

Programmierung und Benutzung des AR488 GPIB Adapters

Thomas Baier, DG8SAQ

Version 1 vom 4.9.2021

Inhalt

Einleitung.....	2
Programmierung des Adapters	5
Nutzung der KE5FX Toolbox	7
Auslesen der Bildschirmansicht von einem HP8593E Spektrumanalyzer:.....	10
Read error: 0x2.....	12
Konfiguration des AR488 Adapters	13
Drahtloser Betrieb	14
Programmieren des ESP8266 Wifi Moduls durch den Arduino hindurch.....	15
Konfigurieren des AR488 Wifi Adapters.....	21
Konfigurieren der KE5FX Plotter Software für drahtlosen Zugang	27

Einleitung

Die AR488 Firmwaren (Arduino und ESP8266) werden John Chajecki weiterentwickelt. Beide basieren auf der Arduino Plattform. Die neuesten Versionen können hier gefunden werden:

<https://github.com/Twilight-Logic/AR488>

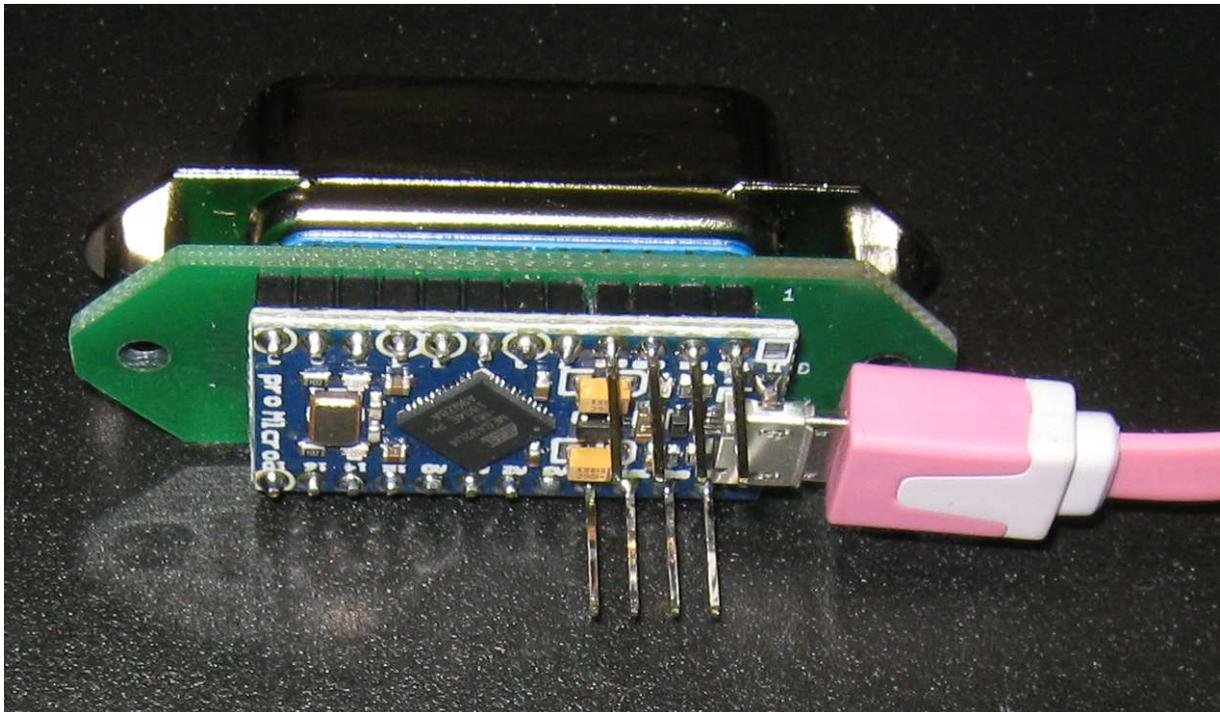
Um diese Firmwaren zu flashen, ist die Arduino Entwicklungsumgebung zu installieren:

<https://www.arduino.cc/en/Main/Software>

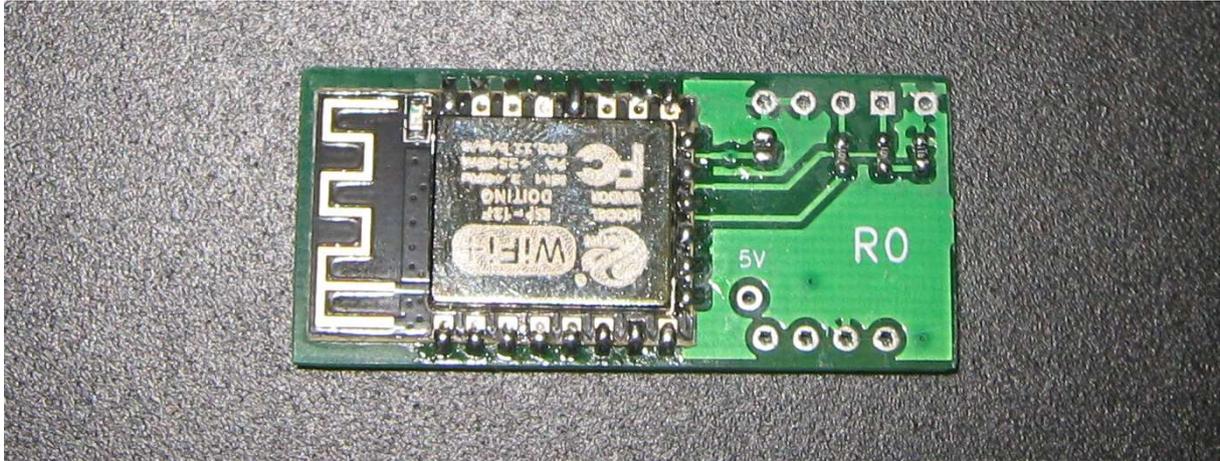
Der AR488-Adapter wurde als vollständig kompatibler Ersatz für den Prologix GPIB-USB Adapter entwickelt. Daher sollte er auch vollständig kompatibel zum KE5FX GPIB Toolkit sein, welches von John Miles KE5FX ausdrücklich für den Prologix Adapter geschrieben wurde. Das Toolkit kann hier heruntergeladen werden:

<http://www.ke5fx.com/gpib/readme.htm>

Ich habe eine kompakte AR488 Adaptervariante entwickelt, welche auf dem Arduino Leonardo ProMicro basiert ...

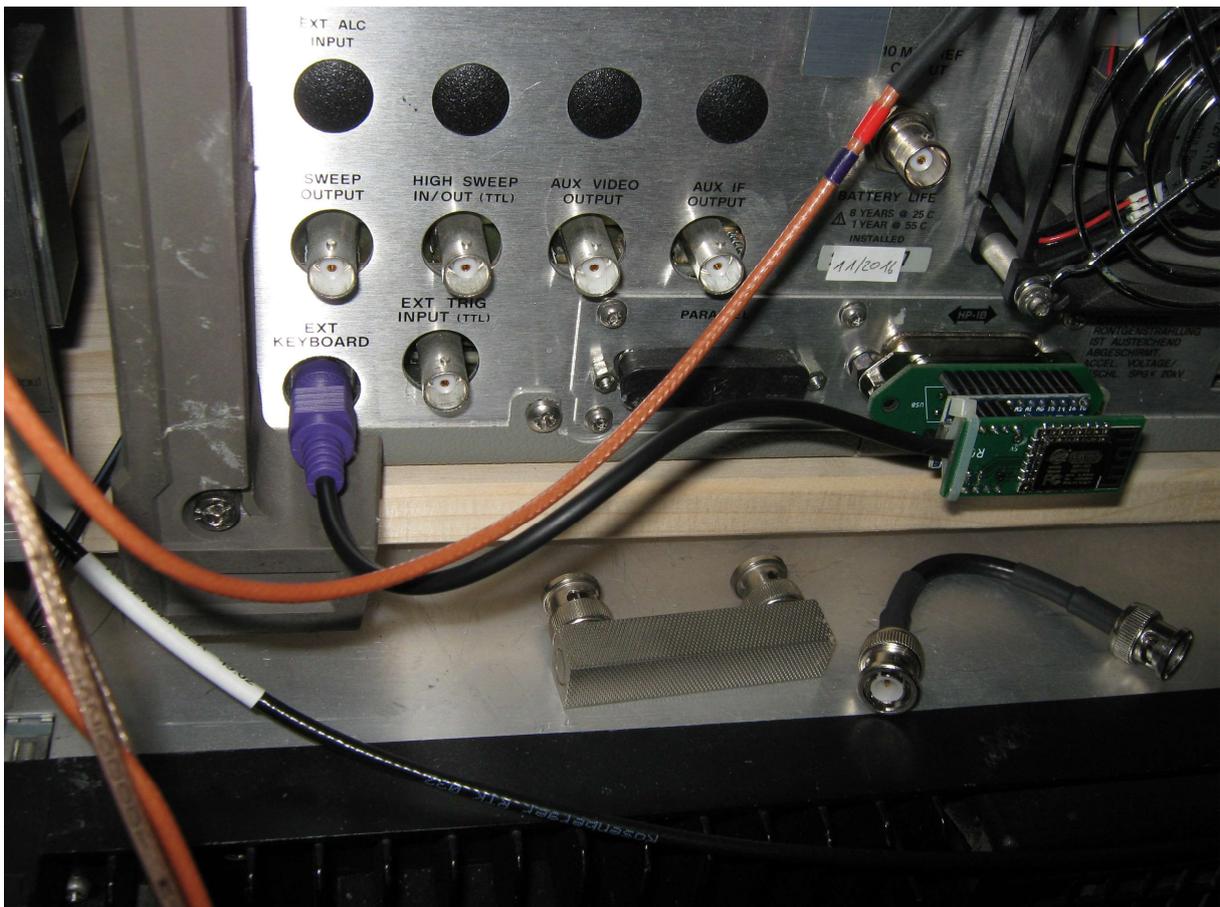


... und ein Wifi-Aufsteckmodul:



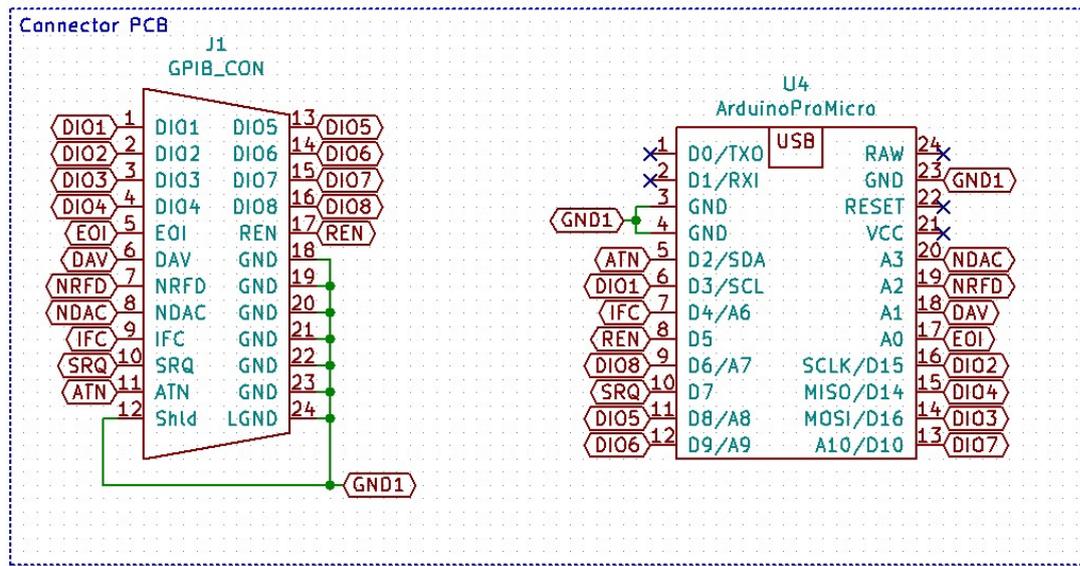
Bitte beachten Sie, dass für den Wifi-Betrieb an Stelle von USB-Betrieb die Firmware des Arduino Leonardo geändert werden muss.

Hier eine Anwendung des AR488 GPIB-Wifi Adapters an meinem HP8593E Spectrumalyzer, wobei die Stromversorgung des Adapter über die PS/2-Schnittstelle des Analyzers erfolgt:

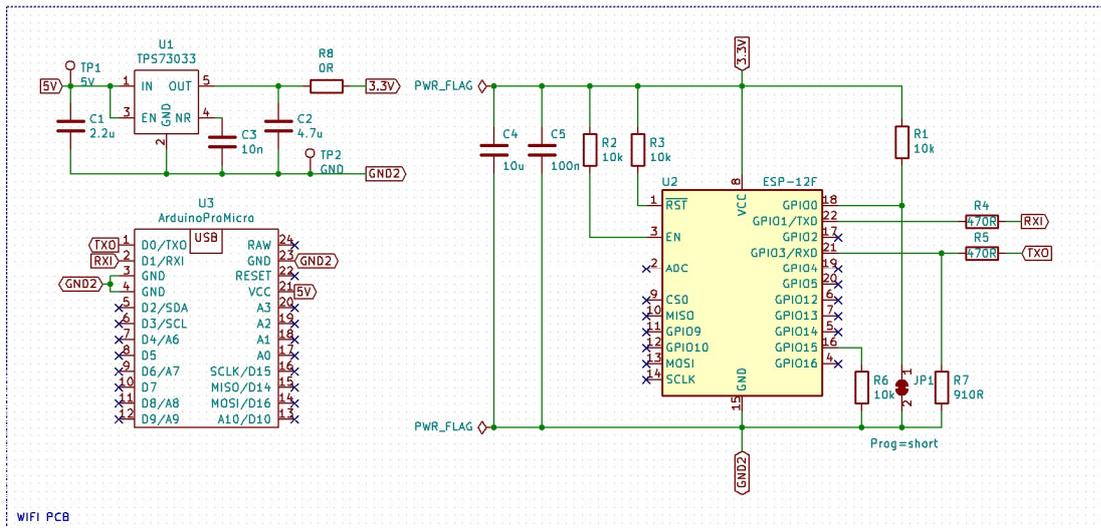


Bitte beachten Sie, dass das KE5FX Toolkit nicht vollständig kompatibel mit dem Wifi-Modus ist. Eine Zusatzsoftware (Redirector) behebt dieses Problem, die Details folgen unten ab Seite 27.

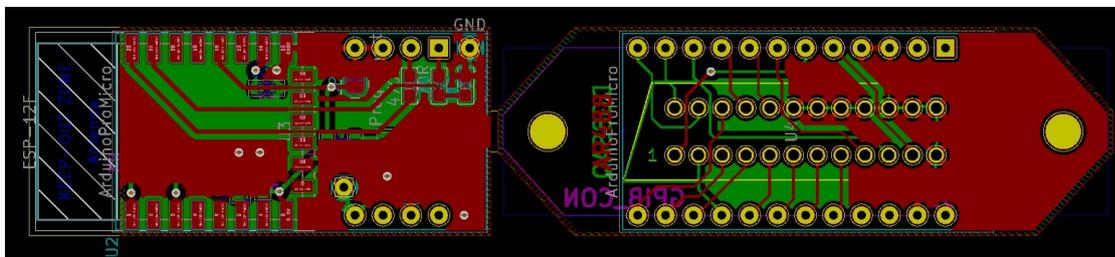
Schaltung des AR488 GPIB-USB Adapters:



Schaltung des Wifi-Boards:



Layouts:



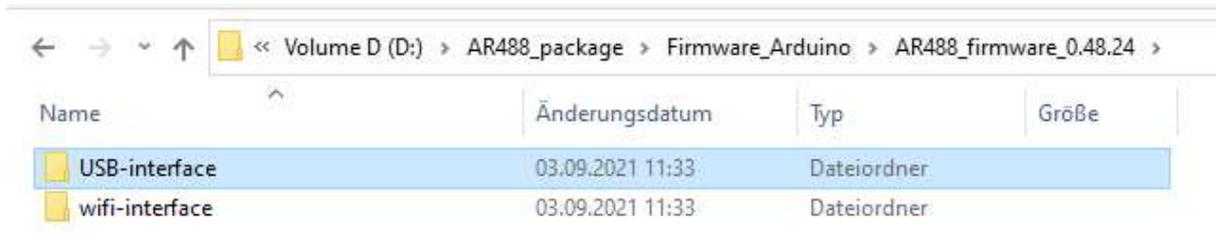
Beide Boards wurden aus Kostengründen als seine Platine gefertigt. Das KiCAD-Projekt und die Gerber-Dateien sind im folgenden Download-Paket enthalten:

<http://www.dg8saq.darc.de/hidden/DG8SAQ-AR488-package.zip>

Dasselbe Download-Paket enthält auch die notwendigen Firmwaren und meine Redirector-Software.

Programmierung des Adapters

Um die Firmware in den Arduino ProMicro zu flashen, ist die Arduino Entwicklungsumgebung zu installieren. Ich stelle zwei Firmware-Varianten V0.48.24 für den Arduino Leonardo ProMicro im Download-Paket zur Verfügung:

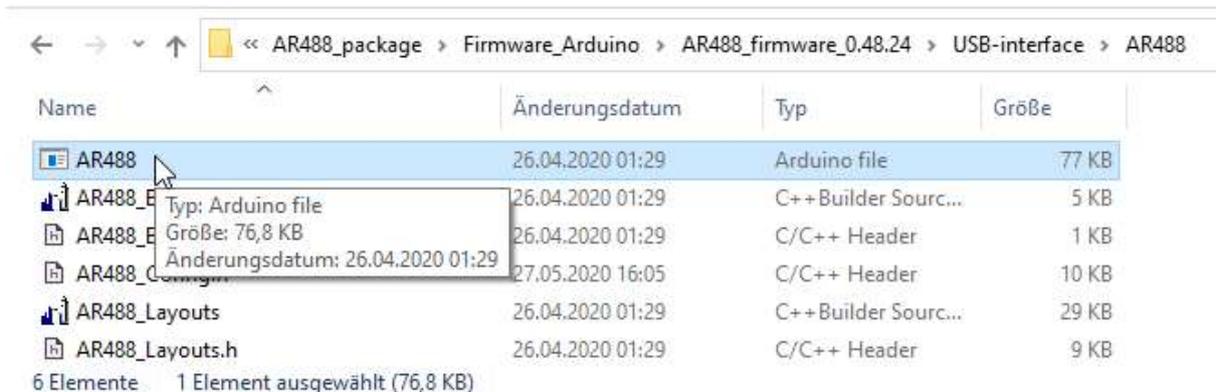


Die USB-Variante nutzt zur GPIB-Kommunikation mit dem PC die USB Schnittstelle über eine virtuelle serielle Schnittstelle auf dem PC.

Die Wifi-Variante nutzt stattdessen den UART auf dem Arduino Board mit 115200 Baud.

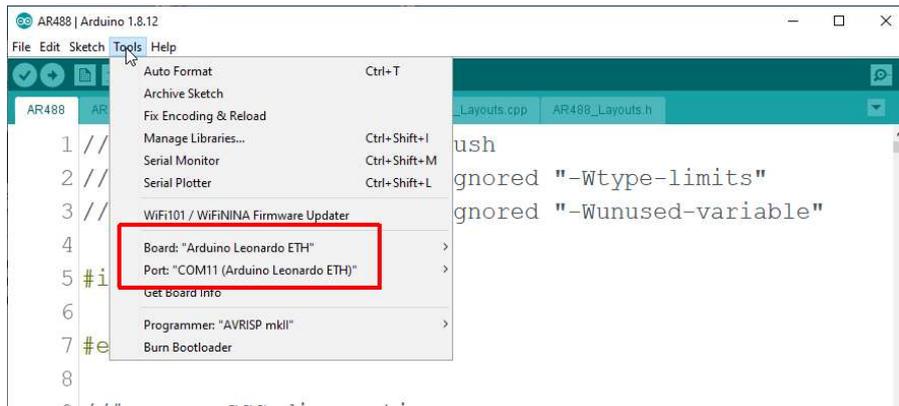
Es gibt inzwischen bei <https://github.com/Twilight-Logic/AR488> eine deutlich neuere Firmware, diese muss der Nutzer aber selbst an die entsprechende Hardware anpassen.

Um die Firmware in den Arduino zu flashen, wird dieser an einen USB-Port des PCs angeschlossen. Die Arduino-Entwicklungsumgebung muss hierfür installiert sein. Wählen Sie im Dateibrowser im entsprechenden Firmware-Verzeichnis die Datei AR488.ino aus...

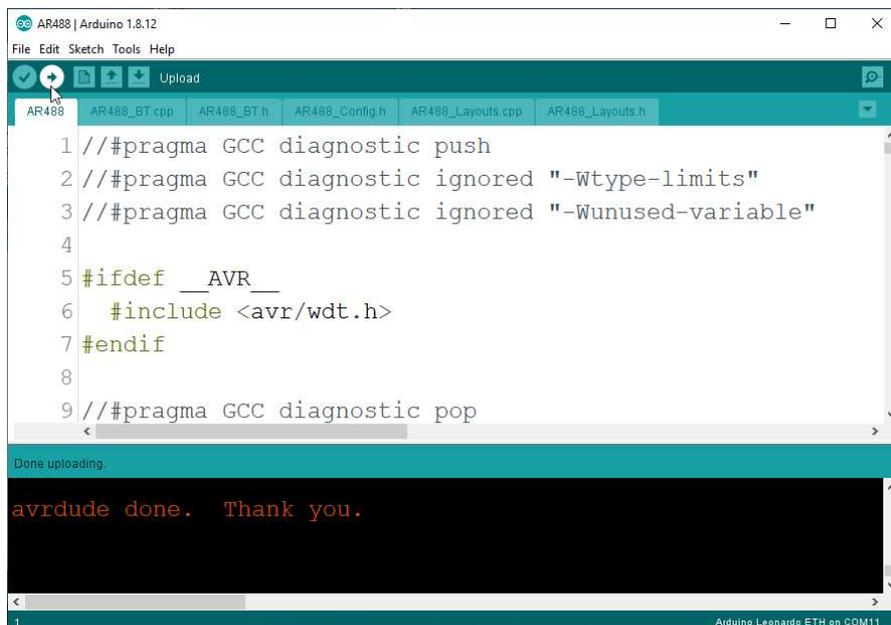


... und doppel-klicken Sie diese. Das Projekt wird dann automatisch mit der Arduino-Umgebung geöffnet.

Unter Tools sollte, wie unten gezeigt, für das Board „Arduino Leonardo ETH“ ausgewählt werden. Als Port muss der Port des Arduino gewählt werden. Den findet man leicht, weil sich das Board als „Arduino Leonardo ETH“ meldet, wie man leicht erkennt.

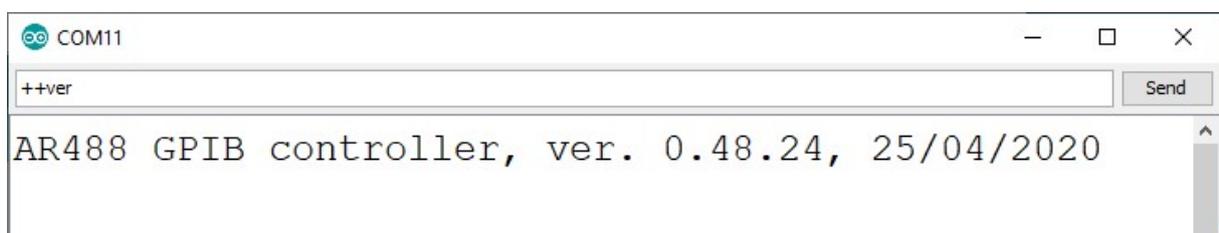


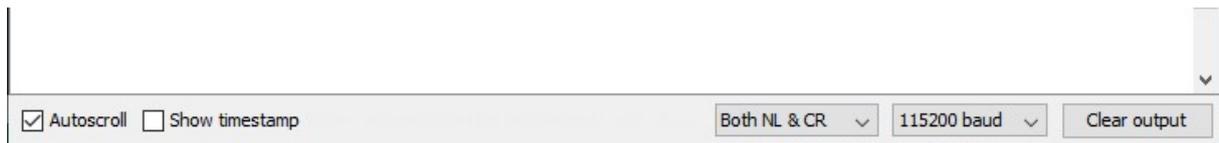
Nun kann man durch Klick auf den grünen Pfeil-Knopf links oben flashen:



Zunächst wird das Programm kompiliert (Fortschrittsbalken rechts unten), dann in den Arduino geschrieben (Meldungen unten).

Damit ist der GPIB-USB-Adapter prinzipiell schon betriebsbereit. Dies kann mit dem in der Arduino Entwicklungsumgebung eingebauten seriellen Terminal (=Serial Monitor) überprüft werden:





Achten Sie darauf, dass unten im Serial Monitor die richtigen oben gezeigten Einstellungen gewählt sind.

Wenn Sie in der oberen Textzeile `++ver` eingeben und die Return-Taste drücken, dann meldet sich der Adapter, wie oben gezeigt, mit seiner Versionsnummer. Damit wissen wir, dass er funktioniert.

Die KE5FX GPIB-Toolbox erwartet allerdings einen Adapter, der sich auf den `++ver`-Befehl mit Prologix meldet. Dies kann ebenfalls mit dem seriellen Monitor mit ein paar Befehlen eingestellt werden.

Man gibt nacheinander folgende Befehle ein, und schließt jeden durch Drücken der Return-Taste ab:

```
++setvstr Prologix GPIB-USB Controller version 6.100
++savecfg
++ver
```

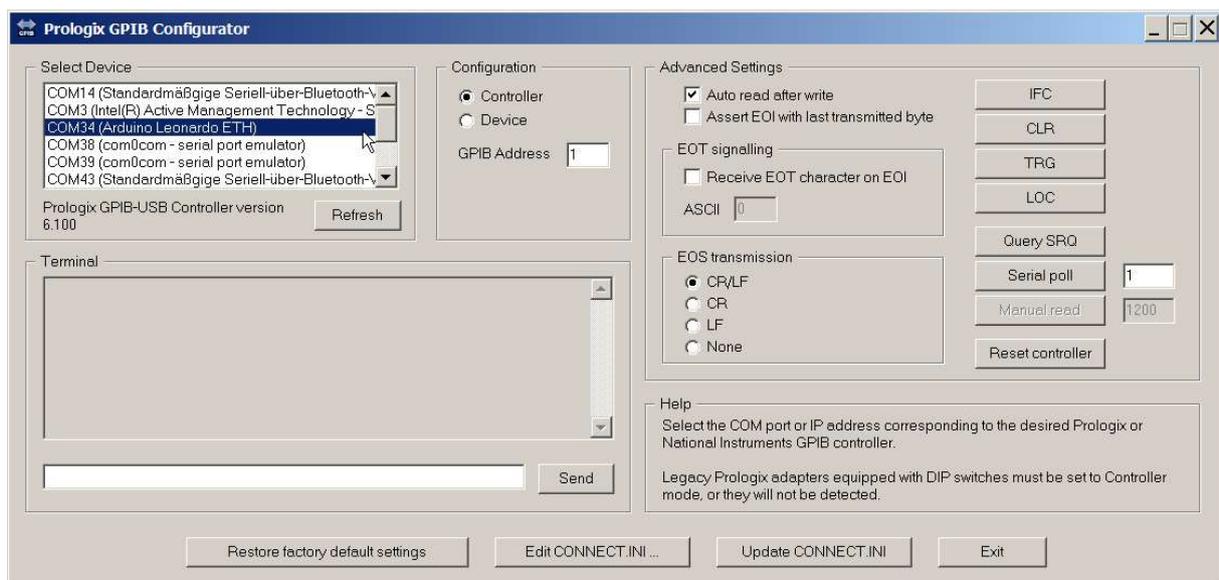
Nach dem letzten `++ver`-Befehl meldet sich der Adapter als Prologix-Adapter. Die tatsächliche AR488-Versionsnummer kann man immer noch mit dem Befehl „`++ver real`“ erhalten.

Damit ist der Adapter zur Benutzung mit der KE5FX GPIB Toolbox bereit.

Nutzung der KE5FX Toolbox

Funktionstest:

1. Downloaden und installieren Sie die KE5FX GPIB Toolbox.
2. Schließen Sie den AR488 Adapter an den Computer an.
3. Starten Sie den KE5FX GPIB Configurator.
4. Wählen Sie den AR488-Adapter mit Namen "Arduino Leonardo ETH" aus:

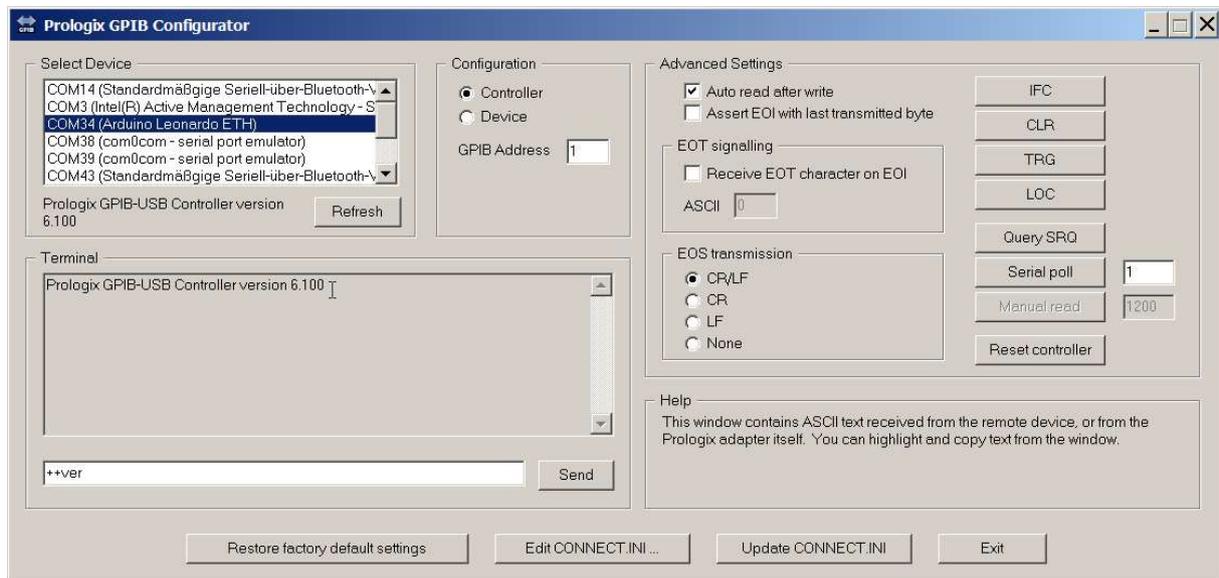


Wenn der "Send"-Knopf nicht ausgegraut ist, dann wurde der AR488 Adapter als Prologix Adapter erkannt.

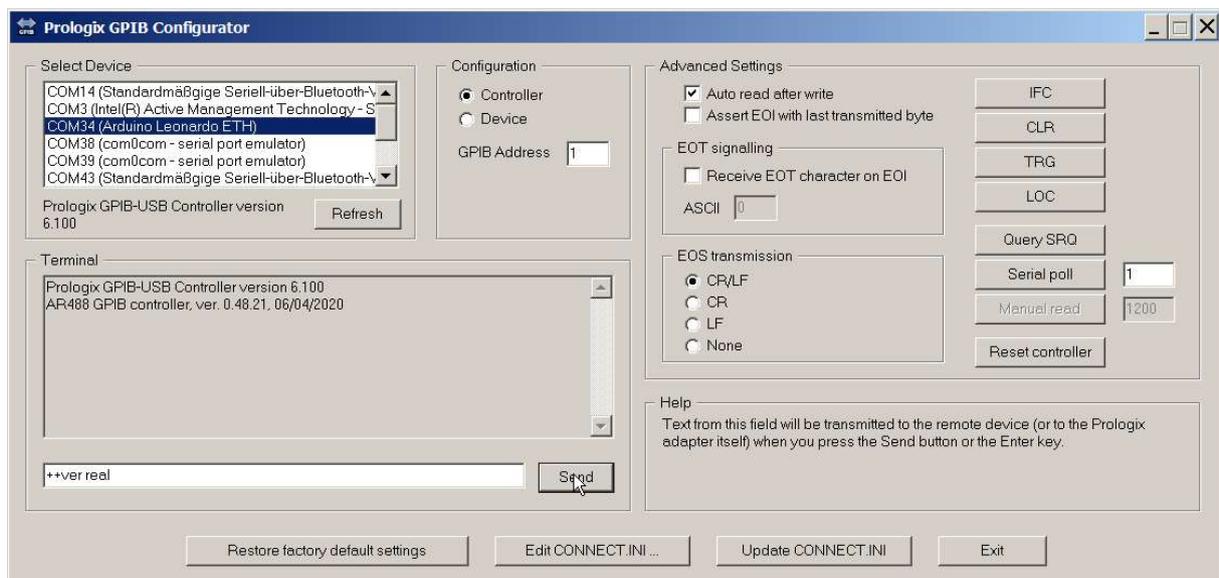
Stellen Sie für die folgenden Versuche sicher, dass rechts "Auto read after write" aktiviert ist.

5. Tippen Sie „++ver“ in das Send Feld und klicken Sie auf den Send Knopf:

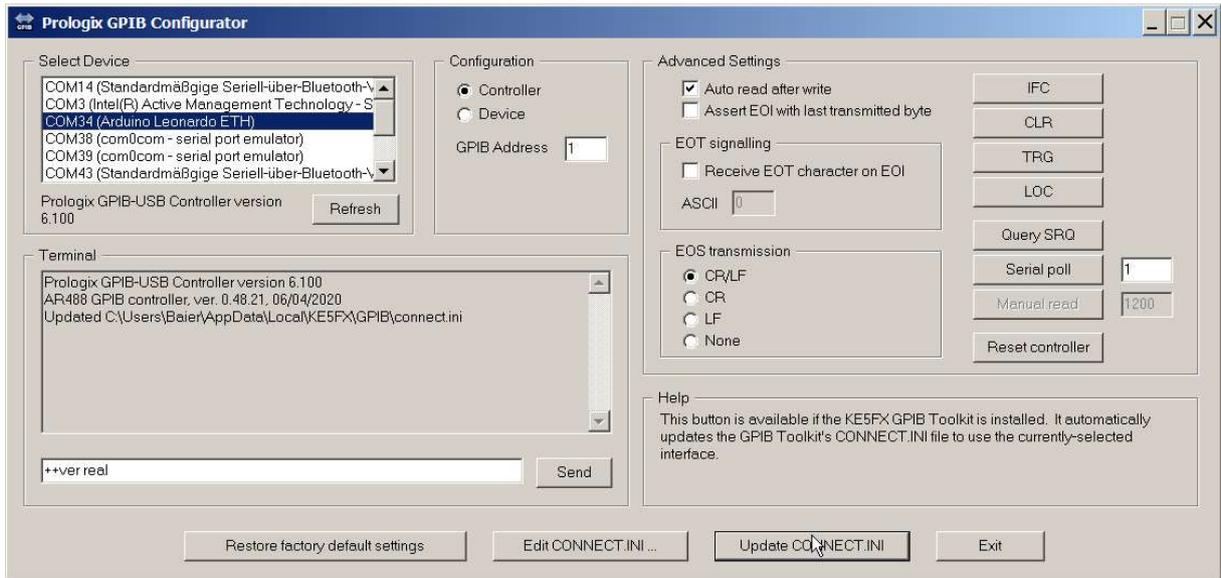
Nach einer kleinen Verzögerung sollte sich der Adapter wie gezeigt als Prologix im Terminal-Feld melden:



Die tatsächliche Versionsnummer erhalten wir mit dem Befehl „++ver real“:



6. Nun drücken Sie unten den “Update CONNECT.INI” Knopf, damit alle KE5FX-Programme den Adapter kennen lernen:



Jetzt schließen Sie den Configurator.

Auslesen der Bildschirmansicht von einem HP8593E Spektrumanalyzer:

Wichtig:

Der GPIB Configurator darf jetzt NICHT LAUFEN, sonst blockiert er die serielle Schnittstelle des GPIB Adapters.

1. Stellen Sie sicher, dass der GPIB Adapter bereits über das USB-Kabel mit Strom versorgt wird.
2. Jetzt schalten Sie den Spektrumanalyzer ein, an den der Adapter angeschlossen ist.

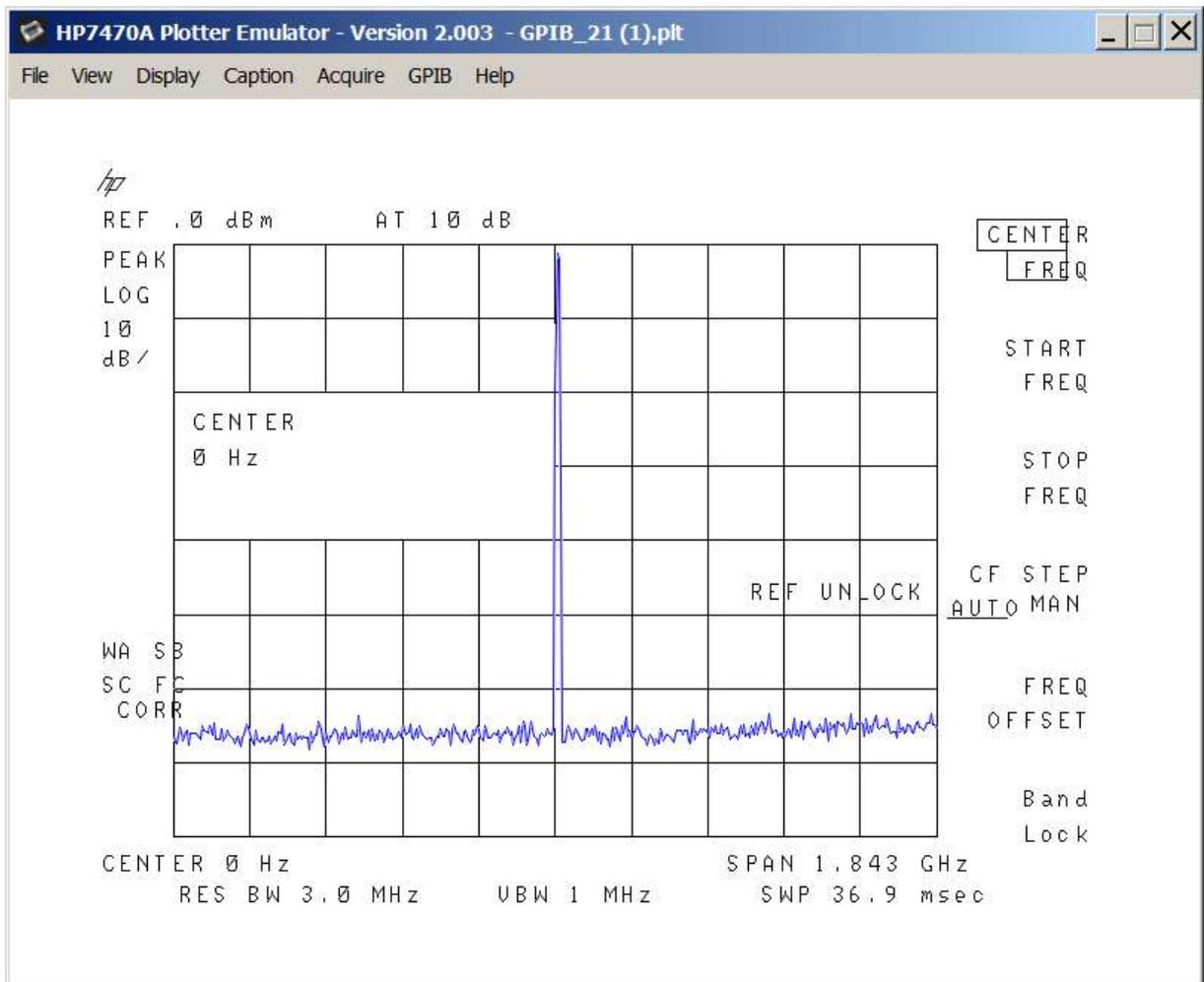
Beim Hochfahren zeigt der Analyzer kurz seine GPIB Adresse (HP-IB Adresse genannt). Notieren Sie diese. Mein Gerät besitzt die Adresse 21.

3. Starten Sie den HP 7470A Emulator aus der KE5FX Toolbox.
4. Wählen Sie "Acquire" and "Request Plot from supported device at address **", wobei * für Ihre GPIB Adresse steht, bei mir also 21:

Device Name	GPIB Address	Function Key
Tektronix 2430A / 2432A / 2440A	4	F2
HP 3561A	11	F7
HP 3585A/B (via HP-GL/2 emulation)	11	F3
Tektronix 49x	3	F4
HP 8566B / 8567B / 8568B	18	F6
HP 8566A / 8567A / 8568A (via HP-GL/2 emulation)	18	F8
HP 8510 or 8753 series	16	F9
HP 54201A	5	F12

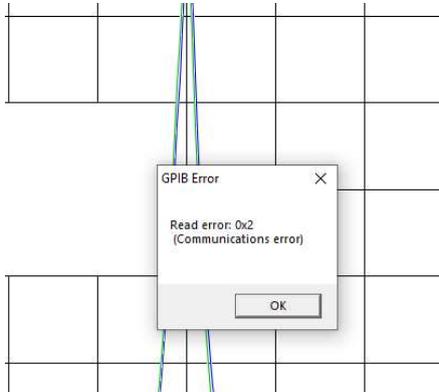
Request plot from supported device at address 1
Request plot from supported device at address 2
Request plot from supported device at address 3
Request plot from supported device at address 4
Request plot from supported device at address 5
Request plot from supported device at address 6
Request plot from supported device at address 7
Request plot from supported device at address 8
Request plot from supported device at address 9
Request plot from supported device at address 10
Request plot from supported device at address 11
Request plot from supported device at address 12
Request plot from supported device at address 13
Request plot from supported device at address 14
Request plot from supported device at address 15
Request plot from supported device at address 16
Request plot from supported device at address 17
Request plot from supported device at address 18
Request plot from supported device at address 19
Request plot from supported device at address 20
Request plot from supported device at address 21
Request plot from supported device at address 22
Request plot from supported device at address 23

Nach einer kurzen Wartezeit erscheint das Spektrum auf dem Computerbildschirm:



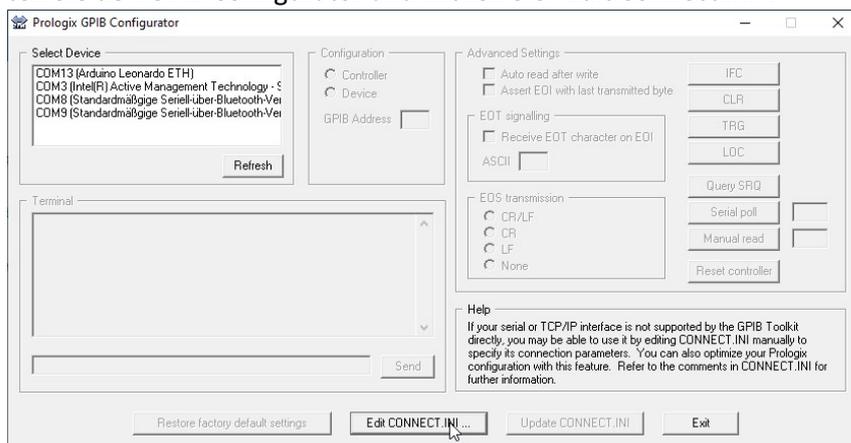
Read error: 0x2

Bei manchen Systemen liest die KE5FX Plotter Software das Schirmbild nur unvollständig oder überhaupt nicht und liefert die Fehlermeldung „Read error 0x2“:



Hier konnte ich das Problem wie folgt beheben:

Starten Sie den GPiB Configurator and klicken Sie "Edit Connect.ini":



Finden Sie die Zeile "write_delay_ms 0" und ändern Sie den Wert von 0 auf 100:

```
connect.ini - Editor
Datei Bearbeiten Format Ansicht Hilfe
;that will not be understood by the adapter or instrument in use
;
is_Prologix 1
;
;Some older Prologix boards may need a delay after writes to avoid
;buffer-overflow problems. Use 0 milliseconds for no delay
;
write_delay_ms 100
;
;Prologix controllers can reset the device to local operation when the
;GPiB connection is closed, but since most GPiB Toolkit applications use the
;Prologix adapter in auto-read mode (++auto 1), the final ++loc command has
;the effect of addressing the instrument to talk. This can cause error
;messages or connection problems with some equipment. You can avoid
;this behavior by setting reset_to_local to 0 to avoid transmitting
```

Speichern Sie die Datei und probieren Sie die Plotter Software erneut.

Konfiguration des AR488 Adapters

Ein neuer AR488 Adapter antwortet auf den „++ver“ Befehl stets mit dem AR488 String. Damit der mit der KE5FX Toolbox arbeitet, muss er aber darauf mit einem Prologix-String antworten. Dies kann mit dieser Befehlsfolge dauerhaft einprogrammiert werden:

```
++setvstr Prologix GPIB-USB Controller version 6.100  
++savecfg
```

Der erste Befehl ändert den Versionsstring, der zweite schreibt ihn dauerhaft ins EEPROM, so dass er auch nach dem Aus- und Wiedereinschalten des Adapters erhalten bleibt.

Manche GPIB Instrumente nutzen nicht CR+LF als Stringterminatoren. Der AR488 Adapter kann umkonfiguriert werden, sodass er mit anderen Terminatoren arbeitet:

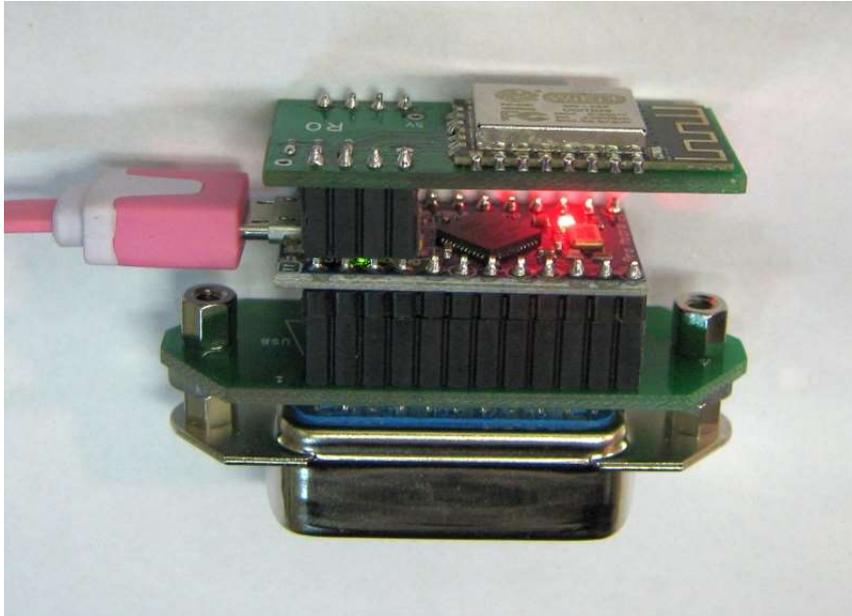
```
++eor <n>  
++savecfg
```

Hier steht <n> für eine Zahl, welche die Terminatorsequenz spezifiziert:

<n>	Sequence	Hex
0	CR + LF	0D 0A
1	CR	0D
2	LF	0A
3	None	N/A
4	LF + CR	0A 0D
5	ETX	03
6	CR + LF + ETX	0D 0A 03
7	EOI signal	N/A

Drahtloser Betrieb

Um den AR488 Adapter drahtlos zu benutzen, muss auf den Adapter ein Funkmodul (Bluetooth, W-LAN) montiert werden. Im Folgenden beschreibe ich die Benutzung meines ESP8266 W-LAN Moduls. Dieses kann aufgesteckt oder aufgelötet werden.



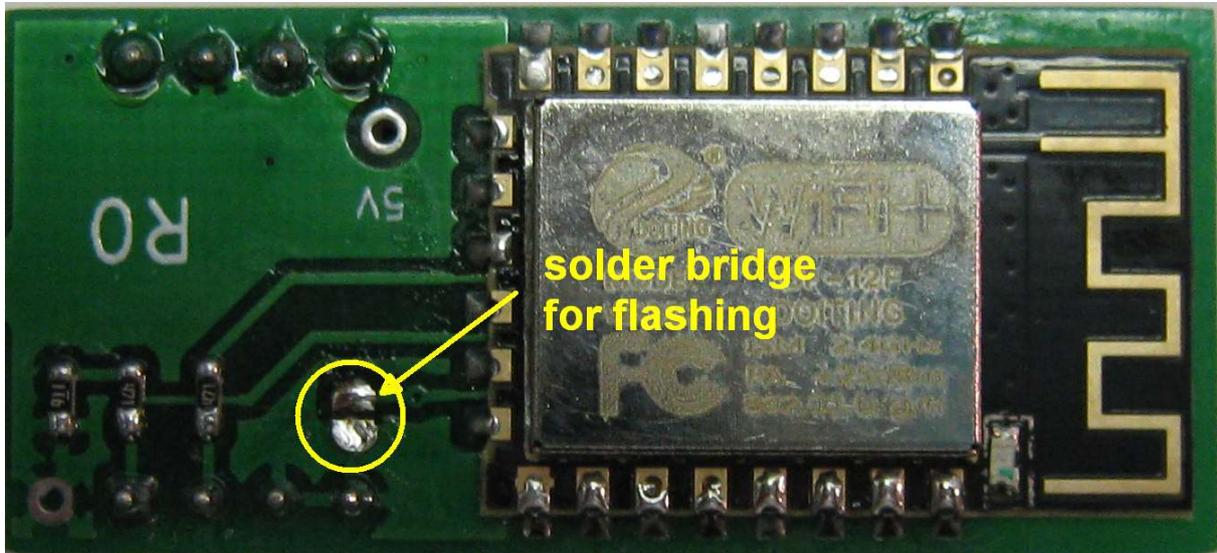
Bei meinem ersten Prototypen oben wurden alle Platinen mit Pfostensteckern aufeinander gesteckt.

Wichtig:

- Bitte beachten Sie, dass der AR488 Adapter auch mit aufgestecktem Wifi-Modul über USB betrieben werden kann. Das Wifi-Modul erhöht allerdings den Stromverbrauch.
- Der Arduino Leonardo benötigt unterschiedliche Firmwares, je nachdem, ob er über USB oder W-LAN arbeiten soll. Wenn Sie die Betriebsart wechseln, müssen Sie also den Arduino umprogrammieren.
- Auch das Wifi-Modul benötigt eine eigene Firmware. Diese kann über einen handelsüblichen USB-Seriell-Wandler einprogrammiert werden. Alternativ kann man aber auch den Arduino des aufgebauten AR488 Adapters benutzen, um das fertig montierte Wifi-Modul zu flashen. Die Prozedur ist allerdings etwas komplizierter und wird im Folgenden beschrieben.

Programmieren des ESP8266 Wifi Moduls durch den Arduino hindurch

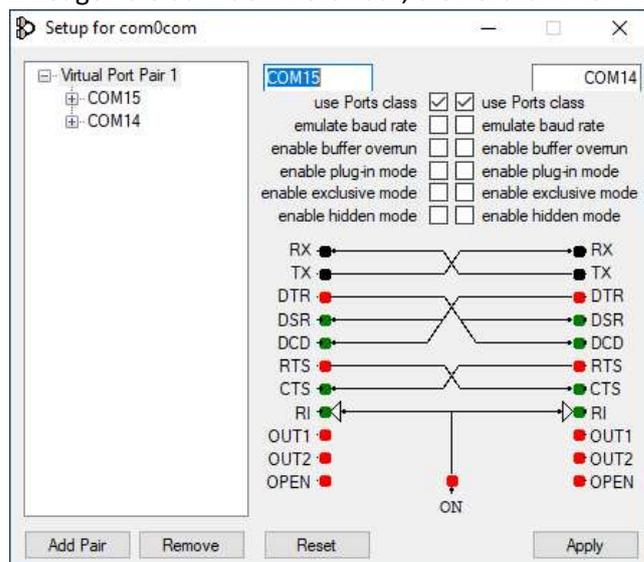
1. Bringen Sie den ESP8266 in den Programmiermodus, indem Sie die unten markierte Lötbrücke schließen:



In obigem Bild ist die Lötbrücke offen für normalen Betrieb. Für die Programmierung muss diese GESCHLOSSEN sein!

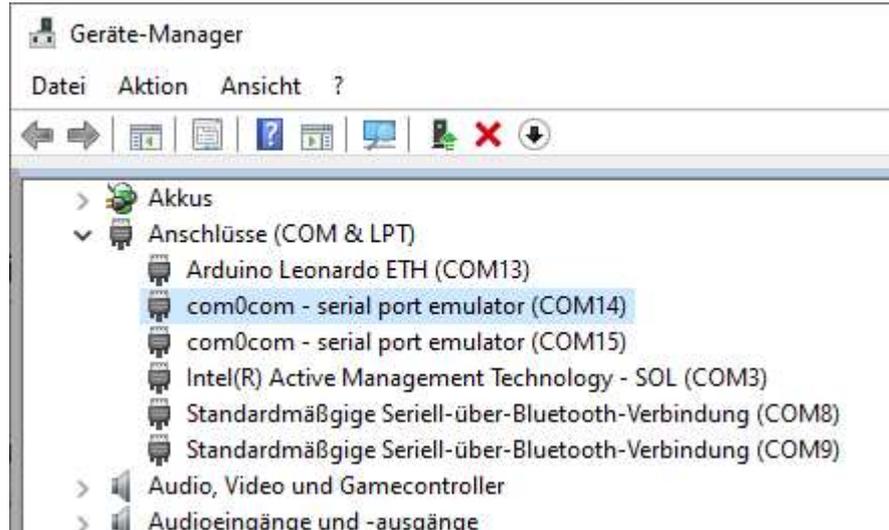
2. Flashen Sie den Arduino Leonardo über die Arduino Entwicklungsumgebung mit der Firmware `leo_usb2serial.ino` aus dem Download-Paket. Diese verwandelt den Arduino in einen USB-Seriell-Wandler. Der Flash-Prozess ist auf Seite 5 beschrieben.
3. Downloaden und installieren Sie die Software `com0com`, die ein virtuelles serielles Null-Modem Kabel erzeugt. Download Link: <https://sourceforge.net/projects/signed-drivers/files/com0com/v3.0/>
Benutzen Sie die passende Variante für Ihre Windows Version (32bit oder 64bit).

Erzeugen Sie damit ein Port-Paar, die Portnummern werden automatisch vergeben:



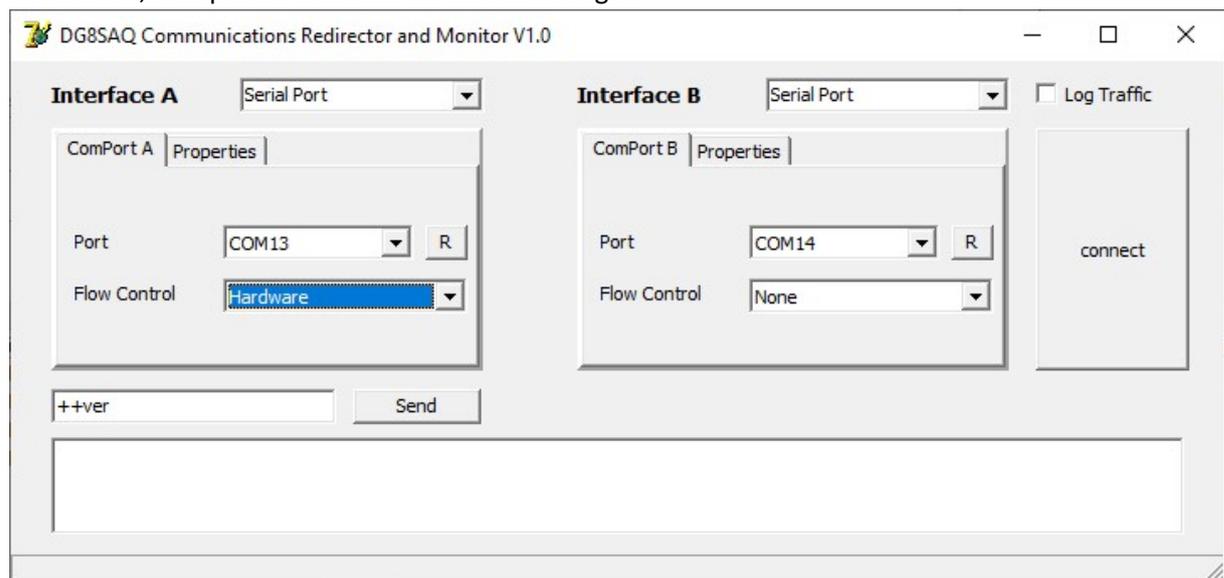
Die so erzeugten virtuellen Ports sind miteinander verbunden, d.h. was man in COM15 hineinschickt, kommt zum COM 16 wieder heraus und umgekehrt. Wir benutzen dies, um das Flasher Tool mit dem Arduino zu verbinden.

Die neu erzeugten Ports sind auch im Gerätemanager sichtbar:



Auf meinem System hängt der Arduino Leonardo an COM13 und das virtuelle Nullmodem-Kabel besitzt die Anschlüsse COM14 und COM15, wie im Gerätemanager leicht zu sehen ist.

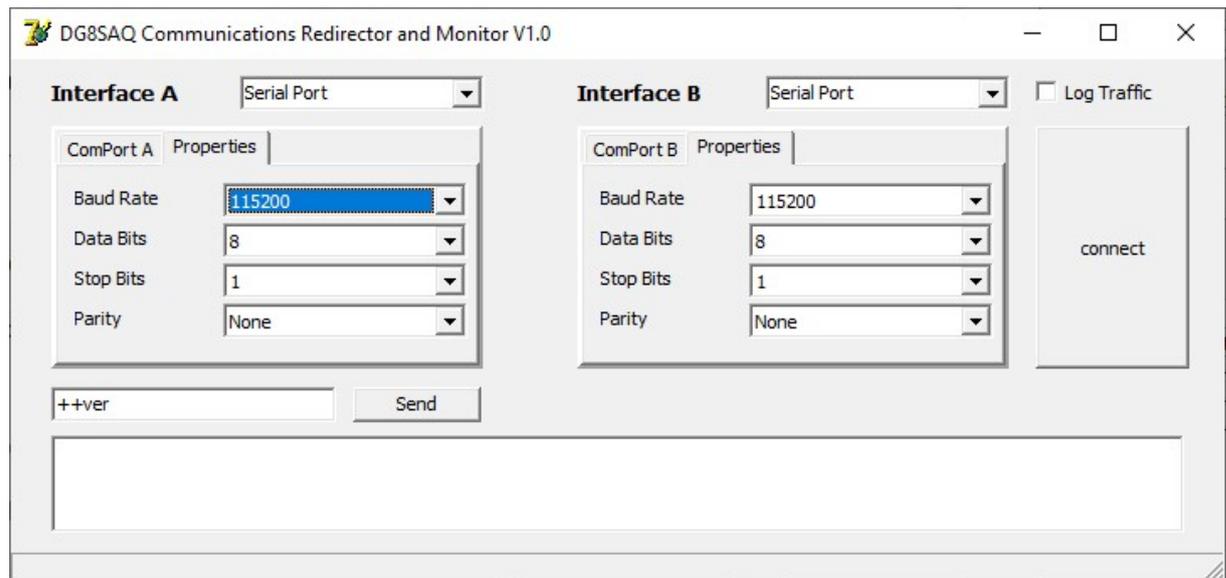
4. Starten Sie meine Software "redirector.exe" aus dem Download-Paket, um den Leonardo COM13 mit dem ersten Nullmodemanschluss COM14 zu verbinden. Ihre Portnummern können abweichen, überprüfen Sie diese im Gerätemanager.



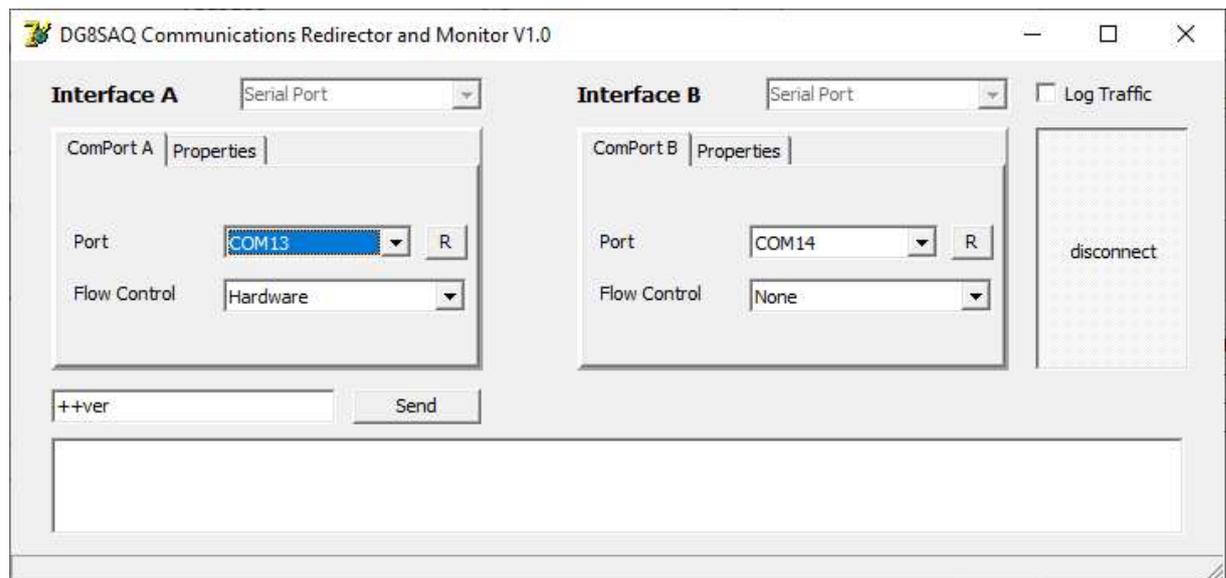
Wichtig:

- Flow Control auf der Arduino Seite muss "Hardware" sein, auf der com0com Seite „None“! Der Redirector und com0com werden einzig und alleine gebraucht, um die unterschiedlichen Flow Controls aneinander anzupassen. Falls jemand eine Arduino USB-Seriell-Firmware kennt, die das einfacher behebt, wäre ich für die Info dankbar.
- Die "Log Traffic" Box DARF NICHT aktiviert sein!

Weitere Einstellungen:



Drücken Sie jetzt den "Connect" Knopf um COM13 mit COM14 zu verbinden:



Hierbei darf keine Fehlermeldung erscheinen. Jetzt ist der com0com Port COM15 durchverbunden über COM14 zum COM13 des Leonardo.

5. Downloaden Sie das NodeMCU Flasher Tool von hier:

32 bit Windows:

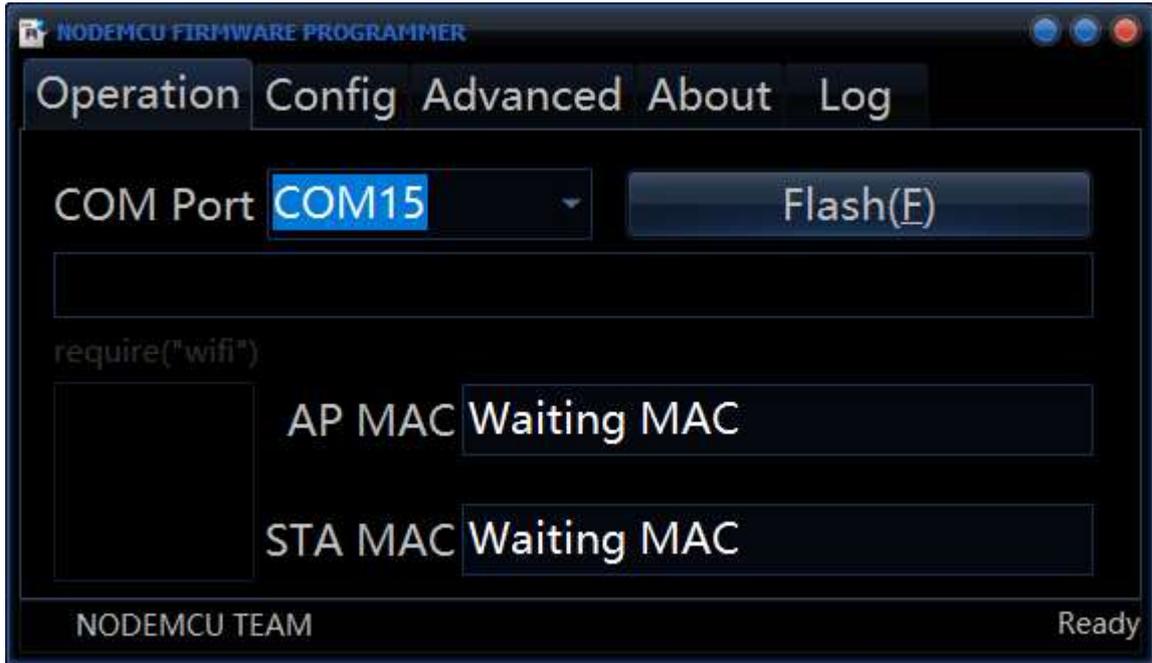
<https://github.com/nodemcu/nodemcu-flasher/tree/master/Win32/Release>

64 bit Windows:

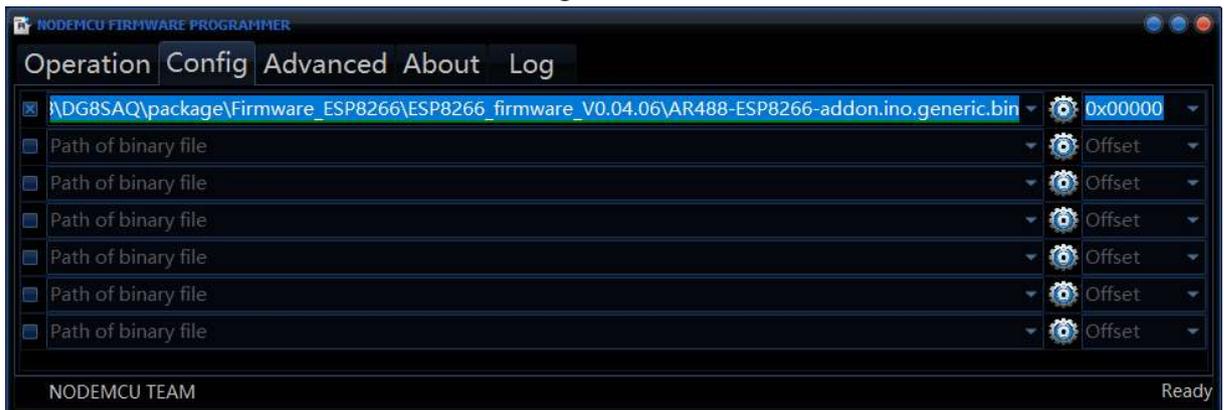
<https://github.com/nodemcu/nodemcu-flasher/tree/master/Win64/Release>

Den Flasher muss man nicht installieren, man kann ihn einfach starten, wie auch mein „redirector“-Programm.

6. Starten Sie den Flasher und wählen Sie den freien com0com port, hier COM15:



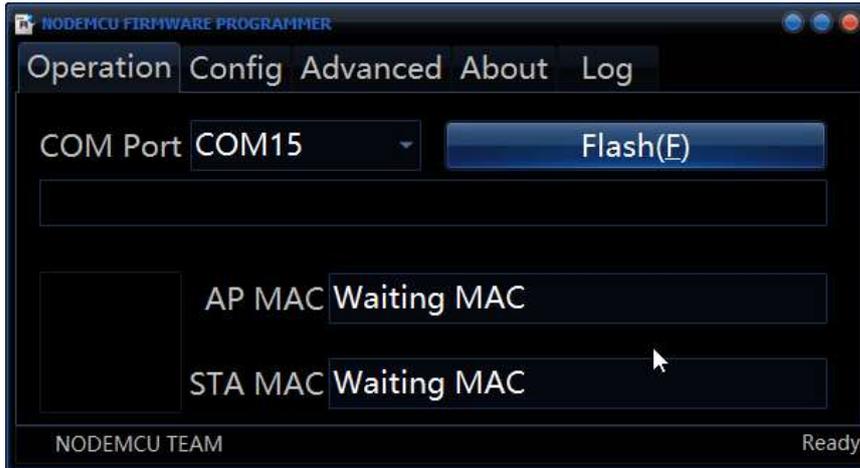
7. Wählen Sie die ESP8266 AR488-Firmware, die geflasht werden soll:



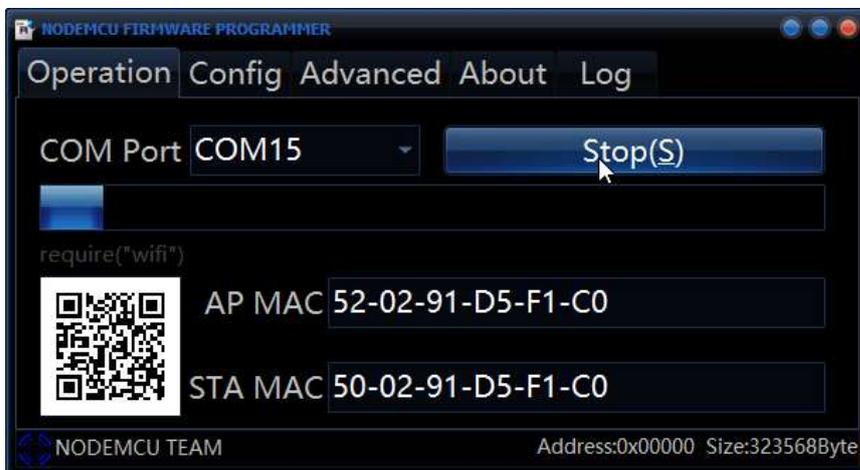
8. Advanced Options für den ESP-01:



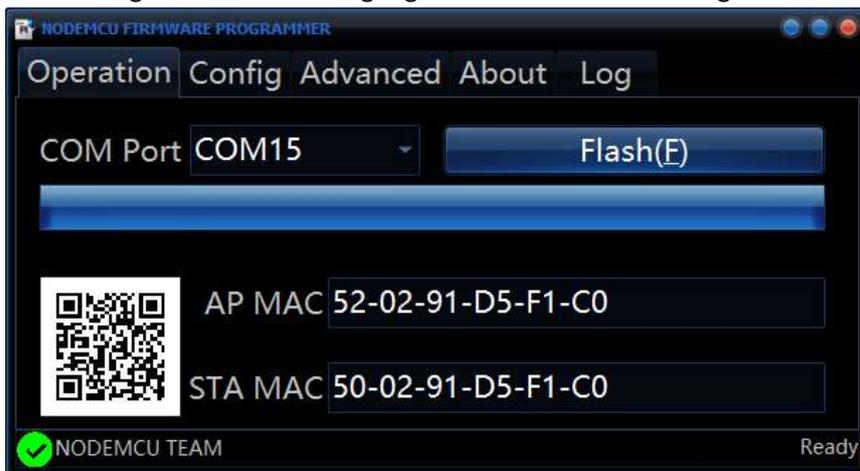
9. Jetzt drücken Sie den FLASH Knopf:



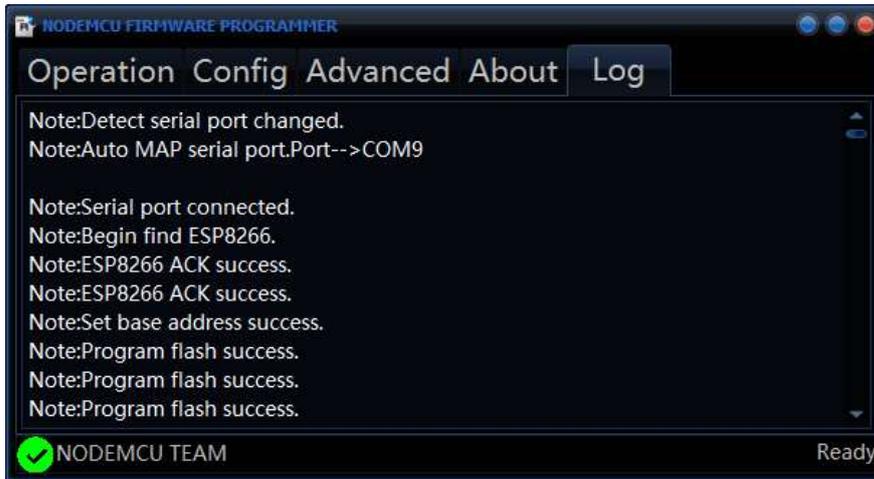
10. Beobachten Sie den Flash-Fortschritt:



11. Nach erfolgreichem Flash-Vorgang leuchtet unten links das grüne Haken-Symbol:



12. Auch am Log sieht man, dass alles gut gegangen ist.



Schließen Sie das Flasher-Tool und das redirector Program.

13. Nun muss der Arduino mit der AR488 Wifi Firmware programmiert werden (s. Seite 5):

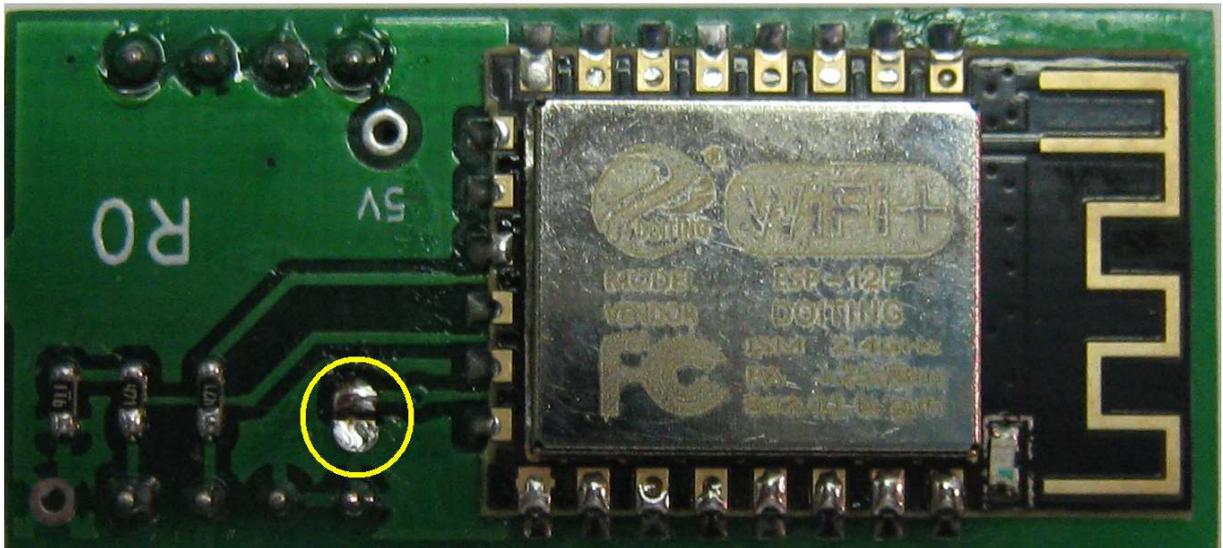


Danach reagiert der Adapter nicht mehr auf GPIB Kommandos über USB, er kann aber nach wie vor über USB umprogrammiert werden.

Hinweis: Die Arduino Firmwares für USB und Wifi unterscheiden sich nur in einer Zeile, die festlegt, ob der AR488 über den USB (CDC) Port oder das Hardware UART kommuniziert:



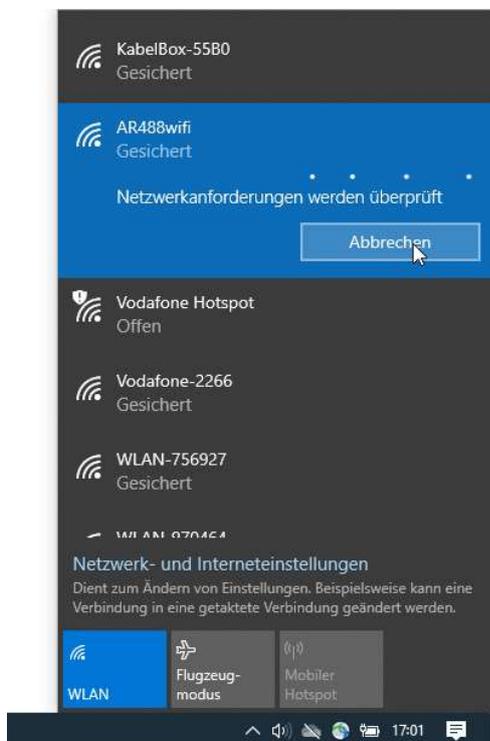
14. Entfernen Sie den AR488 vom Computer und entfernen Sie die Lötbrücke auf dem ESP8266 Adapter, um ihn vom Programmiermodus zurück in den normalen Betriebsmodus zu bringen:



Jetzt kann der Adapter Online gehen.

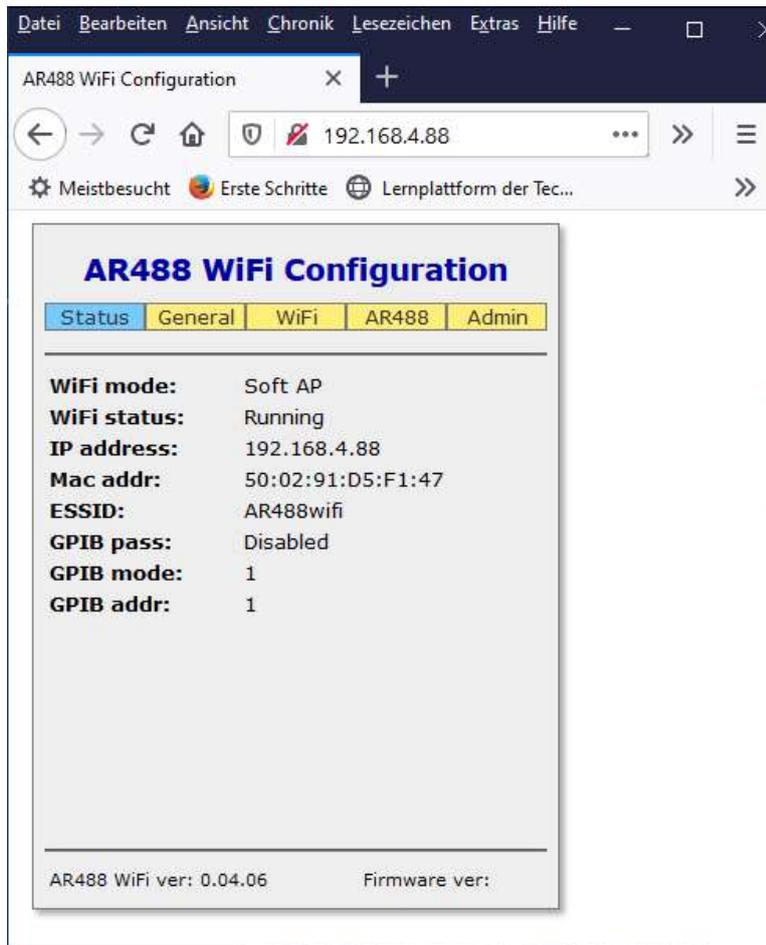
Konfigurieren des AR488 Wifi Adapters

Schalten Sie jetzt den Adapter ein, z.B. indem Sie ihn über das USB-Kabel mit Spannung versorgen. Nach kurzer Wartezeit sollte der ESP8266 als W-LAN Access-Point erscheinen:



Verbinden Sie sich mit diesem. Mein Windows 10 sagt nie, dass es verbunden ist, aber die Verbindung funktioniert trotzdem. Falls Sie Internet über W-LAN beziehen, fällt an diesem Punkt die Internetverbindung zeitweilig aus.

Jetzt können Sie sich mittels eines beliebigen Web-Browsers (z.B. Firefox) mit AR488 Adapter verbinden. Geben Sie dazu in das Adressfeld die IP-Adresse 192.168.4.88 des Adapters ein:



Im Prinzip können Sie den Adapter bereits so benutzen, aber vermutlich haben Sie jetzt keinen Internetzugang mehr. Daher ist es komfortabler, den Adapter mit Ihrem W-LAN Netz zu Hause zu verbinden. Dazu benötigen Sie die Adresse Ihres W-LAN Routers. Diese finden Sie wie folgt heraus:

Verbinden Sie Ihren Computer wieder mit Ihrem Haus-Netzwerk.

Starten Sie das Windows-Programm ipconfig in einem Konsolenfenster. Hier können Sie die gesuchte Adresse identifizieren.

```
C:\Windows\System32\cmd.exe
Verbindungslokale IPv6-Adresse . . : fe80::3d44:be6c:4721:ac10%22
IPv4-Adresse . . . . . : 192.168.111.1
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . :

Ethernet-Adapter VMware Network Adapter VMnet8:

Verbindungsspezifisches DNS-Suffix:
Verbindungslokale IPv6-Adresse . . : fe80::21f8:be08:94af:7359%25
IPv4-Adresse . . . . . : 192.168.217.1
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . :

Drahtlos-LAN-Adapter WiFi:

Verbindungsspezifisches DNS-Suffix: home
IPv6-Adresse. . . . . : 2a02:810d:a80:889:f826:d4f8:45e2:d779
Temporäre IPv6-Adresse. . . . . : 2a02:810d:a80:889:e942:1987:d32c:b7f7
Verbindungslokale IPv6-Adresse . . : fe80::f826:d4f8:45e2:d779%13
IPv4-Adresse . . . . . : 192.168.0.5
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . : fe80::3a43:7dff:fe74:2ad1%13
                            192.168.0.1

Ethernet-Adapter Bluetooth-Netzwerkverbindung 3:

Medienstatus. . . . . : Medium getrennt
Verbindungsspezifisches DNS-Suffix:

c:\Exchange\GPIB_AR488\DG8SAQ\package>
```

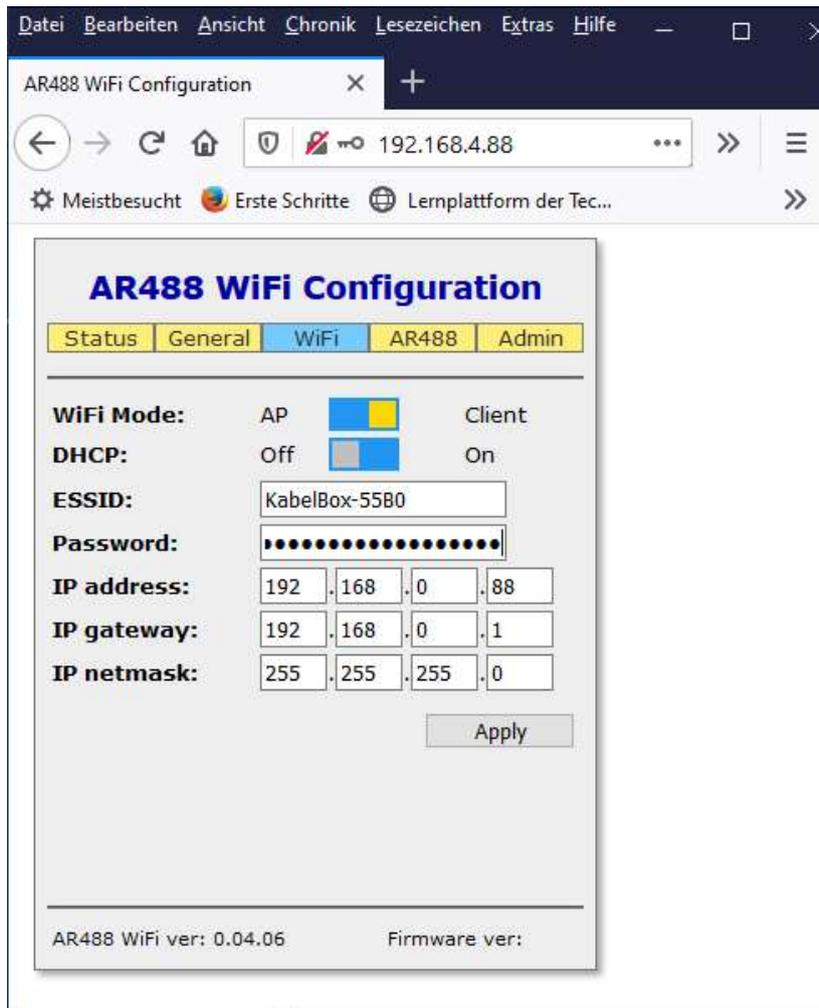
Der W-LAN Adapter des Computers besitzt, die IPv4 Adresse 192.168.0.5, wie oben ablesbar.

Der Router muss also die Adresse 192.168.0.1 besitzen, da Router immer die niedrigste Adresse im Netzwerk haben.

Um unseren AR488 Adapter später im Netzwerk einfach finden zu können, weisen wir ihm eine feste IPv4-Adresse zu, hier 192.168.0.88. Beachten Sie, dass die Basisadresse 192.168.0. identisch mit der Routeradresse sein muss. Die von Ihnen vergebene feste Adresse darf außerdem vom Router nicht bereits andersweitig vergeben sein.

Jetzt können wir den AR488 Adapter konfigurieren:

Verbinden Sie sich erneut mit dem Adapter Access-Point und verbinden Sie sich damit wie zuvor mittels Web-Browser:



Das obige Bild zeigt bereits die geänderten Einstellungen. Wir wechseln von AP (=Access Point) zu Client. Bei fester IP-Adresse bleibt DHCP aus und wir tragen Namen und Passwort unseres Heimnetzes ein. In die Adresszeile tragen wir die Wunschadresse, das Gateway ist unser Router.

Vergessen Sie nicht am Ende auf „Apply“ zu klicken!

Danach schaltet sich der Adapter in den Client-Modus und verbindet sich innerhalb von wenigen Minuten mit Ihrem drahtlosen Hausnetzwerk. Nun sollten Sie den Adapter über Ihr Hausnetzwerk unter der neu vergebenen Adresse (hier 192.168.0.88) erreichen können:



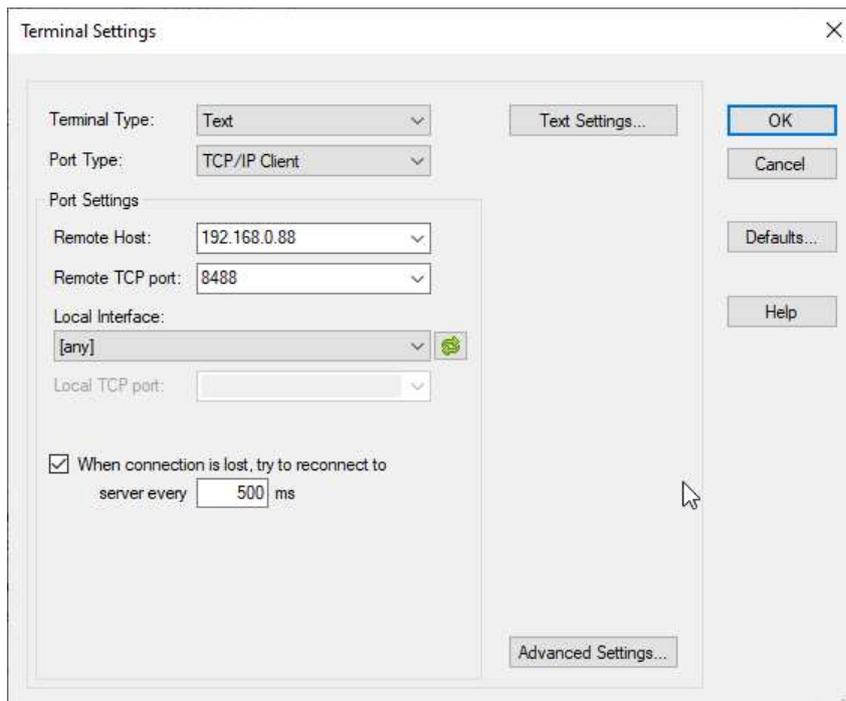
Schalten Sie in der "General"-Rubrik GPIB Pass = On ein und klicken Sie auf „Apply“. Jetzt werden GPIB-Kommandos vom ESP8266 an den Arduino weiter geleitet.

WARNUNG: Aktivieren Sie weder SSL noch benutzen Sie die Admin Funktionen!

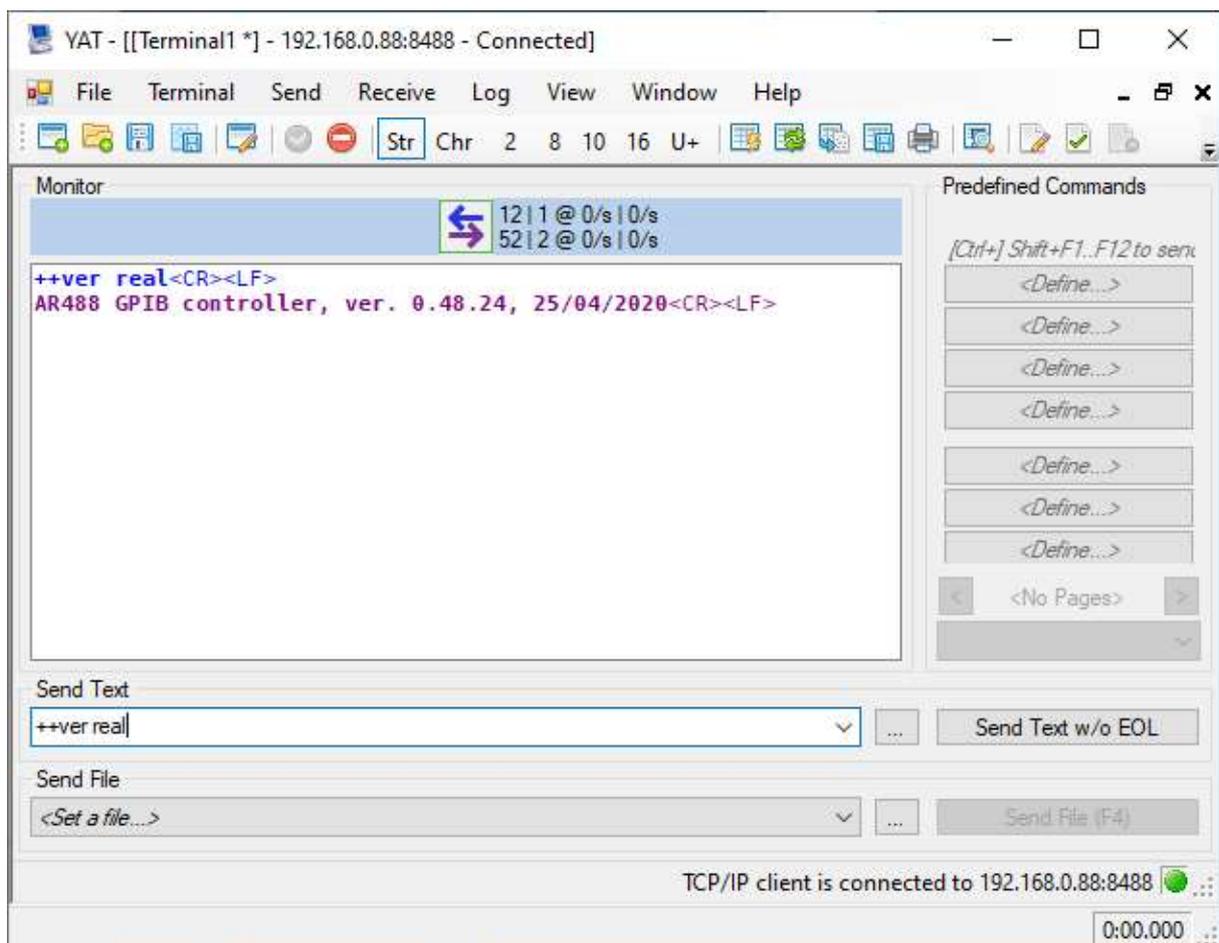
Wir können die ordnungsgemäße Funktion des Adapters jetzt mit dem YAT Terminalprogramm (oder einem ähnlichen) kontrollieren. YAT ist hier verfügbar:

<https://sourceforge.net/projects/y-a-terminal/>

Konfigurieren Sie YAT als TCP/IP Client mit der IP-Adresse des AR488 (hier 192.168.0.88) und der in obiger Maske festgelegten Portnummer (8488):



Bauen Sie jetzt die Verbindung auf und geben Sie den Befehl „++ver real“ ein. Nach Drücken der Return-Taste antwortet der Adapter drahtlos mit seiner Versionsnummer:



Wir können nun also mit einem GPIB-Gerät vollkommen drahtlos über das Hausnetzwerk kommunizieren.

Konfigurieren der KE5FX Plotter Software für drahtlosen Zugang

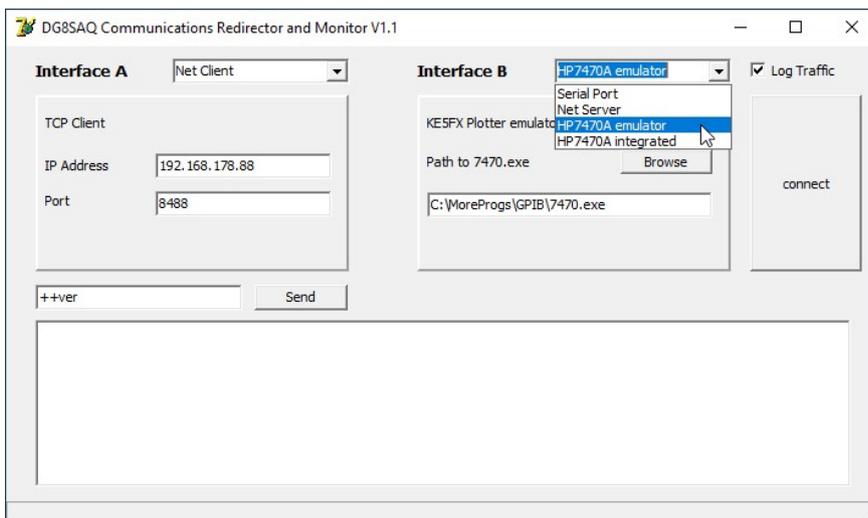
Prinzipiell kann die KE5FX Plottersoftware direkt über W-LAN mit dem GPIB Adapter kommunizieren. Dies erreicht man, indem man in *connect.ini* folgende Zeile aktiviert:

```
interface_settings 192.168.0.88:8488
```

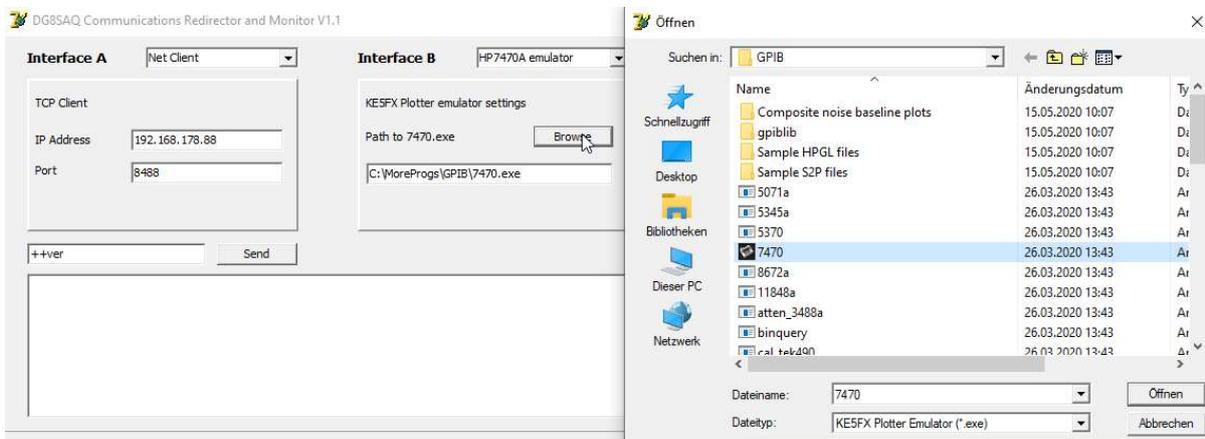
Hierbei ist 192.168.0.88 die Adresse des ESP8266 Chips and 8488 dessen Portnummer, beides haben wir oben selbst festgelegt.

“*Prinzipiell*” bedeutet, dass es bei mir nicht funktioniert. Ich vermute ein Time-out Problem als Ursache dafür. Der Verbindungsaufbau zum ESP8266 Chip dauert vermutlich zu lange und die Plotter Software gibt zu früh auf.

Ich habe das Problem durch Einfügen einer Software-Zwischenschicht mit meinem Redirector-Programm gelöst. Der Redirector kann auf der Eingangsseite als Netzwerk-Client fungieren und ständig die Verbindung zum ESP8266 aufrechterhalten. Damit fällt die Wartezeit für den Verbindungsaufbau weg. Die Ausgangsseite kann entweder wie schon oben auf einen com0com Port umgeleitet werden, oder direkt auf die KE5FX Plotter Software zugreifen:

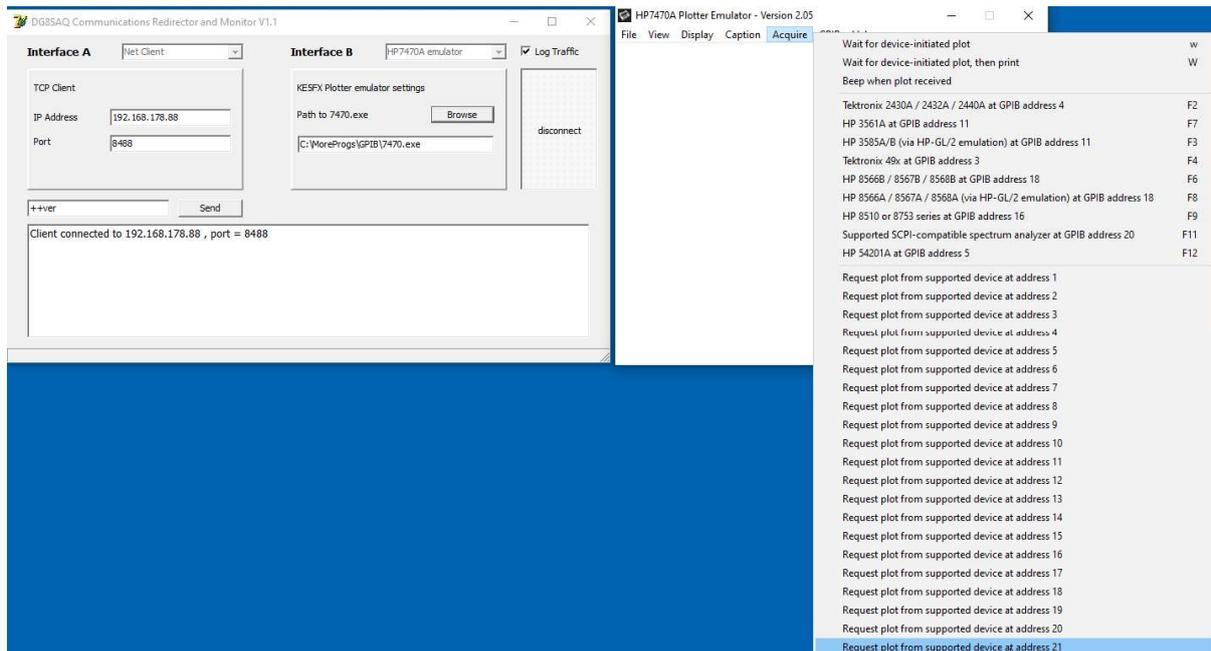


Zunächst muss der Dateipfad zur KE5FX Plottersoftware “7470.exe” angegeben werden. Sie können mittels des Browse-Knopfs danach suchen.

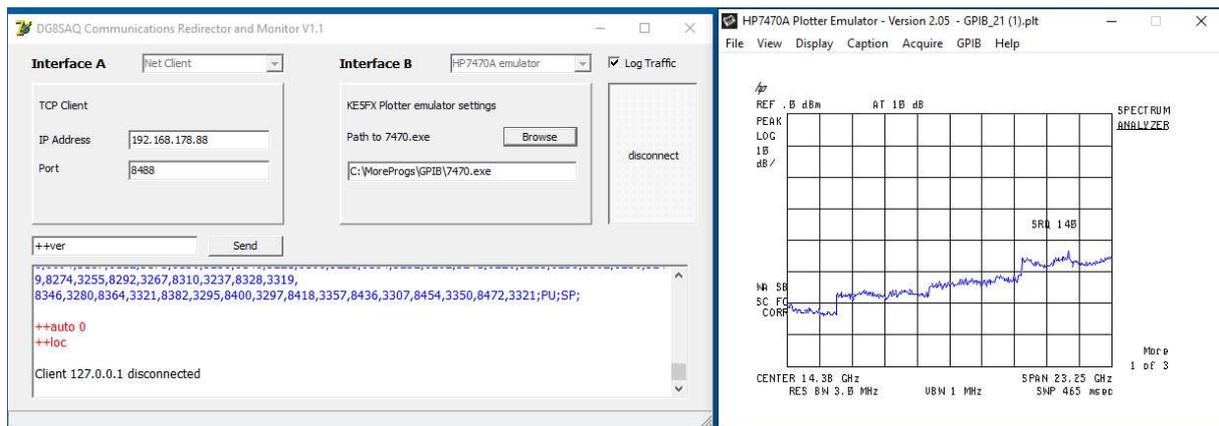


Beachten Sie, dass Interface A konfiguriert ist, um mit unserem ESP8266 zu kommunizieren.

Wenn Sie den Connect-Knopf drücken, wird die Plotter-Software automatisch gestartet. Jetzt können Sie wieder wie gewohnt einen Plot vom Instrument anfordern...

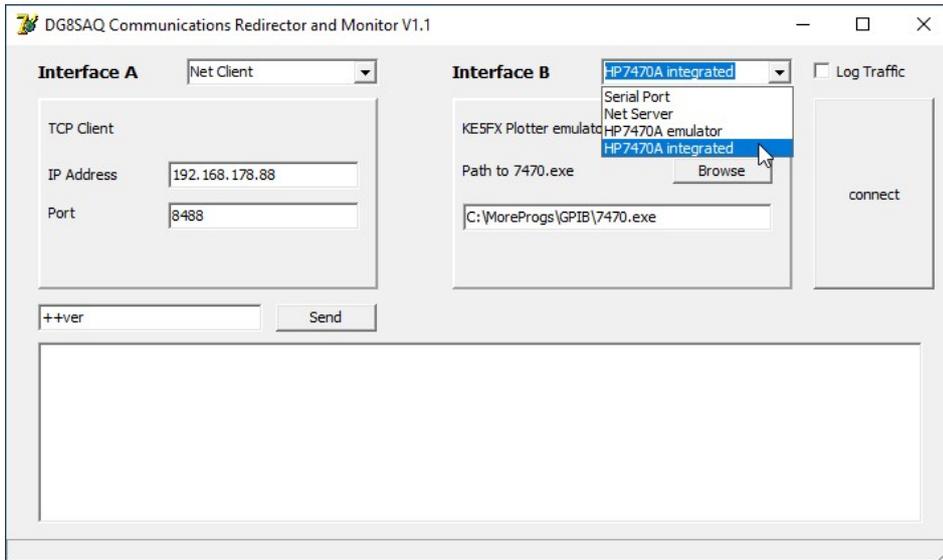


... und tatsächlich empfangen wir das Schirmbild, diesmal drahtlos über W-LAN:

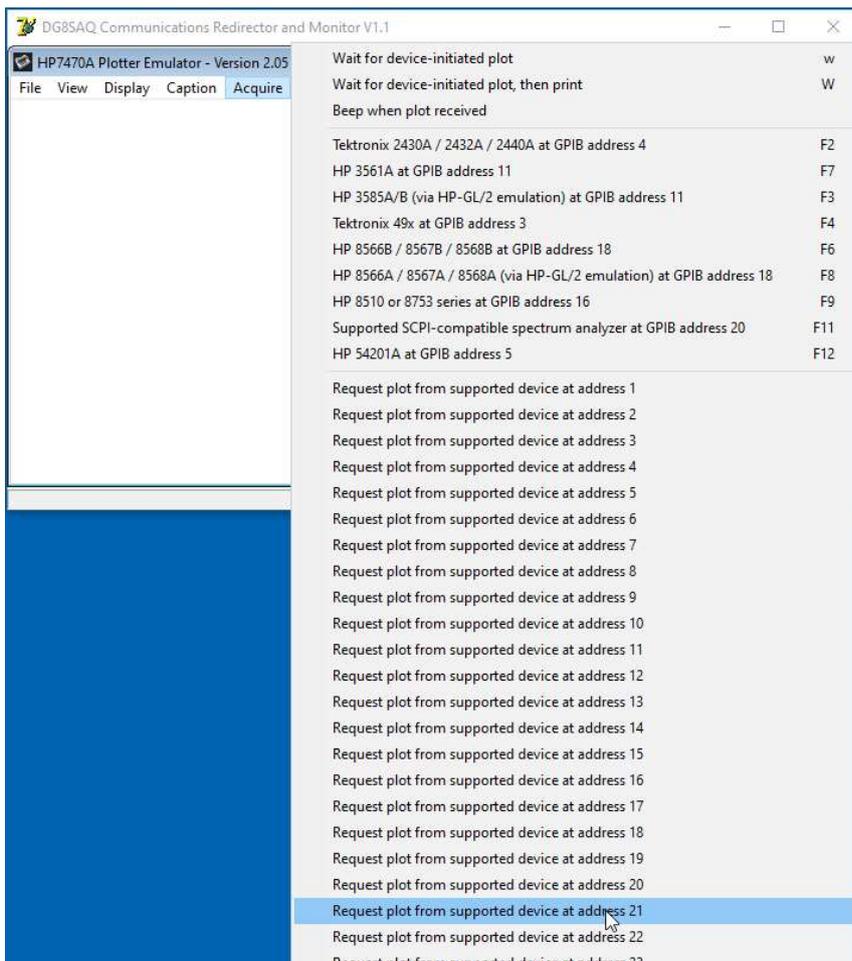


Beachten Sie, dass Sie die Kommunikation zwischen der Plottersoftware und dem GPIB-Adapter durch Aktivierung der "Log Traffic"-Funktion anschauen können. Diese wird im unteren Teil der Redirector-Software angezeigt, blau der Verkehr vom PC zum Adapter, rot der Verkehr vom Adapter zum PC. Oben sieht man blau noch einige HPGL-Kommandos, die der Spektrumanalyzer schickt.

Wenn wir nicht so viele Programmfenster offen haben möchten, dann können wir über “HP7470A integrated” die Plottersoftware im Redirector-Fenster öffnen.



Dies passiert, sobald Sie den “Connect”-Knopf drücken. Auch hier fordern Sie wieder wie gewohnt einen Plot an:



Dieser erscheint dann im Redirector-Fenster:



Beachten Sie, dass durch die Umskalierung der Grafik ins Redirector-Fenster die Qualität der Auflösung am Bildschirm zwar etwas leidet, wenn Sie die Grafikdatei aber abspeichern, besitzt diese die gewohnt hohe Qualität.

Die Verbindung von der Plotter-Software zum GPIB-Adapter wird durch Schließen z.B. mittels dem x-Knopf der Plotter-Software wieder beendet. Natürlich können Sie auch einfach den Redirector selbst beenden.

Ende