

70MHz DDS Generator

Author DK1RI, Version V02.2, 20210908

This paper is published in <https://github.com/dk1ri> as well

Einleitung

Dies ist die Beschreibung eines 70MHz DDS Generators.

Dieses Gerät kann als Sender für einzelne Frequenzen mit Infrarotsteuerung oder als universeller Sender mit Steuerung mit serieller Schnittstelle / I2C verwendet werden.

Das Gerät kann in einem MYC System verwendet werden, aber auch unabhängig davon mit (binären) Befehlen gesteuert werden. Die Befehle sind im Kapitel „Einbindung in das MYC System“ beschrieben.

Das Interface arbeitet als Slave am I2C Bus oder kann über RS232 / (USB) gesteuert werden. Defaultmäßig sind alle Schnittstellen aktiv. Mit dem Initialisierungsbefehl können diese aber deaktiviert werden. Der Initialisierungsbefehl funktioniert aber immer.

Beschreibung

Ursprünglich benötigte ich einen Steuersender, der wenige einzelne Frequenzen in den Amateurfunkbändern liefert. Dieser sollte einen 10W Verstärker [7] ansteuern, um die Abstimmung eines automatischen Antennen - Anpassgerätes zu ermöglichen.

Der Betrieb in dieser Form ist eine gültige Amateurfunklizenz erforderlich!

Für diesen Zweck wird die Schaltung nur teilweise bestückt und die Steuerung erfolgt mit einer Infrarot Fernbedienung ähnlich der Idee nach [8]

Das Gerät kann auch in einen MYC Umfeld als universell steuerbarer Generator verwendet werden..

Die Eagle Daten für die Leiterplatte stehen unter [1].

Die Stromversorgung ist 12V +-10%, Stromaufnahme ca. 140 mA max. Hinzu kommt der Strom des 10W Verstärkers (<2A) und des Relais.

Die Steuerung kann über I2C, USB oder RS232 oder (eingeschränkt) Infrarot erfolgen.

Grundsätzliche Bedienung:

Die Stellung des Schalter an SL11 bestimmt, ob das Interface im Infrarotmode oder MYC Mode arbeitet. Ohne Schalter oder „Schalter aus“ ergibt den Infrarot Mode.

Im MYC Mode ist die Steuerung über Infrarot abgeschaltet und im Infrarot Mode funktionieren die MYC Befehle gegebenenfalls nicht. Im Infrarot Mode werden die RC5 Codes über die serielle Schnittstelle / USB ausgegeben.

Das Umschalten der beiden Modes im Betrieb ist möglich.

Beim Einschalten ist der Generator abschaltet, auch nach dem Umschalten des Mode. Relais und Endstufe sind im Infrarot Mode immer eingeschaltet.

Steuerung mit Infrarot

Der automatische Antennenanpassgerät MFJ 998RT benötigt zur Abstimmung eine Spannung, die einer Leistung von 5 – 8W entspricht. Wenn das Anpassgerät schon auf eine Antenne abgestimmt war, geht die Anpassung an eine geänderte Frequenz zwar sehr schnell. Die verwendete Endstufe SPE, 1,3kW schaltet bei Fehlanpassung aber schneller ab und auch der Transceiver IC705 regelt zu

weit zurück, bevor das Anpassgerät umschaltet.

Nach dem Einschalten im MYC Mode ist der DDS Sender und die Endstufe abgeschaltet. Das Sendesignal des IC705 wird von SL7 auf SL6 durchgeleitet. Dort ist die Endstufe mit dem Anpassgerät angeschlossen. Dies ist die Einstellung für den normalen Sendebetrieb.

Bei Fehlanpassung schaltet die Endstufe ab und auf Durchgang oder man schaltet sie manuell auf Durchgang. Das Interface wird in den Infrarot Mode geschaltet, die LED geht an. Die Endstufe schaltet ein. Der DDS Generator wird durch Auswahl einer geeigneten Frequenz mit der Infrarot Fernbedienung eingeschaltet. Das Ausgangssignal an SL4 geht auf einen 10W QRP Verstärker [7], der nicht zurückregelt. Das Ausgangssignal der Endstufe geht über SL5 – SL6 durch die Endstufe an das Anpassgerät. Die maximale Eingangleistung der Endstufe (10W) wird auch nicht überschritten, falls die Endstufe nicht abgeschaltet wurde.

Nach dem Abstimmvorgang wird der DDS Sender mit dem ersten Infrarotcode wieder abgeschaltet oder mit dem Schalter an SL11 wieder in den MYC mode gebracht.

In der bestehenden Softwareversion sind 19 Frequenzen vorgesehen. Um alle Frequenzen einzustellen, ist eine Fernbedienung mit 20 Tasten nötig.

Die Software hat vorgegebene Infrarotcodes zur Einstellung der Frequenzen. Diese können aber geändert werden.

Zunächst muss man die RC5 Adresse und die Kommandonummern zu den Tasten herausfinden.

Der DDS Generator gibt Adresse und Kommandonummern einer vorhandenen Fernbedienung auf der seriellen Schnittstelle / USB als MYC info aus, Diese Schnittstelle muss dann aber bestückt werden:

0x0F<adresse><wert> (im Hex format)

zum Beispiel: &H0F0003

Es ist möglich, die Defaultwerte der Adresse und Kommandonummern im MYC modes über die serielle Schnittstelle / USB / I2C zu ändern:

0x0D<adresse><wert> (im Hex format)

<adresse>	<Daten>	default code
&H00:	RC5Adresse	1
&H01:	Code zum Abschalten	1
&H02:	Code für 151000 Hz	2
&H03:	Code für 351000 Hz	3
&H04:	Code für 365000 Hz	4
&H05:	Code für 379000 Hz	5
&H06:	Code für 710000 Hz	6
&H07:	Code für 1006500 Hz	7
&H08:	Code für 1405000 Hz	8
&H09:	Code für 1417500 Hz	9
&H0A:	Code für 1434500 Hz	0
&H0B:	Code für 1806500 Hz	56
&H0C:	Code für 2105000 Hz	34
&H0D:	Code für 2120000 Hz	87
&H0E:	Code für 2144500 Hz	33
&H0F:	Code für 2405000 Hz	32
&H10:	Code für 2805000 Hz	12
&H11:	Code für 2870000 Hz	17
&H12:	Code für 2920000 Hz	16
&H13:	Code für 5010000 Hz	41
&H14:	Code für 5190000 Hz	46

RC5 Adresse und Kommandonummern können auch in das Programm DDS_generator.bas

eingetragen werden. Die Konstantenpaare (RC5 Code - Frequenz) heißen Commx / Fx. Das Programm muss neu kompiliert werden und der Prozessor neu programmiert werden. Werden weniger Frequenzen benötigt, werden bei den nicht benötigten Commx ungültige RC5 Codes eingetragen.

Die verwendete Endstufe benötigt 400mVpp (140mVeff) zur Aussteuerung bei 10MHz, bei 30MHz etwa 1Vpp.

Der Ausgangspegel des DDS Generators beträgt 1Vpp an 50Ohm. Mit $R6 = R7 = 100\text{Ohm}$ ergibt sich 1Vpp an 50Ohm. Dieser Pegel kann direkt verwendet werden: ($R8 = 0\text{ Ohm}$ $R9 = -$)

Im MYC System

Neben den Standardbefehlen sind 18 weitere Befehle vorgesehen: für das Einstellen der Frequenz, das Schalten des Relais / der PA, die Eichung und die Eingabe der RC5-Adressen. Weiterhin einige Befehle zum Auslesen der Daten.

Der Generator wird durch Eingabe einer Frequenz eingeschaltet und mit &H010000000 oder &H0300 abgeschaltet. &H0301 schaltet den Generator mit der letzten Frequenz danach wieder ein. Ist der Generator bereits eingeschaltet, stellt &H0301 die letzte gespeicherte Frequenz ein.

Wenn der Temperatursensor eingebaut ist und mit &H0801 aktiviert wurde, wird die Temperatur circa alle 30 Sekunden gemessen. Wenn ein Temperaturkoeffizient des Oszillators eingegeben wurde, wird dieser bei jeder Änderung der Frequenz und Temperatur berücksichtigt.

Weitere Details stehen bei „Eichung“.

Es wird empfohlen, ein abschirmendes Gehäuse zu verwenden.

Einbindung in das MYC System

Details zum MYC System stehen in [3].

Folgende Befehle werden akzeptiert:

Announce:

'Befehl &H00

'eigenes basic announcement lesen

'basic announcement is read to I2C or output

Data "0;m;DK1RI;70MHz DDS Sender;V01.1;1;180;1;26"

,

'Announce1:

'Befehl &H01

'Frequenz

'frequency

Data "1;op,frequency;1;70000000;lin;Hz"

,

'Announce2:

'Befehl &H01

'Frequenz

'frequency

Data "2;ap,as1"

,

'Announce3:

'Befehl &H03

'DDS ein

'DDS on

Data "3;os,switch on;1;0,off:1 on"

'
'Announce4:
'Befehl &H04
'DDS ein
'DDS on
'Data "4;as,as3"
'
'Announce5:
'Befehl &H05
'Eichung
'calibrate
'Data "5;op,calibrate;1;16777215;lin;-"
'
'Announce6:
'Befehl &H06
'Eichung
'calibrate
'Data "6;ap,as5"
'
'Announce7:
'Befehl &H07
'Eichung speichern
'store calibrate
'Data "7;ot,sore calibrate;0,idle;1,store"
'
'Announce8:
'Befehl &H08
'Temperatursensor vorhanden
'temperature sensor available
'Data "8;os,temperature sensor available;1,0,off;1,on"
'
'Announce9:
'Befehl &H09
'Temperatursensor vorhanden
'temperature sensor available
'Data "9;as,as8"
'
'Announce10:
'Befehl &H0A
'Temperatur
'temperature
'Data "10;ap,temperature;1;1024,{0.0 To 102.3};lin;C"
'
'Announce11:
'Befehl &H0B
'Abweichung durch Temperatur
'Tc
'Data "11;op,Tc;1;65536,{-32768 To 32767};lin;ppb/K"
'
'Announce12:

'Befehl &H0C
 'Abweichung durch Temperatur
 'Tc
 Data "12;ap,as11"
 ,
 'Announce13:
 'Befehl &H0D
 'RC5 Codes
 'RC5 Codes
 Data "13,om,RC5 Codes;1;0,20,{adress;1 to 19};b,{0 to 127}"
 ,
 'Announce14:
 'Befehl &H0E
 'RC5 Codes
 'RC5 Codes
 Data "14;am,as13",
 ,
 'Announce15:
 'Befehl &H0F
 'aktueller RC5 Code
 'actual RC5 Code
 Data "15;am,actual RC5 code;1;b,address;b,code",
 ,
 'Announce16:
 'Befehl &H10
 'Relais
 'Relais
 Data "16,os,relais;1;0,off;1,on"
 ,
 'Announce17:
 'Befehl &H11
 'Relais
 'Relais
 Data "17,as,as16"
 ,
 'Announce18:
 'Befehl &H12
 'Tk Messung
 'Tk measurement
 Data "18,os,Tk measurement;1;0,off;1,on"
 ,
 'Announce19:
 'Befehl &H13
 'Tk Messung
 'Tk measurement
 Data "19,as,as18"
 ,
 'Announce20:
 'Befehl &HF0<n><m>
 'announcement aller Befehle lesen

```

'read m announcement lines
Data "240;ln,ANNOUNCEMENTS;100;26"
'
'Announce21:
'Befehl &HFC
'Liest letzten Fehler
'read last error
Data "252;aa,LAST ERROR;20,last_error"
'
'Announce22:
'Befehl &HFD
'Geraet aktiv Antwort
'Life signal
Data "253;aa,MYC INFO;b,ACTIVE"
'
'Announce23:
'Befehl &HFE :
'eigene Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,37,
{0 to 127};a,RS232,1;a,USB,1"
'
'Announce24:
'Befehl &HFF :
'eigene Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,37,{0
to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1;a,USB,1"
'
Announce25:
Data"R $10 IF $9 = 1"

```

Fehlermeldungen

Der Befehl &HFC liefert den letzten Fehler im Format:
aktuelle Befehlsnummer - Fehler - Befehlsnummer beim Auftritt des Fehlers
Dazu werden die empfangenen Befehle von 0 bis 255 umlaufend gezählt.
Nach 254 korrekten Befehlen wird der Fehlereintrag gelöscht.

Reset

Ist der Reset Jumper JP5 beim Anlegen der Versorgungsspannung überbrückt, werden wieder die Defaultwerte eingelesen. Dies ist hilfreich, wenn die aktuelle I2C Adresse verloren gegangen ist.

Watchdog

Es gibt einen kompletten Hardware-reset, wenn die Hauptschleife länger als 2 Sekunde dauert.
Zusätzlich gibt es drei weitere Watchdogs, die in der vorliegenden Firmware für Tests nach ca 10 Sekunden ansprechen. Für „MYC Betrieb“ sollte der Wert auf 1 Sekunde gesetzt werden.
Die Befehlseingabe und Ausführung muss in dieser Zeit beendet sein. Danach werden die bereits

empfangenen Daten gelöscht. Dies soll falsche Eingaben vermeiden. Mit dem &HFC "letzten Fehler" Befehl kann man Eingabefehler sehen.

Bei einem I2C Lesebefehl müssen die Daten innerhalb dieser Zeit vom I2C Master abgeholt werden. Danach werden die Daten gelöscht. Neue Befehle können erst eingegeben werden, wenn alle Daten abgeholt wurden oder die Watchdog Zeit abgelaufen ist. Wird die RS232 / USB Schnittstelle verwendet, werden die Daten sofort ausgegeben.

Bei einem I2C BusLock (SDA pin auf 0) erfolgt auch ein I2C reset.

Software

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega168 oder größer.

Das aktuelle Bascom Programm verwendet einen Atmega328P.

Die Software wurde in BASCOM geschrieben [2]

Um das Programm zu kompilieren, muss das Verzeichnis common_1.11 [6] in das Verzeichnis mit dem Programm kopiert werden

Programmierung des Prozessors

Zur Programmierung des Prozessors ist ein 6poliger ISP Stecker JP6 vorgesehen.

Die Fuses müssen möglicherweise programmiert werden (siehe Bascom Programm) !! Prozessortyp und Frequenz müssen gegebenenfalls angepasst werden.

Der Jumper J1 sollte während der Programmierung entfernt werden.

Serielle (RS232 / USB) Schnittstelle

Schnittstellenparameter: 19k2 8N1.

Alternativ zur RS232 Schnittstelle kann die USB Platine UM2102 von ELV verwendet werden.

Die USB Platine wird plan auf der Oberseite der Interfaces verlötet: der USB Stecker zeigt zum Rand. USB Platine und Leiterplatte müssen voneinander isoliert werden.

I2C

Die Default Adresse ist 37 (&H25).

Mit dem Befehl &HFE03<n> kann die Adresse in n (1 ... 127) geändert werden.

Pullup Widerstände R3 / R4 müssen immer bestückt werden (1k – 10k).

Mit JP2 kann festgelegt werden, ob der externe I2C Bus mit 3V oder 5V betrieben wird.

Wenn auf den 3V Betrieb völlig verzichtet werden soll, kann IC3 (PCA9517), R1, R2, R5, R6, JP2 entfallen und alternativ wird JP3 und JP4 bestückt.

Ganz ohne I2C kann auch SL1, SL2, JP2, JP3, JP4, R1, R2, R5, R6 entfallen.

Der Gesamtwiderstand am I2C Bus sollte bei 1 bis 10 kOhm je nach Leitungslänge liegen

Mit IC3 muss R1 / R2 (<=10k) bestückt werden.

SL1 und SL2 sind parallel geschaltet. Ein Anschluss kann zur Weitergabe des I2C Signals an das nächste Gerät verwendet werden.

Um Buslocks zu vermeiden, wird circa alle 200ms geprüft, ob das SDA Signal auf „0“ liegt.

Ist das 50 mal hintereinander der Fall, wird die I2C Schnittstelle neu gestartet.

SMD

Die Leiterplatte ist teilweise mit SMD bestückt.

Stromversorgung

Die Stromversorgung ist 12V +/-10%, Stromaufnahme ca. 140mA + Strom der PA + Relais

Eichung

Die Eichung sollte zumindest einmal durchgeführt werden.

Wenn die Temperaturkompensation verwendet wird, muss zumindest einmal mit &H07 nach einigen Minuten Betrieb ein Korrekturwert gespeichert werden, da damit auch die Referenztemperatur gespeichert wird.

Die Frequenz des Generators kann mit MYC Befehlen geeicht werden.

Zur Eichung und Messung der Temperaturdrift sollte Tkmessung eingeschaltet sein: &H1201.

Damit ist eine eventuell bestehende Temperaturkompensation nicht aktiv.

Eine definierte möglichst hohe Frequenz wird gemessen und mit &H05xxxxxx korrigiert.

Der Defaultwert ist &H7FFFFFF. Höhere Werte verringern die Frequenz. Der Bereich der Änderung beträgt circa +/-4,7%. &H07 speichert den geänderten Wert und gegebenenfalls die zugehörige Temperatur. Die Einstellgenauigkeit beträgt < 1Hz bei 70MHz.

Bei meinem Oszillator ist der Korrekturwert &H7FFA70.

Ein Temperatursensor ist zur Temperaturkompensation vorgesehen. Dieser muss bestückt und aktiviert sein (&H0801)

Die Temperatur wird mit &H0A in 1/10 Grad ausgelesen.

Die gemessene Temperatur ist etwa 25 Grad höher als die Umgebungstemperatur nach einigen Minuten Betrieb.

Die Temperatur und die Frequenz wird bei zwei möglichst weit auseinanderliegenden Temperaturen und möglichst hoher Frequenz gemessen.

Der Temperaturabweichung ist $((f_2 - f_1) / f_1) / (T_2 - T_1) * 1000000000$

mein Oszillator: $((20,000720\text{MHz} - 20\text{MHz}) / 20\text{MHz}) / (45,0 - 25,0) * 1000000000 = 1800$

anders Beispiel: $((59,9999\text{MHz} - 60\text{MHz}) / 60\text{MHz}) / (45,0 - 25,0) * 1000000000 = -83,3$

Eingegeben wird $32767 + 83 = 32850$: &H0B8707

beziehungsweise $32767 - 83 = 32684$: &H0B7FAC

32767 entspricht einer Abweichung von 0 (default).

Der Temperatursensor hat eine gewisse Trägheit. Es dauert ca 1 Minute, bis eine Temperatur richtig gemessen wird und die Frequenz richtig kompensiert wird. Dies ist besonders beim Einschalten zu beachten.

Bestückung der Leiterplatte

Verwendung von ISP:

JP6

Den Quarzoszillator gibt es für verschiedene Versorgungsspannungen. Wird ein 5V Typ verwendet, kann IC3 entfallen. Pin1 und Pin2 wird dann überbrückt. I2C mit 3V ist dann nicht möglich.

Wird für IC3 (MCP1700-3302) ein 78L03 kompatibler Regler verwendet, ist die Reihenfolge der pins unterschiedlich !!!

Mit MYC mode:

Mit I2C:

Siehe I2C oben.

Mit serieller Schnittstelle:

Bei Verwendung der RS232 Schnittstelle wird IC4, C6 – C9 und SL3 bestückt. **Alternativ** dazu kann der USB Modul UM2102 verwendet werden.

Der USB Modul wird isoliert auf der Bestückungsseite angebracht,. Die mittleren 4 Anschlüsse werden mit JP8 verlötet.

Wenn das Relais und die Endstufensteuerung und die LED nicht benötigt wird, wird kann K2 T1 – T3, D5, R14, R17, R50, SL5, SL6, SL7, SL9, SL10 entfallen.

Der Transistor T3 kann bis zu 4A schalten (IRLML2244). Das reicht für eine 10W Endstufe.

Die Diode D1 muss auch eine entsprechende Strombelastbarkeit haben und kann auch überbrückt werden. Dabei entfällt aber der Verpolungsschutz.

Das DDS Sinusausgangssignal liegt an SL4, das Rechtecksignal an SL6.

Nur Infrarot mode:

Folgende Bauteile brauchen nicht bestückt werden:

SL1, SL2, SL6, JP2, JP3, JP4, IC5, R1 – R4

IC4, C6 – C9, SL3: siehe oben

Da die nicht gemessen wird, kann der Temperatursensor IC8 entfallen.

Anschlüsse

12V Power / (SL12)

Tip	12V
Ring	GND
1,2	GND
3	12V

I2C (SL1, SL2)

1	GND
2	SCL
3	SDA

RS232 (SL3)

1	GND
2	Eingang
3	Ausgang

Sinusout (SL4)

1	GND
2	Signal

PA_Out (SL5)

1	GND
2	Signal

Out (SL6)

1	GND
2	Signal

from_Tx (SL7)

- 1 GND
- 2 Signal

COM_Out (SL8)

- 1 GND
- 2 Signal
- 3 Signal invertiert

12VPA (SL9)

- 1 GND
- 2 -
- 3 12V geschaltet

LED (SL10)

- 1 GND
- 2 -
- 3 LED

IR_Myc (SL11)

- 1 GND
- 2 -
- 3 Signal

Infrarot (SL13)

- 1 GND
- 2 Daten
- 3 5V

Jumper

- JP1 Power
- JP2 I2C: 3V/5V Umschaltung
- JP3 SDA Überbrückung (ohne IC3)
- JP4 SCL Überbrückung (ohne IC3)
- JP5 Reset
- JP6 ISP
- JP7 Testeingang für Filter
- JP8 Anschluss für USB Modul

Versionen

Diese Beschreibung gilt für die
Leiterplattenversion V03.0
Bascom Version V01.1

Copyright

Die Ideen in diesem Dokument unterliegen der GPL (Gnu Public Licence V2) soweit keine früheren, anderen Rechte betroffen sind.

Die Verwendung der Unterlagen erfolgt auf eigene Gefahr; es wird keinerlei Garantie / Gewährleistung / Produkthaftung übernommen.

The ideas of this document can be used under GPL (Gnu Public License V2) as long as no earlier other rights are affected.

The usage of this document is on own risk, there is no warranty.

<https://kn-electronic.de/Bausaetze/FET-PA10.html>

Referenzen

- [1] https://www.dk1ri.de/dhw/70MHz_DDS_generator_eagle.zip
- [2] https://www.dk1ri.de/dhw/70MHz_DDS_generator_bascom.zip
- [3] <https://www.dk1ri.de/myc/MYC.pdf>
- [4] <https://dk1ri.de/myc/Description.txt> oder <https://dk1ri.de/myc/Description.pdf> (englisch)
- [5] <https://dk1ri.de/myc/Definitions.txt> oder <https://dk1ri.de/myc/Definitions.pdf> (englisch)
- [6] https://dk1ri.de/dhw/common_1.11.zip
- [7] <https://kn-electronic.de/Bausaetze/FET-PA10.html>
- [8] <https://www.elektronik-labor.de/AVR/dds/AD9850.html>
- [9] https://dk1ri.de/dhw/infrarot_tester.zip