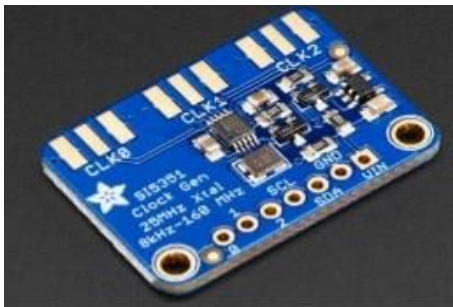


## Quarzersatz durch Digitalen Oszillator

Ein altes 150 Mhz- Funkgerät Telecar TE 160 soll auf die APRS- Frequenz 144,800 Mhz umgestellt werden. Die Sendefrequenz wird durch Verdopplung der Quarzoszillator- Frequenz und Mischung mit der ZF- Frequenz 16,9 Mhz erzeugt. Für die APRS- Frequenz 144,8 Mhz wird hier ein Quarz mit 63.950 Mhz benötigt.

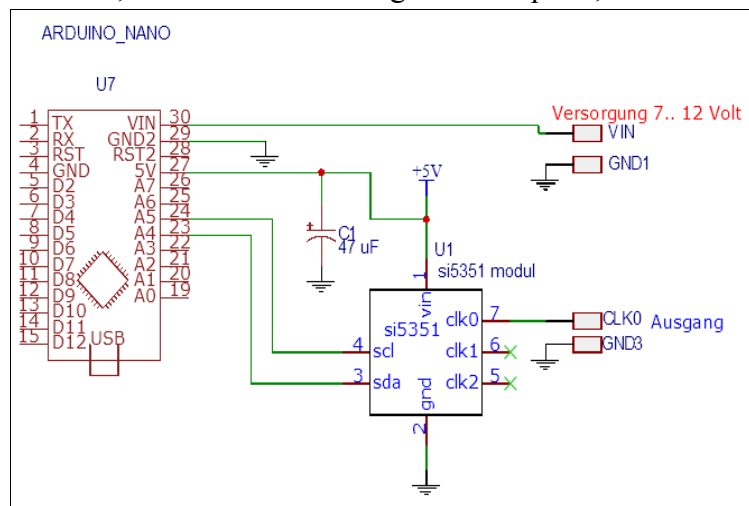


Quelle: [www.adafruit.com](http://www.adafruit.com)

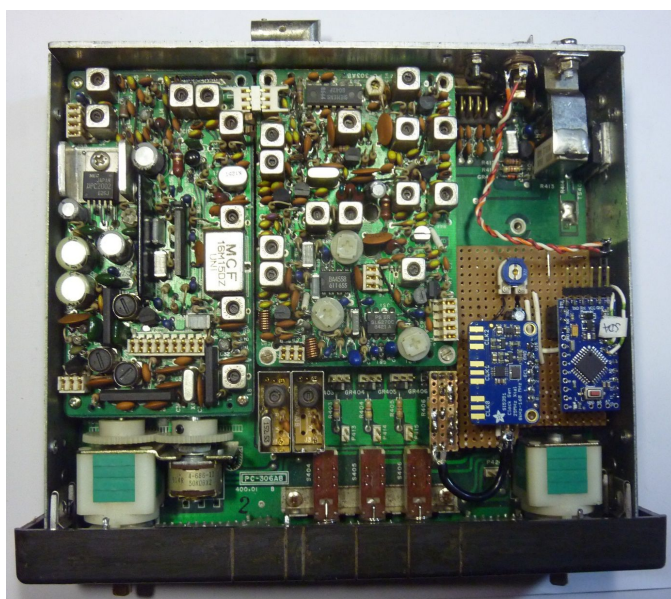
Programmieren und zur Kommunikation wird noch ein USB-Serial-Wandler benötigt. Man kann auch ein 'Arduino-Nano-Modul' verwenden, da ist der USB-Serial-Wandler schon drauf. Das ganze kann mit 7 bis 12 Volt betrieben werden. Beide Module waren von früheren Experimenten schon vorhanden. Die Kosten liegen bei 12 bis 15 Euro.

Warum einen speziellen Quarz kaufen ?

Nach einigen Experimenten[1] mit dem Digitalen Oszillator Chip Si5351( Datenblatt [2] ) wurde beschlossen , einen Aufbau mit diesem Oszillator zu probieren. Das Oszillator Chip ist in SMD-Bauweise, jedoch gibt es von verschiedene Herstellern [3][4] Module, die gut zu handhaben sind. Das Chip wird über eine serielle I2C-Schnittstelle programmiert; man benötigt also noch eine Controller zum 'Booten' des Chips. Hier haben wir uns für ein 'Arduino-Mini-Modul' [5] entschieden, welches einen Atmega 328 Chip hat; zum



Um dieses Modul nun im Telecar einsetzen zu können, wurde das Tonruf-Auswertemodul



des Telecar-TE 160 ausgebaut und durch eine Lochrasterplatine ersetzt, auf dem der SI5351 und der Arduino-Mini aufgesteckt wurden. Die Stromversorgung geschieht über die Steckverbinder der Lochrasterplatine zum Telecar. Das Oszillatorsignal CLK0 wird über ein kurzes Stück Koaxkabel an eine zweite Lochrasterplatine geführt, welche die Quarzsteckplatine ersetzt.

Über eine Programmerweiterung lassen sich auch mehrere Frequenzen mit Hilfe des Kanalschalters abrufen, was aber für den Anwendungszweck als APRS- Transceiver nicht notwendig ist.

### Programmierung:

Der Arduino Nano wird über USB an den PC angeschlossen. Bei Verwendung eines Arduino Mini wird zusätzlich ein USB-Serial-Adapter zur Kommunikation benötigt.

Die Programmierung erfolgt über die Arduino- Oberfläche[6]; durch Verwendung frei zur Verfügung stehender Softwarebibliotheken ist die Programmierung sehr einfach. An dieser Stelle soll nicht auf Einzelheiten der Software eingegangen werden[7].

Die Frequenz wird über ein Terminal z.B. Hyperterminal eingegeben, so dass der Benutzer so wenig wie möglich mit der Prozessor-Programmierung zu tun hat.

Die einmal eingestellte Frequenz wird im EEPROM gespeichert; beim Neustart des Prozessors wird mit der gespeicherten Frequenz 'gebootet'. Der USB-Serial-Adapter wird später nicht mehr benötigt.

### Ausblick

Wer es ganz klein haben will, kann statt eines Arduino- Moduls einen ATTINY-85 als 8-Pin-DIL-Prozessor nehmen; die Software ist dafür vorgesehen. Vom Preis her hat das jedoch keine Vorteile, da die Arduino-Nano- Module ab 2 Euro in China zu kaufen sind. Der digitale Oszillator kann etwa von 150 kHz bis 120 Mhz programmiert werden und kann drei verschiedene Frequenzen gleichzeitig erzeugen. Interessante aktuelle Anwendungen findet man beim QRP-Labs QCX-Transceiver[8] und beim Projekt uBitx- Transceiver[9].

### Referenzen:

- [1] Experiment : <http://dk2jk.darc.de/arduino/swt2016/digitaler-oszillator.html>
- [2] Datenblatt : <https://www.silabs.com/documents/public/data-sheets/Si5351-B.pdf>
- [3] Si5351 Modul : <http://www.qrp-labs.com/synth.html>
- [4] Si5351 Modul : <https://www.adafruit.com/product/2045>
- [5] Arduino Mini : <https://www.adafruit.com/product/2378>
- [6] Arduino : <https://www.arduino.cc>
- [7] Software zum Projekt : [http://dk2jk.darc.de/arduino/index\\_arduino.html#quarzersatz](http://dk2jk.darc.de/arduino/index_arduino.html#quarzersatz)
- [8] Qrp\_Labs : <http://www.qrp-labs.com/qcx.html>
- [9] uBitx : <http://www.hfsignals.com/>

Januar 2018

Heribert Schulte, DK2JK

Günter Schweppe , DK5DN