drücken des Knopfes kann ein Cursor ein- bzw. ausgeblendet werden. Der zugehörige Wert (Temperatur und Uhrzeit) wird unten eingeblendet. Der Cursor kann durch drehen des Knopfes bewegt werden.



Hier erfolgt die Abbildung der aktuellen Positionen von Sonne und Mond sowie deren Auf- und Untergangszeiten für die jeweiligen Koordinaten, an denen sich die Uhr befindet. Mondphase und Sichelneigung werden ebenfalls berechnet und (soweit es die Auflösung des Displays zulässt) astronomisch korrekt dargestellt. Bezüglich der Elevation wurde von der üblichen Darstellung als Projektion auf eine gedachte Ebene abgewichen, da dies wenig intuitiv erschien. Vielmehr entspricht die Position des Symbols hier direkt der beobachteten Erhebung, also  $90^\circ$  für den Fall, dass das Symbol im Zentrum steht,  $0^\circ$  falls es am äußeren Kreis steht (Horizont) und  $45^\circ$  für die halbe Strecke

zwischen diesen.

Ist das jeweilige Objekt untergegangen, so wird das Symbol durch den Buchstaben „S“ bzw. „M“ ersetzt.