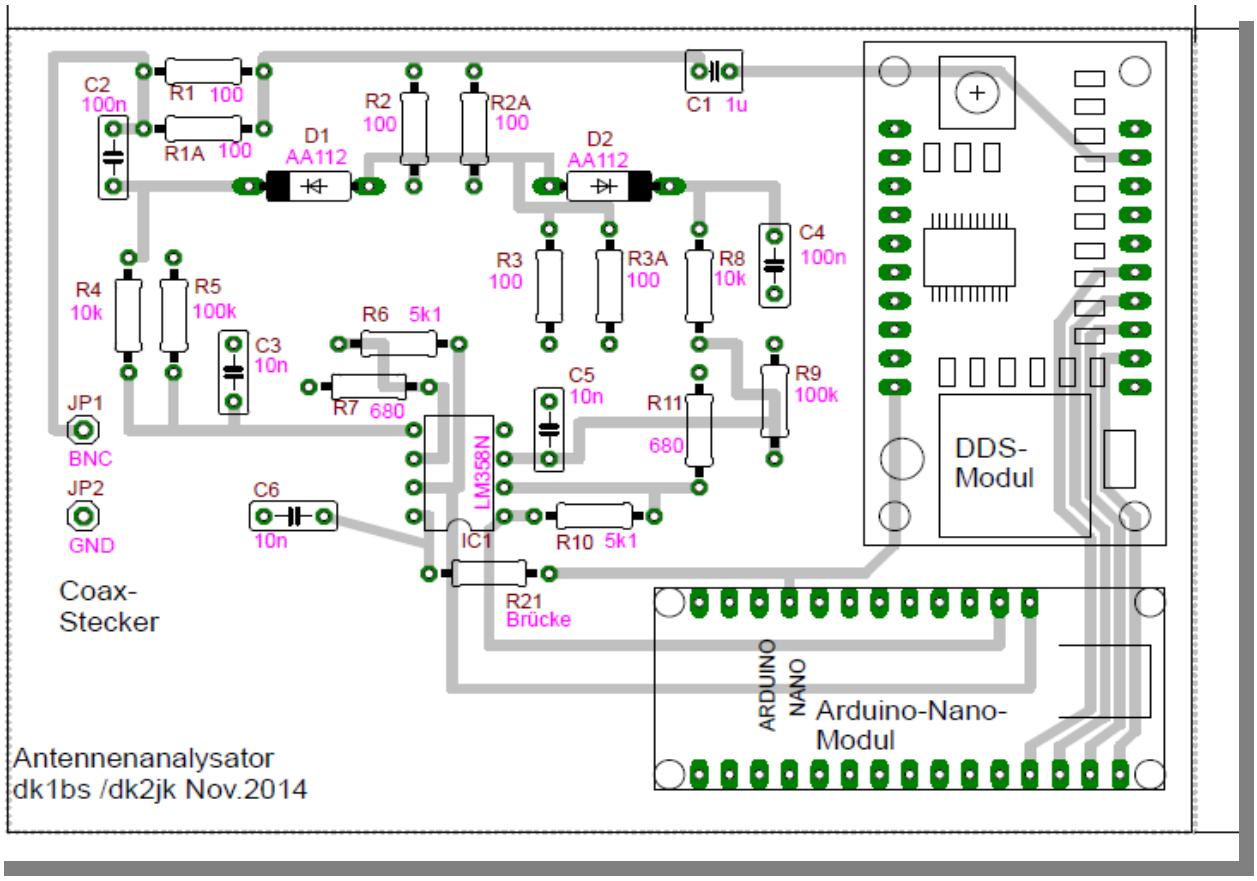
Bauanleitung zum Antennen - Analyser

Bestückung



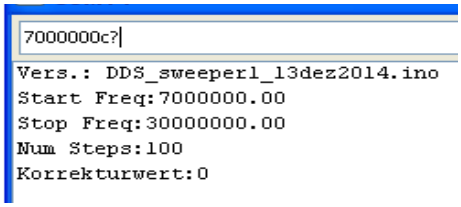
Wir beginnen mit dem Einlöten der niedrigsten Bauelemente: Widerstände und Dioden. Am einfachsten scheint es zu sein, die Widerstände aus dem Bausatz zu messen und an die vorgesehenen Plätze zu löten. R21 ist eine Drahtbrücke. Bei den Dioden auf die Polarität achten (Balken !). Als nächstes kommt die IC- Fassung IC1 dran. Dann werden die Kondensatoren bestückt– hier gibt es nur 3 verschiedene : 0.01 (10nF), 0.1 (100nF), 1.0 (1uF). Die beiden Module sitzen auf Federleisten; diese müssen evtl. auf die richtige Stiftanzahl gekürzt werden. Die Ausrichtung der Federleisten erfolgt entweder mit dem aufgesteckten Originalmodul oder einer ähnlichen IC- Fassung. An den Anschluss JP1 wird ein etwa 20mm langes Drahtstück angelötet.

Das DDS- Modul und das Arduino - Modul werden an die vorgesehenen Plätze gesteckt; man muss vielleicht die Stiftleisten etwas ausrichten, da die Stifte nicht immer ganz gerade sind.
Jetzt folgen zunächst einige Überprüfungen, bevor die Platine in das Gehäuse eingelötet wird.

Nun benötigen wir die Windows-Software für den Arduino (siehe unten „Software“). Wenn man in einer Gruppe arbeitet, so ist es günstig, wenn nur eine Person das Laden der Software und die folgenden Tests macht. Wichtig ist erst einmal, dass die serielle Schnittstelle (COMx) vom Rechner erkannt wird (siehe: „Windows Treiber Installieren“ und evtl. „Arduino Nano mit gefälschten USB-Serial-Chips“).

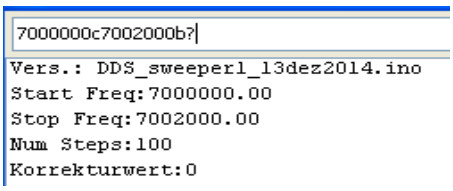
Erste Tests

Die Module „DDS“ und „Arduino Nano“ stecken an den passenden Stellen auf der Platine. Nun verbinden wird den USB-Port mit dem Rechner. Jetzt sollte am Nano-Board die grüne und am DDS-Board die rote Power-LED leuchten.
Wir nehmen eine KW-Empfänger und stellen ihn auf eine Frequenz ein (Beispiel: 7,001 Mhz). Als Betriebsart wählen wir CW oder SSB.
(Verweis: 'Liste der Kommandos').
Jetzt tippen wir am Terminal („Arduino – Serial - Monitor“) ein:
7001000c?

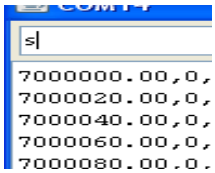


Die Eingabe erfolgt in „Hertz“ . Das Kommando 'c' stellt diese Frequenz ein und das Kommnado '?' zeigt die aktuellen Einstellungen auf dem Bildschirm an.
Im KW-Empfänger sollte jetzt ein Signalton hörbar sein; Vielleicht kann man einen kurzen Draht als Antenne in die Koaxbuchse als „Sendeantenne“ stecken.

Wir können auch die Schrittweise des DDS ausprobieren:
(Die Eingaben werden immer mit der Enter - Taste abgeschlossen).
Wir tippen ein;
7000000c7002000b?

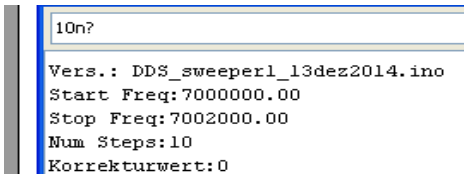


Jetzt 's' eingeben:



Das Terminal zeigt die eingestellten Frequenzen an; im KW-Empfänger müsste jetzt ein ansteigender oder abfallender Ton hörbar sein – je nach eingestelltem Seitenband.
Noch ein Experiment ! Wir ändern die Schrittanzahl (Num Steps) auf 10, indem wir eingeben:

```
10n
```



Wenn wir jetzt wieder 's' eingeben, sollte man im KW-Empfänger deutlich die Frequenzschritte hören können.
Wir haben bis jetzt nachwiesen, dass der DDS Frequenzen erzeugt.

Grob-Test der Messbrücke

Als Messobjekt schließen wir einen 50-Ohm-Widerstand an. Die Messbrücke ist jetzt symmetrisch. Das SWR sollte idealerweise 1.0 sein. Eine Kontrolle erfolgt mit dem Kommando 's' am Terminal („Arduino – Serial - Monitor“) :

```
S
1000000.00,0,1007.81,257.00,1.00
6800000.00,0,1007.78,258.00,1.00
12600000.00,0,1023.53,258.00,3.00
18400002.00,0,1039.37,259.00,5.00
24200000.00,0,1063.24,261.00,8.00
30000000.00,0,1079.37,262.00,10.00
end
```

Die 3. Spalte gibt den SWR - Wert mal 1000 wieder; es sollten also bei einem 50-Ohm-Abschlusswiderstand Werte etwas größer als 1000 entsprechend einem SWR von 1.0 angezeigt werden.

Einbau in das Weißblechgehäuse

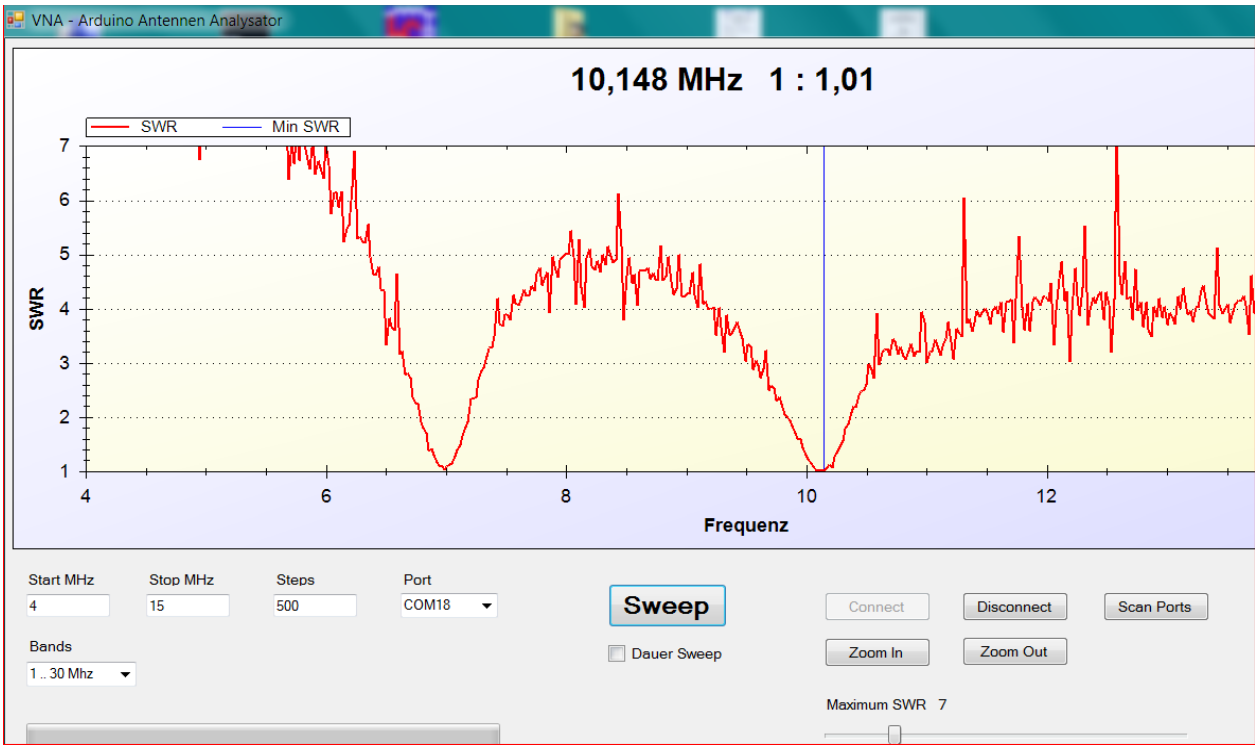
Erst wenn der Grobtest erfolgreich war, wird die Platine in das Gehäuse eingelötet. Beim Einbau in das Weißblechgehäuse orientiert man sich an dem Ausschnitt für die Mini-USB-Buchse. Die USB-Buchse muss mittig im Ausschnitt sichtbar sein. Es ist ausreichend, wenn die Platine nur an wenigen Punkten an das Gehäuse gelötet wird. Für die Fixierung der Seitenwände legt man die beiden L-förmigen Seitenwände auf die Grundplatte und lötet die Seitenwände innen zusammen. Evtl. ist ein etwas stärkerer Lötkolben nützlich.

Das VNA - Programm

Die Software ist unter DotNet programmiert und benötigt eine aktuelle Version des DotNet Frameworks (Version 4.0 oder höher). Dieses kann kostenlos von Microsoft herunter geladen werden. Ein Software Installation ist nicht erforderlich. Es reicht das Verzeichnis mit den Programmdateien an eine beliebige Stelle zu kopieren und die Datei **VNA.exe** zu starten.
Nach dem Programmstart wählt man unter Port die Com - Schnittstelle des USB – Adapters aus. Mit einem Klick auf **Connect** wird eine RS232 Verbindung zum Arduino Modul hergestellt. Unter **Start MHz** und **Stop MHz** ist der Frequenzbereich 1 .. 30 MHz voreingestellt. Diesen kann man beliebig ändern oder mit der Auswahlbox **Bands** ein Amateurfunkband wählen. Durch einen Klick auf **Sweep** werden die Daten an das Arduino Modul gesendet, dieses antwortet mit den entsprechenden SWR - Werten. Aus diesen wird dann eine Verlaufskurve gezeichnet. Unter **Steps** ist als Wert 101 eingestellt. Dies bedeutet, dass bei einem Frequenz - Sweep 101 Frequenzen gemessen werden. Man kann diesen Wert bis auf 1000 erhöhen, dann bekommt man eine höhere Auflösung mit bis zu 1000 Messpunkten – die Messung dauert dann aber etwas länger. Die Auflösung erhöht sich auch, wenn man den Frequenzbereich verkleinert, z.B. statt 1 .. 30 MHz >> 13 .. 15 MHz. Dies erreicht man auch durch Drücken der Tasten **Zoom In** und **Zoom Out** und dann **Sweep**. Die Grafik ist auf einen SWR Bereich von 1 .. 10 voreingestellt. Mit dem Schieberegler **Maximum SWR** kann man den Wert zwischen SWR 2 .. 30 zoomen. Zusätzlich kann man die Grafik auch zoomen, indem man mit der linken Maustaste einen Rahmen um einen Teilbereich zieht. Wenn man mit der rechten Maustaste auf die Grafik klickt, kann man die aktuelle Grafik in eine Datei speichern. Dies ist für eine spätere Auswertung sehr praktisch. Darüber hinaus bietet das Auswahlmenü auch noch einige zusätzliche Optionen.

(Quelle: Programm und Text von Norbert Redeker DG7EAO)

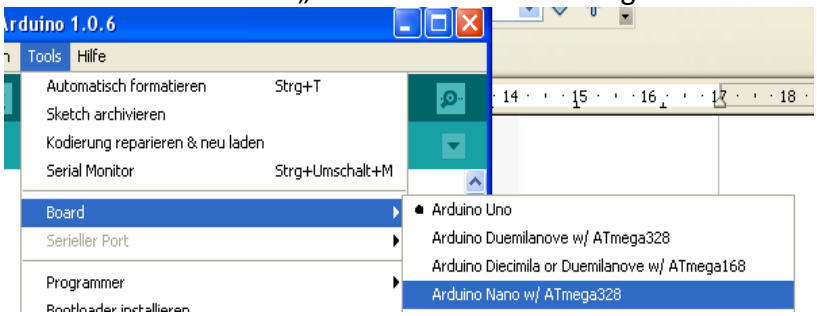
Beispielansicht:



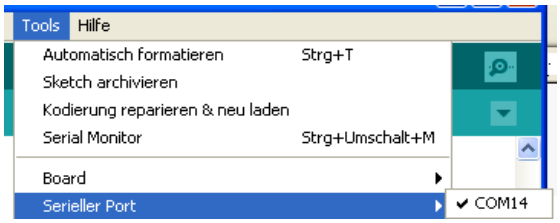
Software Laden: Arduino - Programm

Wir wählen das Antennenanalyser - Programm

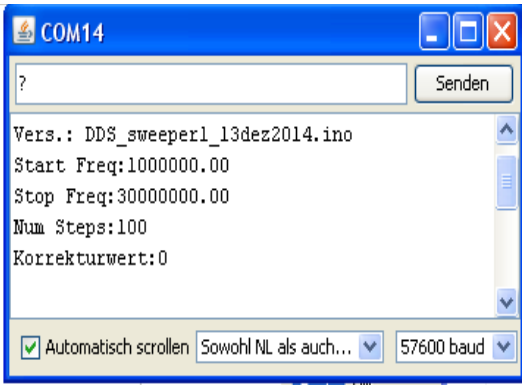
Es öffnet sich die Arduino - Oberfläche.
Über 'Tools' wählen wir zuerst das Board „Arduino Nano mit Atmega 328“



Dann wählen wir die Schnittstelle:



Jetzt kann über „Datei → Upload“ oder über den Rechtspfeil
das Programm zum Nano-Board geladen werden. Kontrolle beim Laden: Es blinken 2 rote Leuchtdioden.
Kontrolle nach dem Laden: Serial Monitor aufrufen, 57600Bd;
'?'<ENTER> eingeben:



Kommandos des Arduino - Programms

Das Arduino - Programm enthält einen einfachen Kommandointerpreter. Ein Kommando besteht aus der Eingabe einer Zahl gefolgt von einem Buchstaben. Bei Senden des Buchstabens wird das Kommando sofort ausgeführt.

Kommando	**Bedeutung**	**Beispiel**
A oder a	Startfrequenz	1000000A = 1 MHz
B oder b	Endfrequenz	30000000B = 30 MHz
C oder c	Feste Frequenz ; der DSS gibt diese Frequenz sofort aus; Der Wert wird auch als Startfrequenz gespeichert.	7032000C = 7,032 MHz
N oder n	Anzahl der Schritte N für 'Sweep'	100N = 100 Schritte
S oder s	'Sweep' ; der Signalgenerator fährt die Frequenz von Start- bis Endfrequenz in N Schritten durch ; Messwerte werden über serielle Schnittstelle ausgegeben.	S 1000000.00,0,1007.81,257.00,1.00 6800000.00,0,1007.78,258.00,1.00 12600000.00,0,1023.53,258.00,3.00 18400002.00,0,1039.37,259.00,5.00 24200000.00,0,1063.24,261.00,8.00 30000000.00,0,1079.37,262.00,10.00
K oder k	Korrekturwert	5K =Beim Rücklaufwert wird 5 abgezogen.
? oder h	Ausgabe der aktuellen Werte	? Vers.: DDS_sweeper1_13dez2014.ino Start Freq:1000000.00 Stop Freq:30000000.00 Num Steps:100 Korrekturwert:0

Abgleich

Es ist eine einfache Möglichkeit zum Abgleich vorgesehen, basierend auf der Tatsache, dass bei einem idealen Abschluss mit 50 Ohm der Rücklaufwert Null sein muss. Sollte der gemessene Wert positiv sein, so kann mit dem implementierten Abgleich ein Korrekturwert gespeichert werden, der als Offset bei der Berechnung des Rücklaufwertes abgezogen wird. Der Korrekturwert wird im EEPROM nichtflüchtig gespeichert.
Anmerkung: Ein Abgleich bei negativem Offset ist z.Z. nicht vorgesehen.

Beispiel:

```
S
1000000.00,0,1135.34,284.00,18.00
1133333.37,0,1135.34,284.00,18.00
1266666.62,0,1127.82,283.00,17.00
1400000.12,0,1136.36,282.00,18.00
1533333.25,0,1135.85,283.00,18.00
1666666.75,0,1135.85,283.00,18.00
```

Die Werte für Rücklauf befinden sich in der letzten Spalte (hier 17...18)
Um den Rücklaufwert nahe Null zu bekommen, geben wir einen Korrekturwert ein.
Im Beispiel :

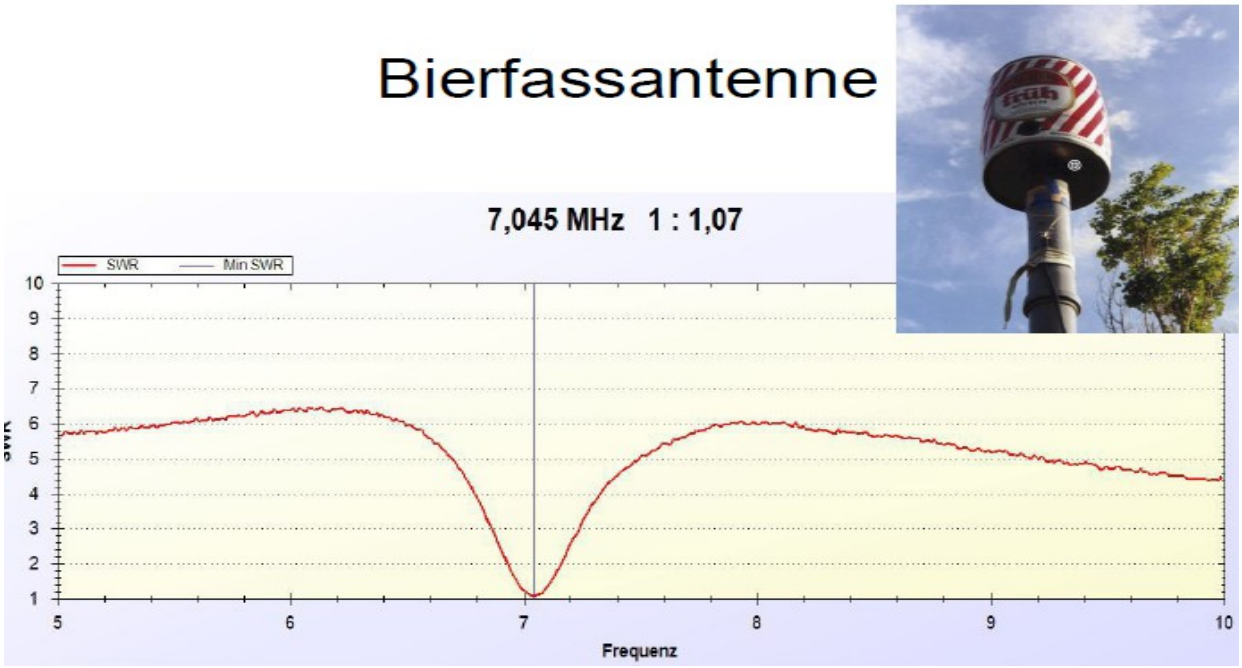
```
17k <enter>
```

Kontrolle mit

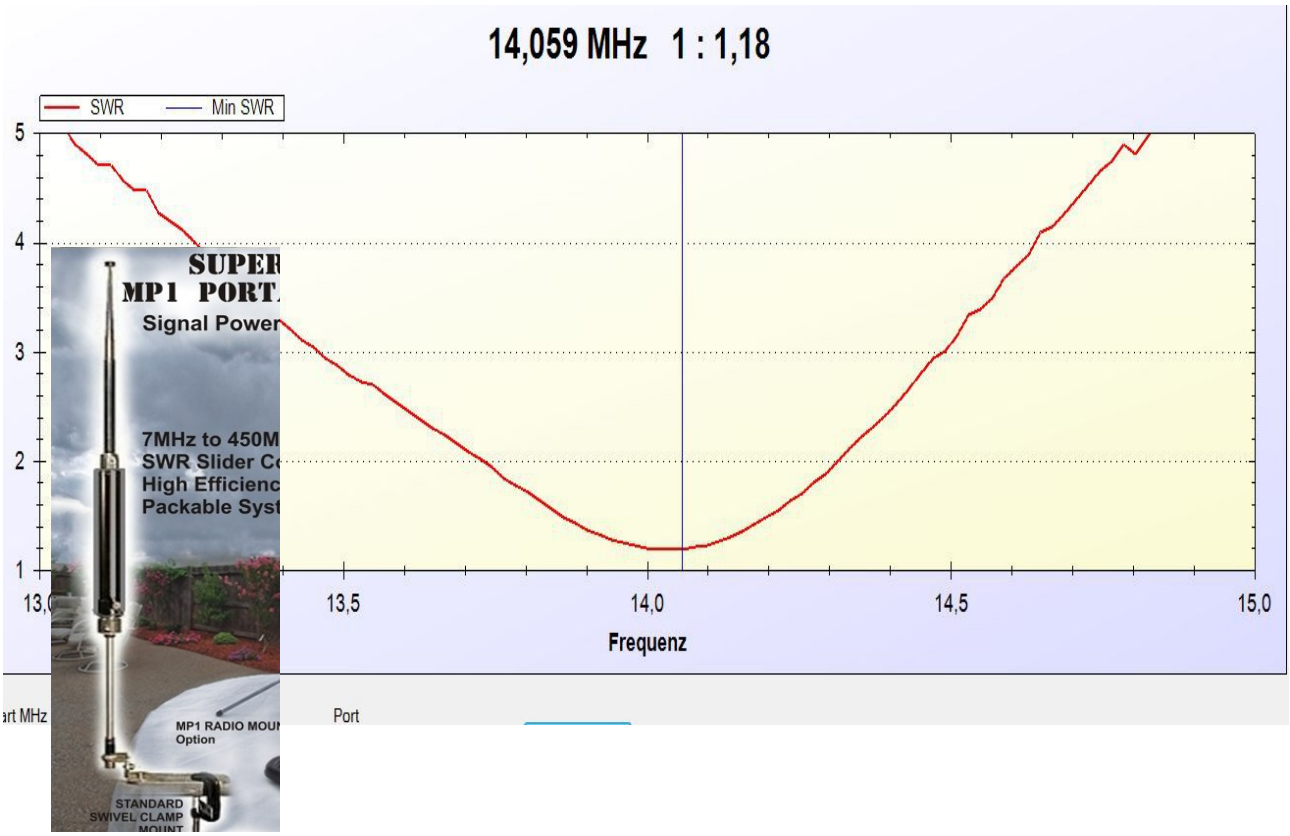
```
s <enter>:
```

```
S
1000000.00,0,1285.71,8.00,1.00
1290000.00,0,1007.07,284.00,1.00
...
2449999.75,0,1007.14,281.00,1.00
2740000.00,0,1007.14,281.00,1.00
3030000.00,0,1007.14,281.00,1.00
end
```

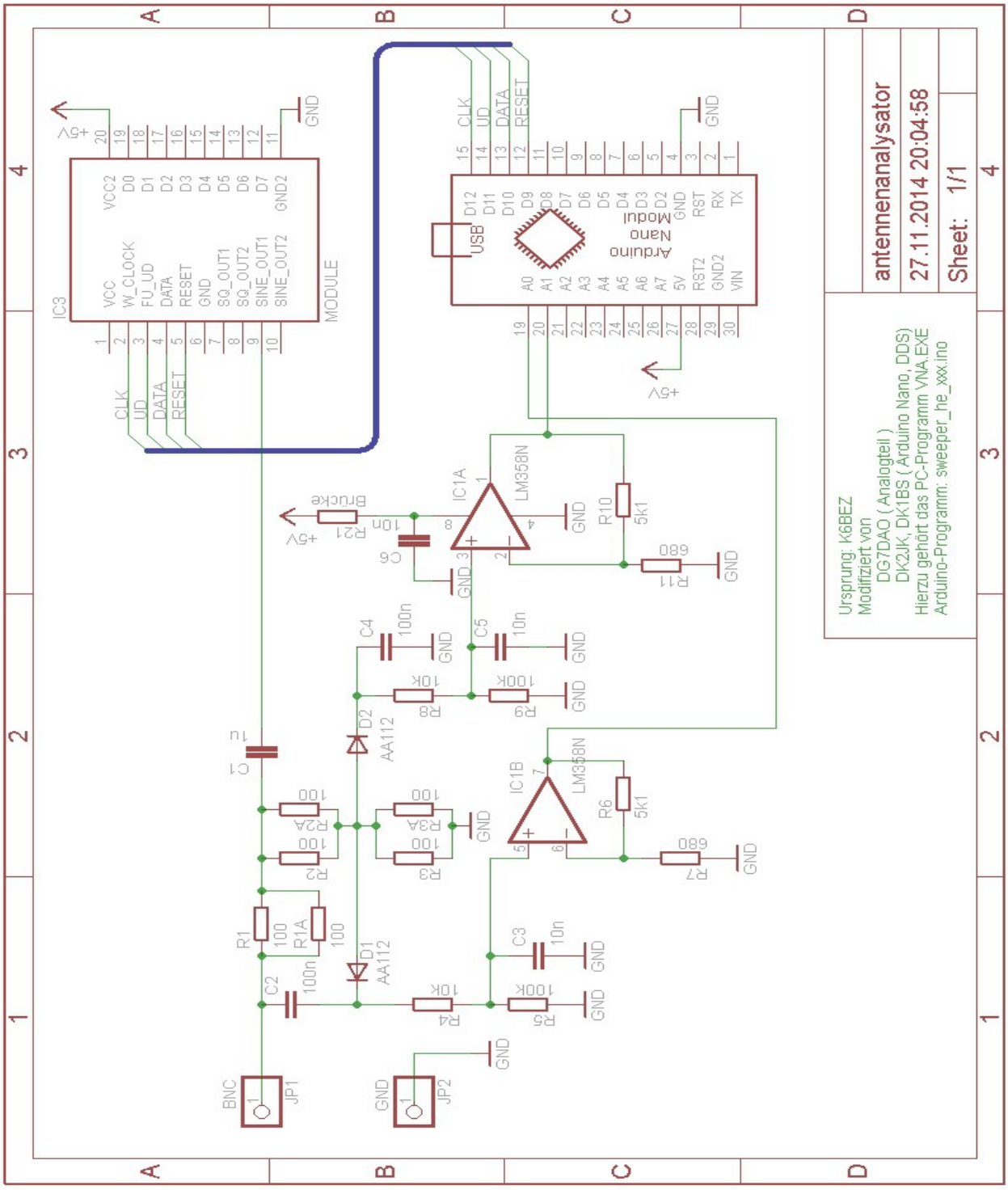
Beispielmessung „Bierfassantenne“



Beispielmessung „MP-1 Portabel Vertikal Antenne“

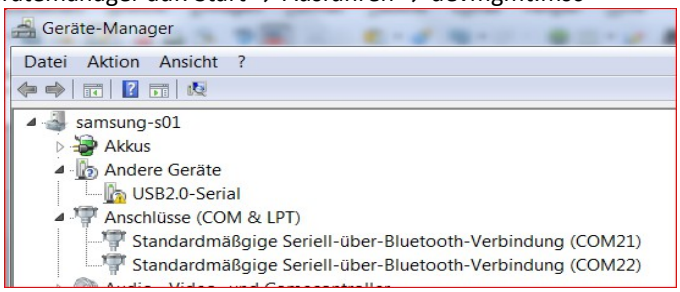


Schaltplan


antennenanalysator
27.11.2014 20:04:58
Sheet: 1/1
Ursprung: K6BEZ
Modifiziert von
DG7DAO (Analogteil)
DK2JK, DK1BS (Arduino Nano, DDS)
Hierzu gehört das PC-Programm VNA.EXE
Arduino-Programm: sweeper_he_xxx.ino
IC3
MODULE
VCC
W_CLOCK
FU_UD
DATA
RESET
GND
SQ_OUT1
SQ_OUT2
SINE_OUT1
SINE_OUT2
GND2
CLK
UD
DATA
RESET
+5V
Arduino Nano Modul
USB
IC1A
IC1B
LM358N
Bruecke
R21
C6
R10
5k1
R11
680
C4
100n
C5
10n
R8
10K
R9
100K
D2
AA112
C1
R2A
R3A
R2
R3
100
R6
R7
R1
R1A
D1
C2
C3
R4
R5
BNC
JP1
JP2

Anhang A: Windows Treiber „USB 2.0 Serial “ Installieren

Die Arduino - Software wird von der Arduino - Oberfläche über die serielle Schnittstelle geladen. Dazu wird zunächst das Nano-Board allein über ein Mini-USB-Kabel mit dem Rechner verbunden. Jetzt müsste Windows eigentlich „Neue Hardware gefunden“ sagen, tut er aber nicht...
Wir rufen den Gerätemanager auf: Start → Ausführen → devmgmt.msc



Der Rechner findet ein Gerät „USB-2.0-Serial“. Eine automatische Installation des Treibers scheint nicht möglich zu sein. Deshalb laden wir von der Herstellerseite den Treiber „CH341SER.ZIP“ :

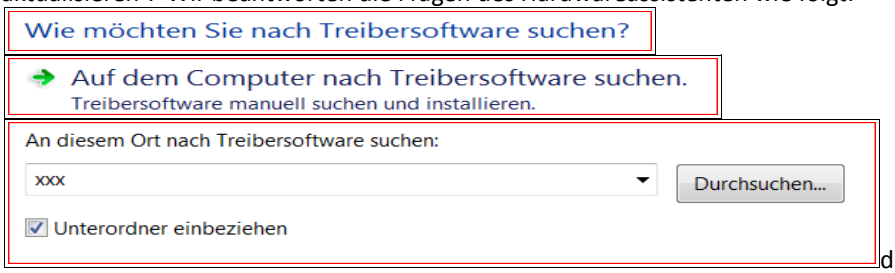
http://wch.cn/downloads.php?name=pro&proid=65

Eine Kopie befindet sich hier :

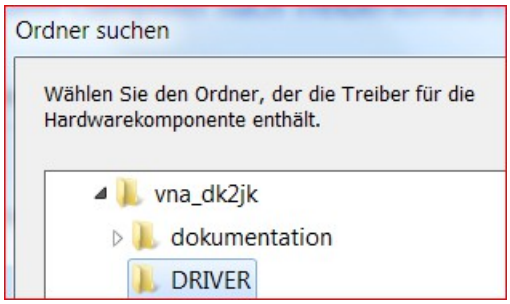
http://www.dk2jk.darc.de/vna_dk2jk/nuetzliche%20windows%20programme.html

„CH341SER.ZIP“ muss als erstes entpackt werden (Ordner merken !)

Jetzt wieder im Gerätemanager: Wir wählen „USB-2.0-Serial“ an; dann erscheint der Hardwareassistent mit „Treibersoftware aktualisieren“. Wir beantworten die Fragen des Hardwareassistenten wie folgt:



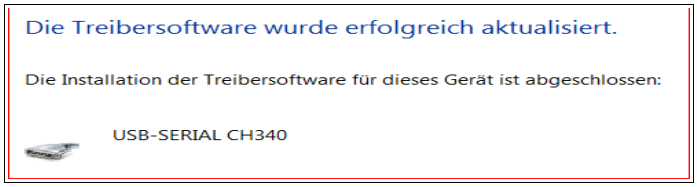
durchsuchen...



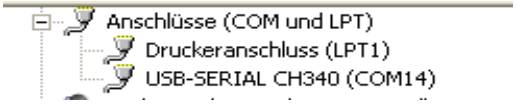
(der Ordner, wo der Treiber entpackt wurde)

OK → Weiter -> Installieren....

Gerätemanager zeigt:



In meinem Beispiel ist jetzt „USB_SERIAL CH340“ als „COM14“ installiert.



Anhang B: Arduino Nano mit gefälschten USB- Serial- Chips

Von den Arduino - Nano- Modulen gibt es verschiedene Versionen; die Chips für den USB – Serial -Wandler sind unterschiedlich. Bei unserem Projekt wurden die Nanoboards mit dem Baustein „CH340“ verwendet. Hierfür gilt die Beschreibung in Anhang A. Die „offiziellen“ Nano-Boards haben ein Chip von der Firma FTDI. Damit gibt es keine Probleme; Windows erkennt diesen Chip und installiert automatisch den richtigen Treiber. Da viele Arduino - Boards heute aus China kommen – auch die von einigen deutschen Ebayhändlern – kann es vorkommen, dass dort gefälschte FTDI-Chips drinstecken. Diese sind kaum zu erkennen, da die Beschriftung gleich ist. Wenn es eine Fälschung ist, so wird diese bei älteren Windowsrechnern erkannt, es sei denn, man hat ein Windowsupdate gemacht. Die neuesten registrierten Treiber von Windows sind so gestaltet, dass die Fälschung erkannt wird und der Baustein so umprogrammiert wird, dass er nicht mehr zu gebrauchen ist. Doch Halt! Die Programmierung erfolgt im EEPROM; sie kann also rückgängig gemacht werden. Die Internet- Foren sind voll mit diesem Thema
(Stichwort: 'unbrick ftdi arduino').

Folgende Vorgehensweise wurde für die gefälschten Chips getestet:

1. 'guten' Arduino' anschließen; Warten bis Treiber gefunden wurde.
2. Windows- Update abschalten ! (Siehe letzter Punkt 22)
3. FTDI- Treiber deinstallieren / löschen.
4. Alten FTDI Treiber besorgen z.B.: Treiber 2.10.0.0
5. Arduino mit gefälschtem Chip anschließen (Treiber wird nicht gefunden)

Wichtig ist, dass bei der Installation nichts automatisch gemacht wird. Also folgende Reihenfolge ausgehend vom Gerätemanager :

6. Treiber aktualisieren (Hardware USB)
7. Auf dem Computer suchen
8. Aus einer Liste auswählen
9. Alle Geräte anzeigen
10. durchsuchen
11. Treiber FTDI Version 10.0.0 anwählen
12. *ftdibus.inf* wahlen
13. Warnung ignorieren
14. Treiber aktualisieren (Gerätetreiber)
15. Auf dem Computer suchen
16. Aus einer Liste auswählen
17. Alle Geräte anzeigen
18. durchsuchen
19. Treiber FTDI Version 10.0.0 anwählen
20. *ftdiport.inf* wahlen
21. Warnung ignorieren
22. Wenn es dann einmal funktioniert:
    Windows Treiber Update Blockieren ! → Geräte und Drucker öffnen → Rechtsklick auf die Festplatte (PC-Name) ->Geräteinstallationseinstellungen öffnen und Einstellungen vornehmen : nie Treibersoftware von Windows update installieren → übernehmen !

Quellen

K6BEZ:

http://www.hamstack.com/hs_projects/antenna_analyzer_docs.pdf

DG7EAO :

http://lima05web.wordpress.com/2014/03/22/arduino-antennen-analysator-dg7eao/

DK2JK:

http://www.dk2jk.darc.de/vna_dk2jk/

DDS:

http://www.analog.com/static/imported-files/data_sheets/AD9850.pdf

Arduino:

http://www.arduino.cc/