

## Teil 2

# Bedienung der SDR-Console ohne Maus und Tastatur

Sandor Dubronner, DM4DS

Wie schon im Journal Nr. 3 Jg.47 zu lesen war, inspirierte mich die Erfahrung von Charly, DK3ZL, während seiner Namibia-DXpedition mit dem ELAD TMate2. Dies ist eine externe Bedieneinheit, die alle wichtigen Funktionen der SDR-Console steuern kann, auch wenn diese nicht im Vordergrund ist bzw. man sich gerade im Logbuch oder einer anderen Anwendung befindet. Jedoch wollte ich selbst etwas bauen und nach einem Blick in die Steuermöglichkeiten der SDR-Console war schnell klar, wohin die Reise gehen würde.



Bild 1: Externe Bedieneinheit für die SDR-Console

**D**a in der SDR-Console von Simon Brown die ELAD TMate2-Konsole bereits als auswählbare Option vorhanden war, war mein erster Gedanke, deren Protokoll zu nutzen. Leider konnte ich zum damaligen Zeitpunkt keine ausreichende Dokumentation zur TMate2 finden. Da die SDR-Console (nachfolgend SDRC) aber sowohl CAT als auch MIDI spricht und beide Protokolle sehr gut dokumentiert sind, hatte ich die Qual der Wahl.

CAT (Computer Aided Tuning) wird meist zur Kommunikation eines Programms (z.B. eines Logbuchs) mit einem Transceiver oder mit einem Rotor über eine RS232 (serielle) Schnittstelle genutzt. Über CAT lassen sich z.B. die

aktuelle Frequenz abfragen oder welche Modulationsart gerade aktiv ist, aber auch Zustände setzen. CAT ist also bidirektional (Senden und Empfangen) und kommt daher für mein Vorhaben infrage.

Simon Brown hat sich bei seiner Implementierung in der SDRC an das recht bekannte Format des Kenwood TS-2000 gehalten. CAT funktioniert vereinfacht ausgedrückt wie folgt: Es wird ein kurzer Befehl, bestehend aus zwei Buchstaben und ggf. einer Zahl, per COM-Schnittstelle an die SDRC gesendet und diese antwortet mit einem „String“. Die verfügbaren Befehle finden sich zur Übersicht in der **Tabelle**. CAT eignet sich somit sehr gut, um Zustände wie Einstellungen

abzufragen bzw. diese zu setzen. Möchte ich wissen, welche QRG gerade aktiv ist, sende ich ein „FA;“ und die SDRC wird entsprechend antworten. Möchte ich eine bestimmte Frequenz einstellen, sendet man diese mittels absoluter Angabe z.B. „FA10489704000“ an die SDRC. Will man einen kontinuierlich abstimmbaren VFO realisieren, muss man beispielsweise permanent einen Rotaryencoder abfragen und die doch relativ lange Frequenz immer wieder einzeln an die SDRC übermitteln, nachdem der absolute Wert zuvor auf dem Arduino mit der eingestellten Sprungweite ausgerechnet wurde. Dies summiert sich gerade beim VFO schon zu erheblichen Datenmengen und Rechenaufgaben und ist in der Praxis so langsam, dass sich damit einfach kein flüssiges „übers Band“ drehen bewerkstelligen lässt. Die SDRC mittels CAT auf Sendung zu bringen und zurück auf Empfang zu schalten ist hingegen eine Leichtigkeit. Wie so oft: Gut für das Eine, schlecht für das Andere.

Da es primär ja um Funktionen wie VFO, Lautstärkereger, Empfangsbandbreitenfilter und nicht zu vergessen um eine PTT geht, stellte sich CAT nur als zweite Wahl heraus. Somit überlassen wir CAT dem Logbuch oder anderen Programmen um die SDRC zu steuern. Was stattdessen?

## Die Alternative heißt MIDI

MIDI „arbeitet“ anstatt mit Absolutwerten, die jedes Mal berechnet werden müssen, mit relativen Angaben. Dies ist nur die halbe Wahrheit, aber nehmen wir dies der Einfachheit halber nun einmal so an.

Dies bedeutet im Falle eines Lautstärkereglers, dass ein Befehl „lauter“ oder „leiser“ ist, anstatt exakt anzugeben wie laut oder leise die Einstellung sein soll. Unsere Fernbedienung vom Fernseher macht es übrigens genauso.

Bei der Implementierung eines VFOs wird bei MIDI anstelle eines langen Absolutwertes bei CAT z.B. 9-20-1 für nach unten und 9-20-2 für nach oben geschickt, unabhängig auf welcher QRG wir uns gerade befinden (ohne jetzt zu tief ins Detail gehen zu wollen, wofür die einzelnen Werte stehen). Es muss nichts gerechnet werden, der Arduino erkennt die Drehrichtung eines Rotaryencoders und sendet den verhältnismäßig kurzen „String“. MIDI erlaubt damit ein flüssiges übers Band drehen.

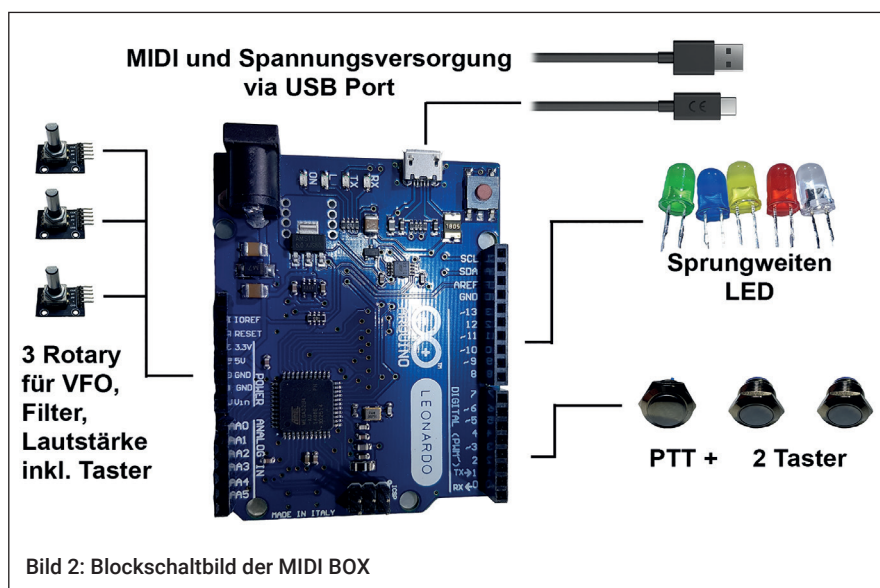
Bei der Lautstärke kann ebenfalls ein Rotaryencoder verwendet werden, da hier ebenfalls nur übermittelt wird in welche Richtung gedreht wird und den Rest macht die SDRC. Die GUI der SDRC begrenzt auch gleich den Abstimmbereich, sodass wir nicht aus dem definierten Bereich geraten, auch wenn wir am Rotaryencoder weiterdrehen. Dies alles entlastet nicht nur die Schnittstelle, sondern reduziert auch die Anforderungen an unseren Arduino erheblich.

Für die Lautstärkeeinstellung ist natürlich auch ein POTI denkbar, dann kommt dieser Abstimmknopf aber nicht für andere Funktionen infrage, weshalb ich mich universell für drei Rotaryencoder entschieden habe. Wer weiß schon, was in drei Jahren ist und ob ich die MIDI BOX nicht auch einmal für eine andere Software nutzen möchte?

Durch das minimalistische MIDI-Protokoll kommen wir mit einem kleinen kostengünstigen Arduino aus. Meine Wahl fiel zuerst auf einen Arduino UNO mit einem CH340 USB-Chip. Da der Arduino UNO von Hause aus aber kein MIDI spricht, muss der USB-Chipsatz mittels



Bild 3: LEDs zeigen die aktuelle Sprungweite an



„FLIP“ (Software von Atmel), zwischen dem normalen Kommunikationsmodus und dem MIDI-Modus umgeschaltet (geflippt) werden. Leider musste ich lernen, dass sich der CH340 nicht „flippen“ lässt! Also habe ich zum Arduino UNO mit ATmega16u2 gewechselt, denn dies ist der Chipsatz, der sich flippen lässt. Denken Sie bitte daran bei Ihrer Bestellung!

Ist das Projekt fertiggestellt, stellt das „flippen“ kein großes Hindernis dar. Während dem Entwickeln und Testen ist es allerdings mehr als nervig: Ist der ATmega auf MIDI geflippt, kann er mittels der ArduinoIDE nicht mehr programmiert werden. Hat man ihn zum erneuten Programmieren zurück auf COM geflippt, spricht er kein MIDI mehr.

Beim nun fertigen hier vorgestellten Projekt spielt dies jedoch keine Rolle mehr, da der Sketch (so nennt sich das Programm, welches mit der ArduinoIDE erzeugt wird) fertiggestellt ist und die Pins zum Anschluss der einzelnen Rotaryencoder und Taster bekannt sind. Man kann den Arduino UNO flashen, flippen, alles anschließen und ist fertig. Ein hin und her flippen ist nicht nötig.

Gibt es eine elegantere Lösung? Ja. Man nehme einen Arduino Leonardo oder Micro Pro. Der Leonardo ist Pin-kompatibel zu dem UNO, verwendet jedoch einen ATmega32u4, welcher gleichzeitig MIDI und CAT spricht. Der Arduino Leonardo ist etwas teurer als der UNO, dies macht sich aber aus meiner Sicht durch das einfachere Handling bezahlt.

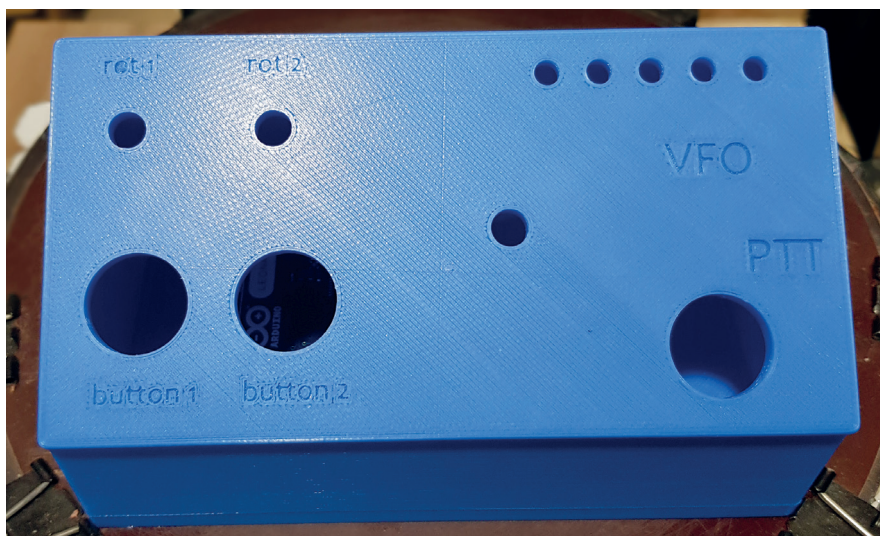


Bild 4: Passendes Gehäuse, mit einem 3D Drucker hergestellt



Tabelle	
Send	Description
FA;	Read RX1
FB;	Read RX2
FC;	Read RX3
FD;	Read RX4
FE;	Read RX5
FF;	Read RX6
MD;	Current VFO mode
IF;	Transceiver status
SM;	S-meter
PS;	Power ON status
GT;	AGC status
SH;	Filter (high)
SL;	Filter (low)
AG;	AF gain
AG0;	AF gain (Main)
AG1;	AF gain (Sub)
NA;	Current radio name
ID;	Current radio ID
SZ;	Sample rate
FT;	TX VFO
FR;	RX VFO
MG;	Unsupported
SA;	Unsupported
ZZ0;	Unsupported

Genug der allgemeinen Infos. Schließlich möchte ich das Projekt vorstellen, so dass es leicht nachgebaut werden kann. Was benötigen wir außer dem Arduino? Für mein Setup habe ich mich für drei Rotaryencoder mit eingebauten Tastern

sowie drei weitere Taster entschieden (**Bild 2**). Zwei der Taster hängen zwar mangels Pins des Leonardos parallel, bieten aber in der Praxis eine benutzerfreundliche Bedienung. Der nächst größere Arduino MEGA hätte mehr Pins, kam mir aber insgesamt etwas überdimensioniert vor.

Funktionen der drei Rotaryencoder und deren integrierter Taster sowie der drei Zusatztaster:

1. (großer) Drehknopf: Drehen: VFO-Abstimmung  
Drücken: Einstellung der Sprungweite des VFOs
  2. Drehknopf: Drehen: Lautstärkeeinstellung des aktuellen Empfängers  
Drücken: Stummschalten (Muten) bzw. Einschalten des aktuellen Empfängers
  3. Drehknopf: Drehen: Einstellung der Empfangsfilterbandbreite.  
Drücken: Stummschalten bzw. Einschalten aller Empfänger
1. Taster: PTT
  2. Taster: Ebenfalls Stummschalten bzw. Einschalten aller Empfänger
  3. Taster: Kann noch individuell belegt werden

Dies ist meine Funktionsbelegung, kann aber von jedem adaptiert werden wie es ihm beliebt. Was ist übrigens mit Sprungweite gemeint? Da der Rotaryencoder kein grob/fein Raster hat, nutze ich den Taster des Rotaryencoders dazu die Sprungweite pro Rastschritt umzuschalten. Die feinste Abstimmsschritt-

weite, die die SDRC unterstützt, sind 5 Hz. Damit würde es ewig dauern übers Band zu drehen. Drückt man einmal auf den VFO erhöht sich die Sprungweite auf 50 Hz (mein Standard), ein nochmaliges Drücken auf 500 Hz usw. Nach 5-maligem Drücken, ist man wieder bei 5 Hz. Die aktuelle Abstimmsschrittweite wird mittels einer LED-Reihe über dem VFO angezeigt (**Bild 3**). Danke für die Idee an Piero, IK1IYU.

Falls Sie meinen Bauvorschlag nur als Anregung für eigene Weiterentwicklungen sehen, gibt es noch einige Punkte zu beachten. Alle Arduinos haben sogenannte Interrupt Pins. Diese werden, egal was der Arduino gerade macht, unmittelbar ausgewertet. Dieses machen wir uns beim VFO zunutze, denn dies wird wohl bei jedem die höchste Priorität haben, oder?

Der Ein oder Andere wird sagen: PTT. Aber mal ehrlich, wenn Sie die PTT drücken wollen, drehen Sie dann noch gleichzeitig am VFO oder an der Lautstärke? Wohl eher nicht. Man drückt die PTT und danach dreht man vielleicht noch an der Lautstärke um sich ohrschmeichelnd zurück zuhören.

Die restlichen Anschlüsse für Rotaryencoder und Tasten können frei gewählt werden. Mit den verwendeten Tastern und Rotaryencodern sind wir schon am Limit des Leonardo, auch einer der Gründe weshalb ich mich gegen einen wesentlich kleineren Micro Pro mit ATmega32u4 entschieden habe. Dieser müsste mit einem I2C Bus Extender ausgestattet werden, um genügend Anschlüsse zu haben. Für Profis eventuell eine Alternative. Der Nachbaufreundlichkeit geschuldet, beschränken wir uns hier auf den Leonardo.

Die Inhaber eines 3D-Druckers wird es freuen, dass das passende Gehäuse (**Bild 4**) ebenfalls Teil des Projektes ist und die STL-Datei zur freien Verfügung steht.

Dieses Gehäuse besteht aus zwei Teilen und sämtliche Aussparungen sind bereits enthalten. Trotzdem kann es komplett ohne Supportstrukturen gedruckt werden. Enthalten sind die drei kleinen Löcher für die Rotaryencoder, als auch die LED-Leiste oben rechts sowie die drei großen Taster (mit einem Durchmesser von 19 mm), welche man neuerdings auch günstig bei Pollin.de erwer-



Bild 5: Gehäuse mit montierter LED-Leiste und großen Tastern

ben kann (**Bild 5**). Letztere haben sich bei mir in zwei Jahren der Praxis mehr als bewährt.

Auf der Grundplatte sind bereits vier Erhöhungen als Abstandshalter integriert, auf welche der Leonardo mittels M3x10-Schrauben montiert werden kann. Da es den Leonardo sowohl mit großem USB-Anschluss (wie für Drucker üblich) als auch mit Mikro-USB gibt, stelle ich für beide Modelle ein passendes Gehäuse zur Verfügung.

Wer keinen Zugang zu einem 3D-Drucker hat oder sich das Projekt nicht zutraut, kann sich wie auch bei meinen anderen Projekten, gerne an mich wenden. Egal ob nur Druckteile oder eine komplette MIDI BOX, wenn ich helfen kann, dann helfe ich.

Die grundlegenden Funktionen und was wir beachten müssen sind geklärt. Nun fehlen nur noch der Sketch und das genaue Schaltbild. Sie mögen es mir verzeihen, das werden Sie hier nicht finden. Um trotz gelegentlichem Updaten des Projektes das Nachbauen auch nach

Jahren zu gewährleisten, liegen vielmehr sämtliche Infos zentral auf


<https://github.com/dm4ds>.

Dort finden Sie sämtliche Pin-Belegungen und auch den Sketch zum Programmieren des Arduinos. Man kann sich von Github übrigens auch automatisch informieren lassen, sobald ein Update verfügbar ist.

Abschließend sei noch gesagt, dass ich an einer Erweiterung des hier vorgestellten Modells arbeite. Noch kämpfe ich aber mit der ein oder anderen Hürde. Dieser „große Bruder“ wird beides vereinen: MIDI + CAT. Er enthält außerdem eine Erweiterung um einen DDS um z.B. Quindar Töne, wie sie die NASA bei den Apollo-Missionen verwendete, erzeugen zu können. Auch das leidige Thema CW mit der SDRC wird endlich gelöst werden. Ich erzeuge einen Sinuston, welcher mittels USB-Soundkarte der SDRC übertragen wird. Auf einem 4-zeiligen Display können die eingestellte Empfangsfrequenz und weitere Parameter abgelesen

werden. Trotz dieser vielen zusätzlichen Merkmale gehe ich übrigens davon aus, dass das hier vorgestellte Modell wohl für viele Nutzer eher die Wahl ist, auch weil der „große Bruder“ ein wesentlich größeres Gehäuse haben wird.

Ein Vorgeschmack des „großen Bruders“ war ja bereits in der Nr. 3 Jg. 47 zu sehen. Leider erschien der Artikel nicht unter meinem Namen/Rufzeichen, wofür sich die AMSAT-DL bei mir mehrmals entschuldigt hat. Andere Magazine, in denen Entwicklungen von mir unter anderem Rufzeichen/Namen veröffentlicht wurden, interessierten sich wenig, um nicht zu sagen überhaupt nicht darum. Gelebter Ham-spirit.

Es bleibt mir nur noch, Ihnen ein fröhliches Nachbauen zu wünschen und ich hoffe, dass Ihnen meine MIDI BOX ähnlich viel Vergnügen bereiten wird wie mir. Ich beabsichtige zu gegebener Zeit auch meine Weiterentwicklung zu veröffentlichen. Ihr Feedback zur MIDI BOX ist meine Motivation den „großen Bruder“ fertigzustellen. 

## Das QO-100 Worldwide DATV-Netz

Da ich immer versuche, den Lebenszyklus der Projekte hier zu verlängern, habe ich mir ein paar Ideen rund um QO-100 ausgedacht:

Die erste Idee war die QO-100 DX Alert WhatsApp Gruppe, die jetzt fast 250 Mitglieder auf der ganzen Welt hat. Die Gruppe hat mir (und anderen) sehr geholfen, das DXCC/SAT zu erreichen und es gibt mittlerweile auch einen Vorschlag an den DARC ein WAE/SAT zu implementieren!

QO-100 bietet auch die tolle Möglichkeit, auf DATV zu lernen und zu experimentieren, dies war beispielsweise hier in Brasilien bisher völlig unbekannt!

Um die OM's zu animieren, Live-QSO's zu machen und etwas zu präsentieren, das mit QO-100 / Amateurfunk zu tun hat oder einfach nur ein „Hallo“ in jeder Sprache zu sagen, wurde die Idee eines weltweiten DATV-Netzes erstmals am 19. September 2020 umgesetzt.

Das QO-100 WW DATV Net ist ein wöchentliches „freies Netz“, das jeden Sonntag ab 1300 UTC stattfindet.

Die Downlink-Frequenz und Sendepa-



**QO-100 World DATV Net**  
10499.25MHz 333 2/3 QPSK Sundays 13:00 UTC

Participant List:  
<https://bit.ly/3hJypIB>

parameter sind wie folgt: 10 499,250MHz, DVB-S2, QPSK, 333ks, FEC 2/3.

Weitere Informationen finden Sie unter:

<https://bit.ly/3hJypIB>

Dort ist auch ein Tx-Log zu finden, in dem die beteiligten OM's ihre Sendung ankündigen können. In dem Log sind auch Links zu Aufnahmen der vergangenen WW DATV Net Veranstaltungen enthalten.

Es würde mich freuen, wenn auch mehr Stationen der AMSAT-DL an dem Netz teilnehmen würden. Wir haben immer noch eine sehr geringe Beteiligung von EU-Stationen. Andere Regionen sind stärker vertreten, insbesondere OM's aus Indien fanden die Idee einfach toll und sind immer dabei.

Ed Erlemann, PY2RN  
(überarbeitete deutsche Version von Matthias Bopp, DD1US)