

JTAG-Flash für Opticum 9600HD-S

0. Vorwort

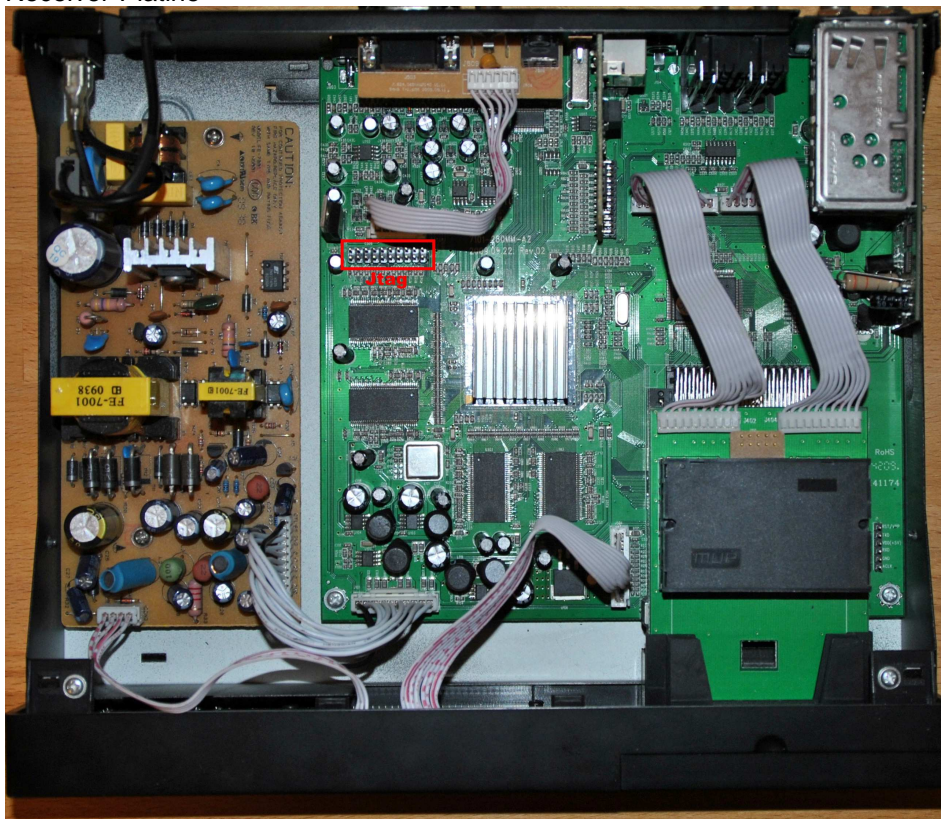
Diese Jtag-Anleitung wurde in Zusammenarbeit von „Rimini“ & „mir (Daniel)“ erstellt. Wobei die Hauptarbeit hierbei Rimini geleistet hat & ich dies nur auf diesen Receiver zurecht geschnitten habe.

JTAG bei den STi71xx Chips mit ST40 Cores geht z.Z. nur mit JTAG-Adapttern auf der Basis von FTDI FT2232H/FT4232H USB-Chips. Ganz fertige JTAG-Adapter gibt es eigentlich gar nicht dafür, aber der preiswerteste und geeignete Adapter ist der „ELV Highspeed Mini USB Modul UM-FT2232H, Komplettbausatz“ . Bis Ende 2010 gab es keine preiswerte und einfache Möglichkeit das JTAG-Interface dieser Receiver-Chips zu verwenden. Dazu musste man von STMicroelectronics ein „ST Micro Connect“ (kurz STMC) besitzen, was hauptsächlich den professionellen Entwicklern von Firmware für diese Receiver-Chips schon wegen des hohen Preises von mehreren Hundert US-Dollar vorbehalten war. Ende 2010 hatte dann STMicroelectronics mit einem Update der frei verfügbaren STLinux-Distribution (siehe STLinux.com) eine Beschreibung für ein „ST Micro Connect Lite“ (kurz STMCLT) mit der nötigen Softwareunterstützung für STLinux-Entwickler veröffentlicht. Die darin enthaltene Abbildung zeigt eindeutig, dass der Hauptbestandteil ein FTDI FT4232H Chip ist. Recht schnell wurde dann im „Avi+ Repair Tipps Forum“ (hauptsächlich von YLG80) die unveröffentlichte Schaltung und Funktion des „ST Micro Connect Lite“ mit Hilfe eines FTDI FT4232H Mini-Moduls ermittelt. Mit einem solchen „ST Micro Connect Lite“ Clone ist es seitdem recht preiswert, Receiver mit STMicroelectronics Chips (nicht nur die ST40 Cores, auch die älteren ST20 Cores) über JTAG zu flashen. Außerdem eignet sich ein solcher „ST Micro Connect Lite“ Clone auch für Hobby-Programmierer als Debug-Adapter zum Echtzeit-Debugging über das JTAG-Interface.

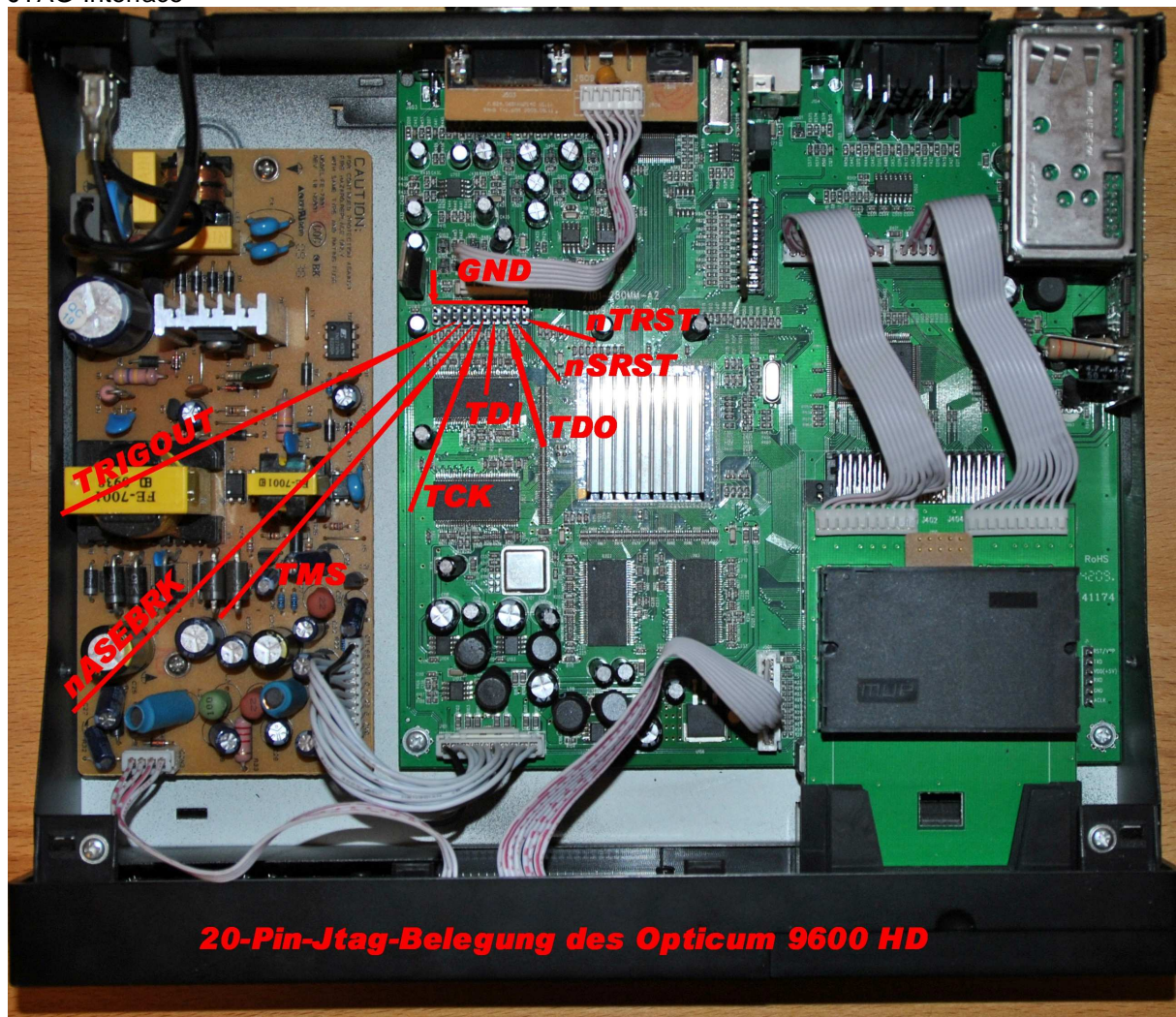
1. Hardware

1.1. Der Opticum 9600 HD S

Receiver-Platine



JTAG-Interface

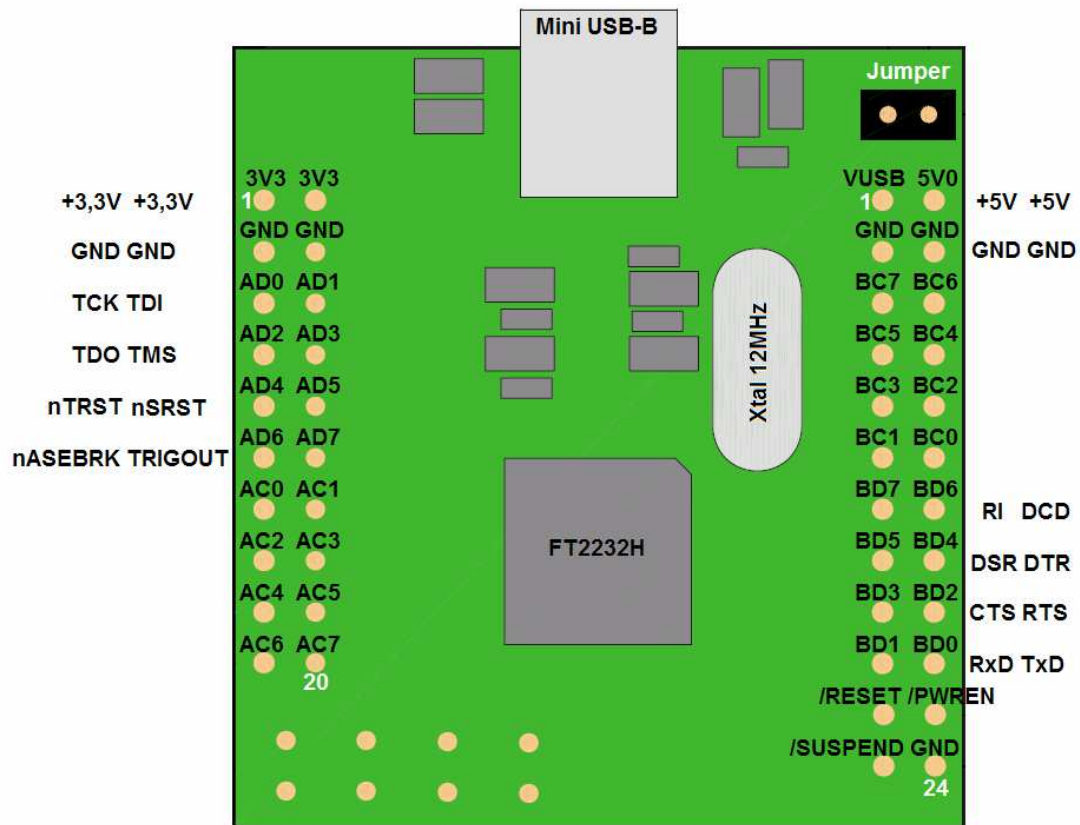
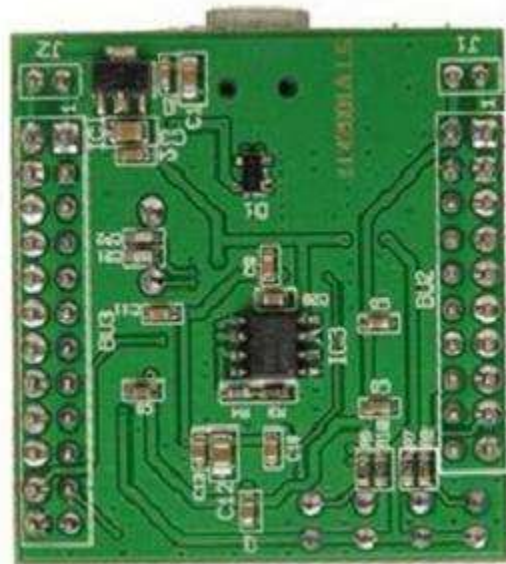
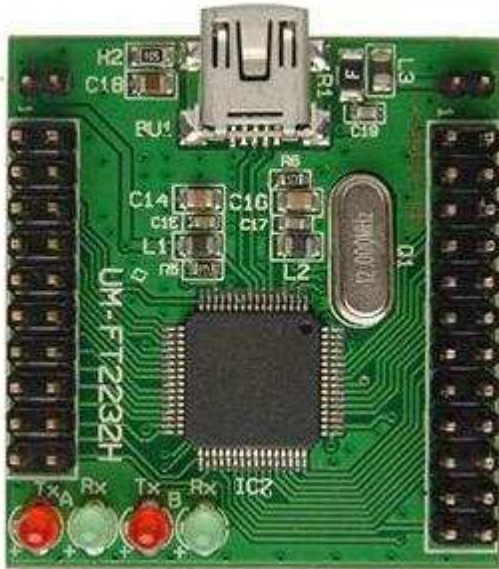


1.2. JTAG-Adapter

Wie bereits erwähnt, benötigt man einen „ST Micro Connect Lite“ Clone als JTAG-Adapter. Die Software von STMicroelectronics unterstützt nicht nur den FTDI FT4232H Chip, sondern auch den FT2232H Chip als „ST Micro Connect Lite“ Clone. Ein besonders preiswertes Modul mit FT2232H Chip gibt es als „ELV Highspeed Mini USB Modul UM-FT2232H, Komplettbausatz“. Bausatz ist da eigentlich schon übertrieben, es ist eine fast komplett bestückte Platine mit allen nötigen SMD-Bauteilen. Von Hand muss man selber nur noch wenigstens den mitgelieferten 12MHz Quarz und einen Jumper einlöten. Ob man dann noch die Steckerleisten für die Signale bestückt oder dort direkt Kabel einlötet, bleibt jedem selber überlassen. Die LED-Dioden muss und sollte man nicht unbedingt bestücken, sie benötigen nur zusätzlichen Strom vom USB-Port und haben keinen wirklichen Aussagewert. Wie auch beim „ST Micro Connect Lite“ Adapter kann man bei dem „ELV Highspeed Mini USB Modul UM-FT2232H“ neben einem JTAG-Anschluss noch gleichzeitig den zweiten Kanal des FT2232H Chips als seriellen Port verwenden. Die 5V tolerant Low-Voltage TTL Anschlüsse der FT4232H/FT2232H Chips kann man direkt mit einem einfachen TTL/ RS-232 Wandler (z.B. MAX232) verbinden und dann als USB/RS-232 Adapter verwenden. Einen dazu passenden Bausatz gibt es als „ELV TTL - nach RS 232-Umsetzer, Komplettbausatz“, der sogar den Vorteil hat, dass er eine 9-polige Buchse für die RS-232 Signale besitzt und man kein zusätzliches Nullmodem-Kabel benötigt. Es werden aber nicht alle Kontakte benötigt für den Jtag. Gebraucht werden, „**GND**“, „**nTRST**“, „**nSRST**“, „**TDO**“, „**TDI**“, „**TCK**“, „**TMS**“.



ELV Highspeed Mini USB Modul UM-FT2232H



2. Software

Ein „ST Micro Connect Lite“ Clone JTAG-Adapter ist sowohl mit dem STLinux-Paket (aktuelle Version 2.4) von STMicroelectronics unter Verwendung einer Fedora basierende Linux-Distribution, wie auch mit dem „ST40 Mirco Toolset“ (ab R5.1.0) in Verbindung mit dem „ST Micro Connection Package“ von STMicroelectronics unter WindowsXP (32Bit) benutzbar.

Neuere Linux-Distributionen haben für den „ST Micro Connect Lite“ JTAG_Adapter entsprechende Treiber im Linux-Kernel, aber auch da braucht man wenigstens für WindowsXP noch die Treiber für den FT4242H bzw. FT2232H Chip von FTDI (<http://www.ftdichip.com>) und die die MProg 3.5 Utility, um unter WindowsXP das „ELV Highspeed Mini USB Modul UM-FT2232H“ zum „ST Micro Connect Lite“ Clone JTAG-Adapter umprogrammieren zu können.

2.1 Das STMicroelectronics „ST40 Micro Toolset“

Mittlerweile hat STMicroelectronics auf der Web-Seite für STLinux auch die Zugangsdaten für ihren FTP-Server veröffentlicht, so dass man regulär auch an die nötige Software kommt:

`ftp://sts-tools:sts-tools@ftp.st.com/products/`

(Falls man nochmals nach den Zugangsdaten gefragt wird, sind diese für Login-Name und Passwort jeweils „sts-tools“.)

Im Verzeichnis *st40tools* findet man dann alle bisher veröffentlichten Versionen des „ST40 Micro Toolset“. Ein „ST Micro Connect Lite“ oder Clone JTAG-Adapter wird ab der Version R5.1.0 unterstützt und im Unterverzeichnis *ST40_GNU_R5.1.0* befindet sich die zugehörige Windows-Installationsdatei ***stm-st40.510-5.1.0-MSWin32-x86.exe***. Startet man diese Datei, erfolgt eine Windows-typische Installation. Zuerst wird die Installation mit Hilfe von InstallAnywhere vorbereitet und dann in aufeinander folgenden Schritten, die man einzeln bestätigen muss, durchgeführt. Die Vorgaben in den einzelnen Schritten sollte man so belassen, wie sie sind. An zwei Stellen muss man aber von Hand eingreifen:

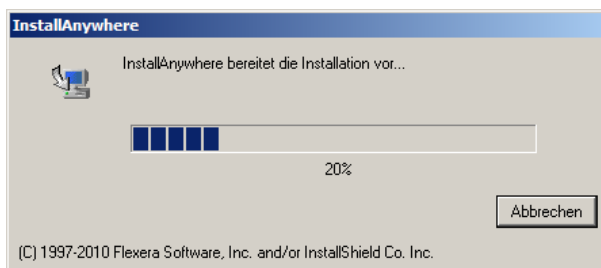
1. Man muss die „ST License“ akzeptieren, damit man bei der Installation fortsetzen kann.
2. Es macht für die Hobby-Nutzung keinen Sinn, die Installation zu registrieren und muss/sollte deshalb die vorgegebene Auswahl zur Registrierung deaktivieren.

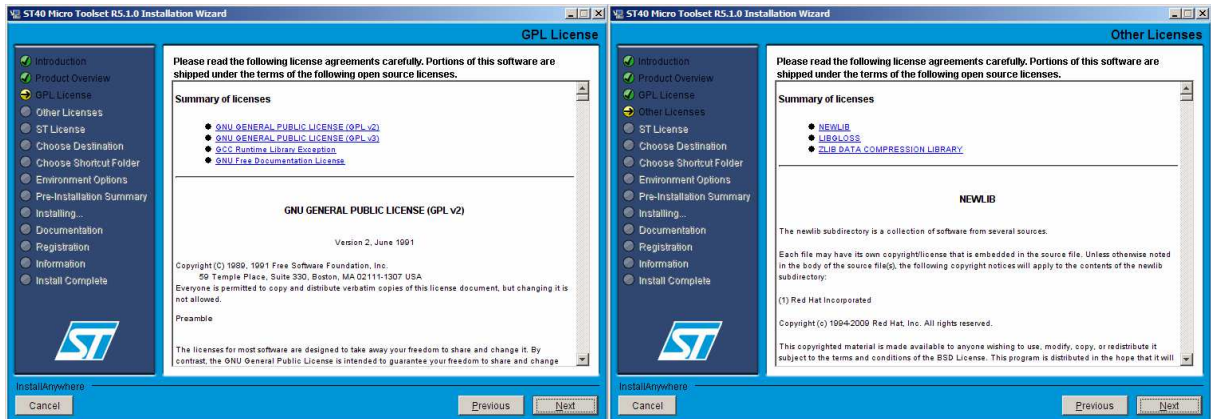
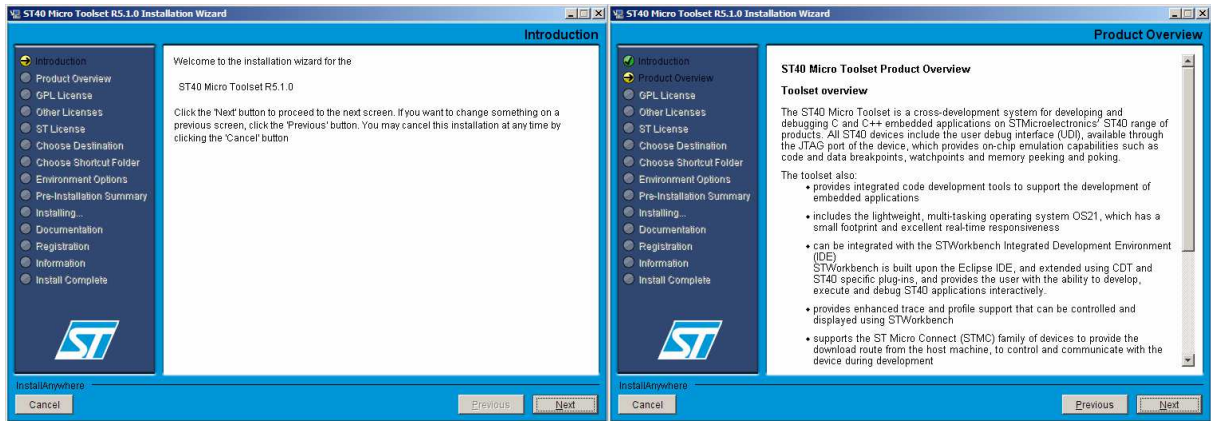
Ob man sich den Index für die installierten Dokumentationen vor der Registrierung anzeigen lässt oder nicht, kann man selber entscheiden. Darauf hat man nach abgeschlossener Installation auch später jederzeit über das „Programm-Menü“ von Windows einen Zugriff.

Zum Schluss der Installation wird man noch darauf hingewiesen, dass man auch noch das „ST Micro Connection Package“ installieren muss, wenn man die „ST Target Packs“ mit dem „ST40 Micro Toolset“ verwenden will. Das folgt dann im nächsten Abschnitt.

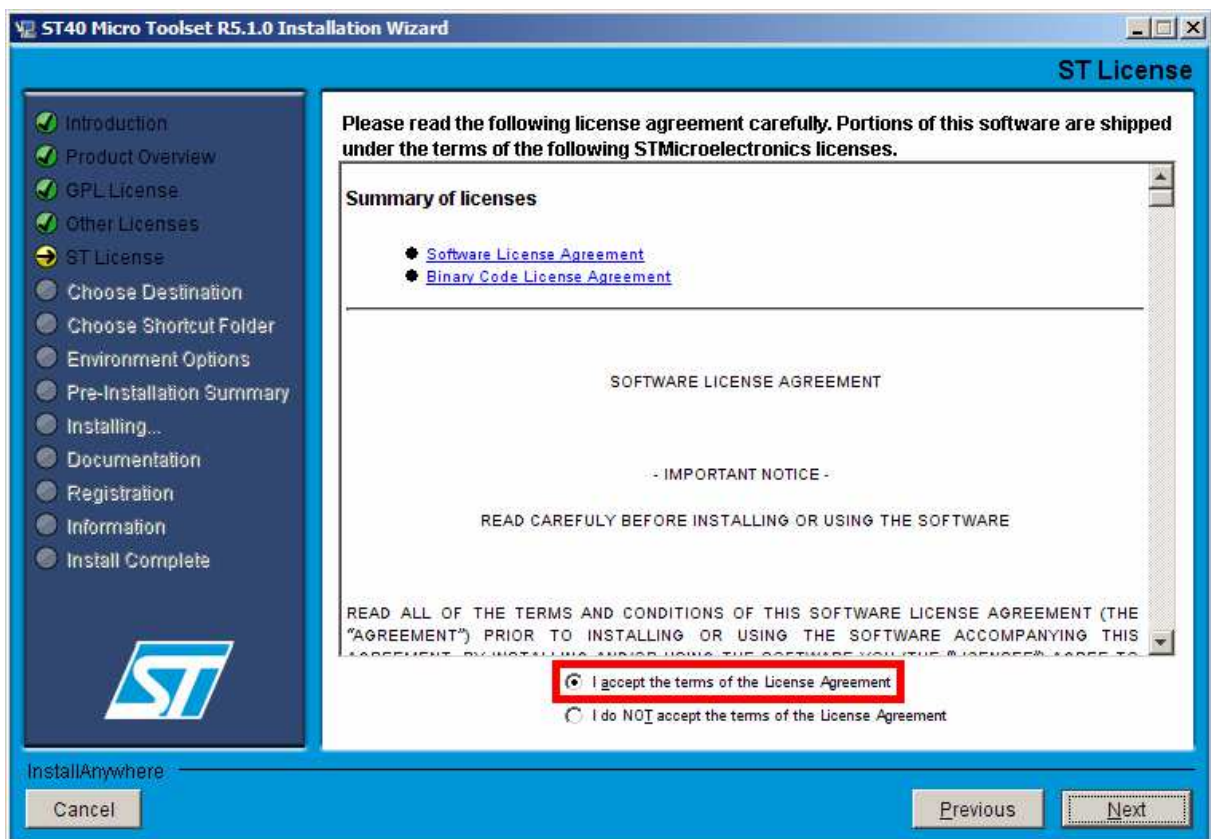
Der Abschluss der Installation dauert je nach Leistungsfähigkeit des verwendeten PCs ein paar Sekunden. Also warten und nicht ungeduldig sein.

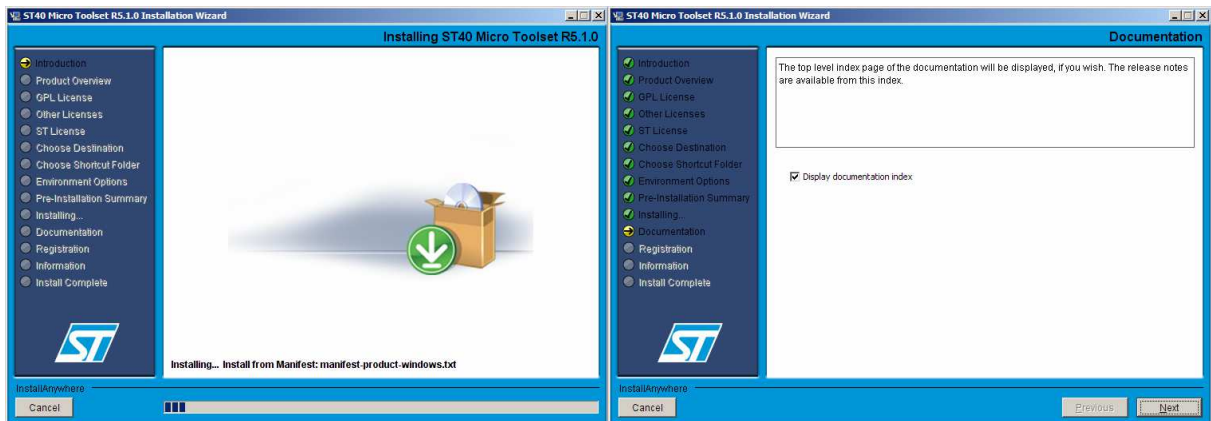
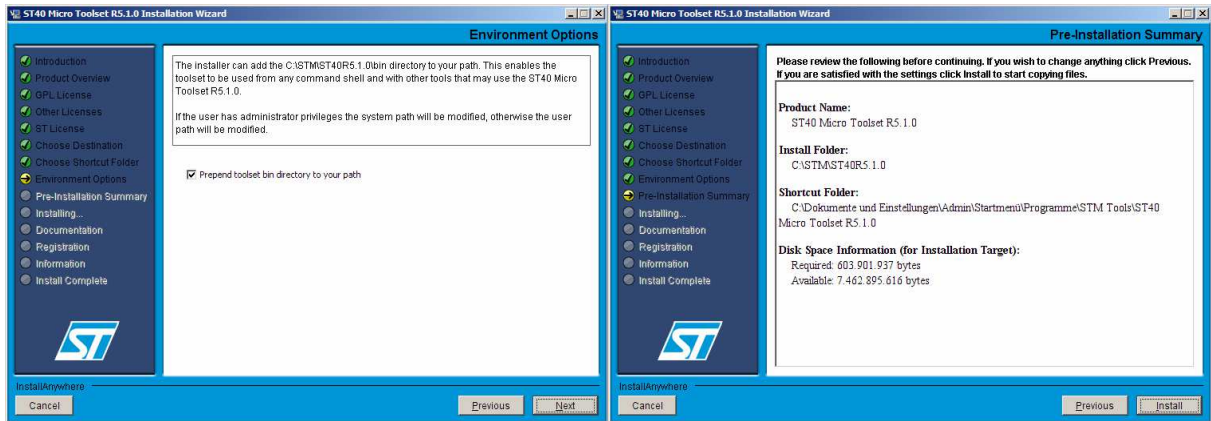
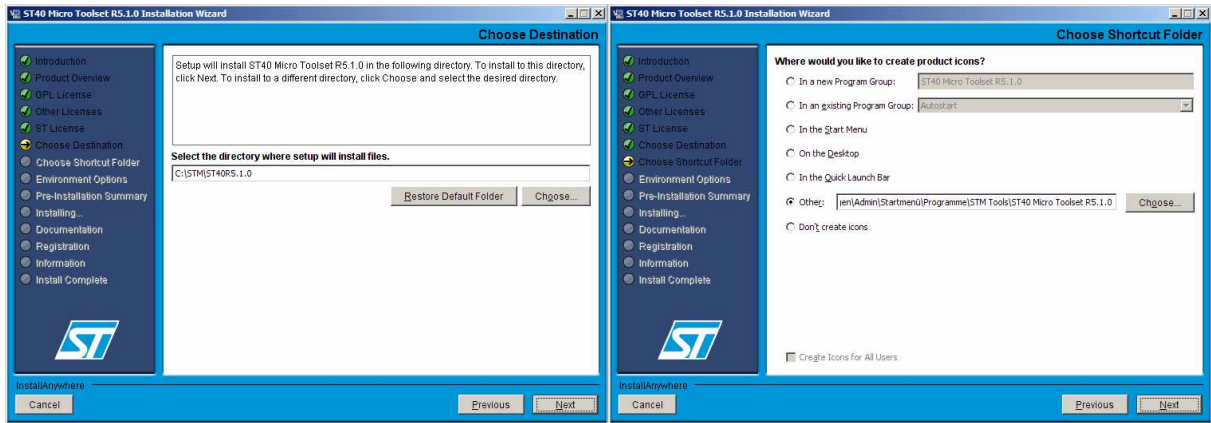
Los geht es:



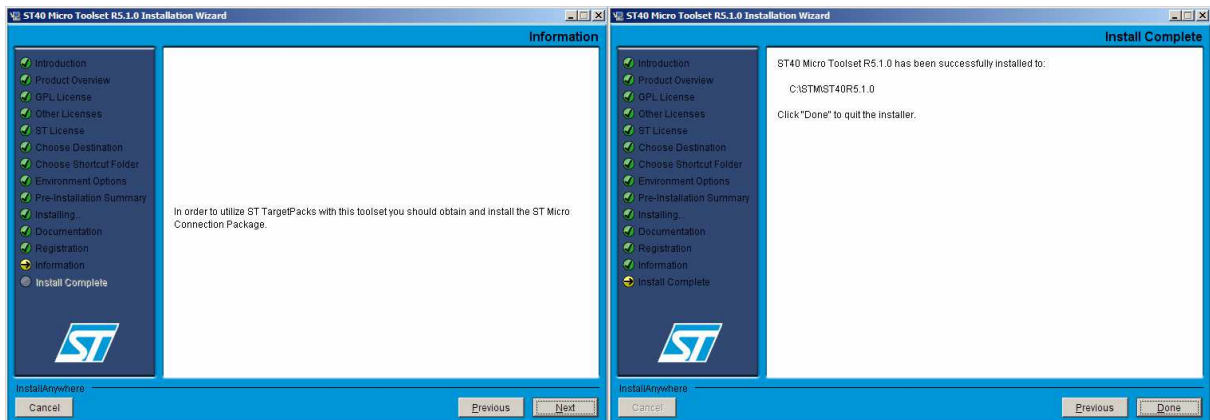
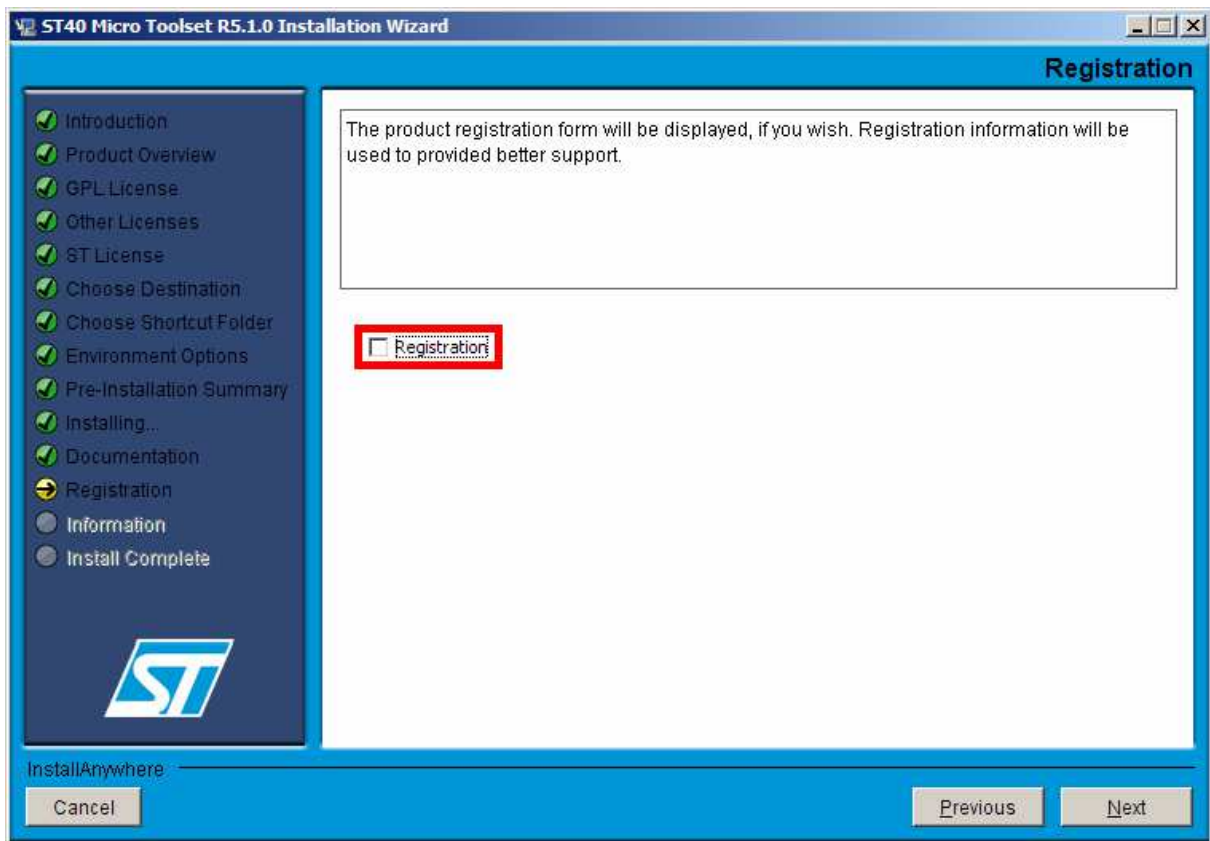


ST License muss man akzeptieren, um fortsetzen zu können:





Registrierung ist für Hobby-Nutzer nicht sinnvoll nutzbar und muss/sollte daher deaktiviert werden:

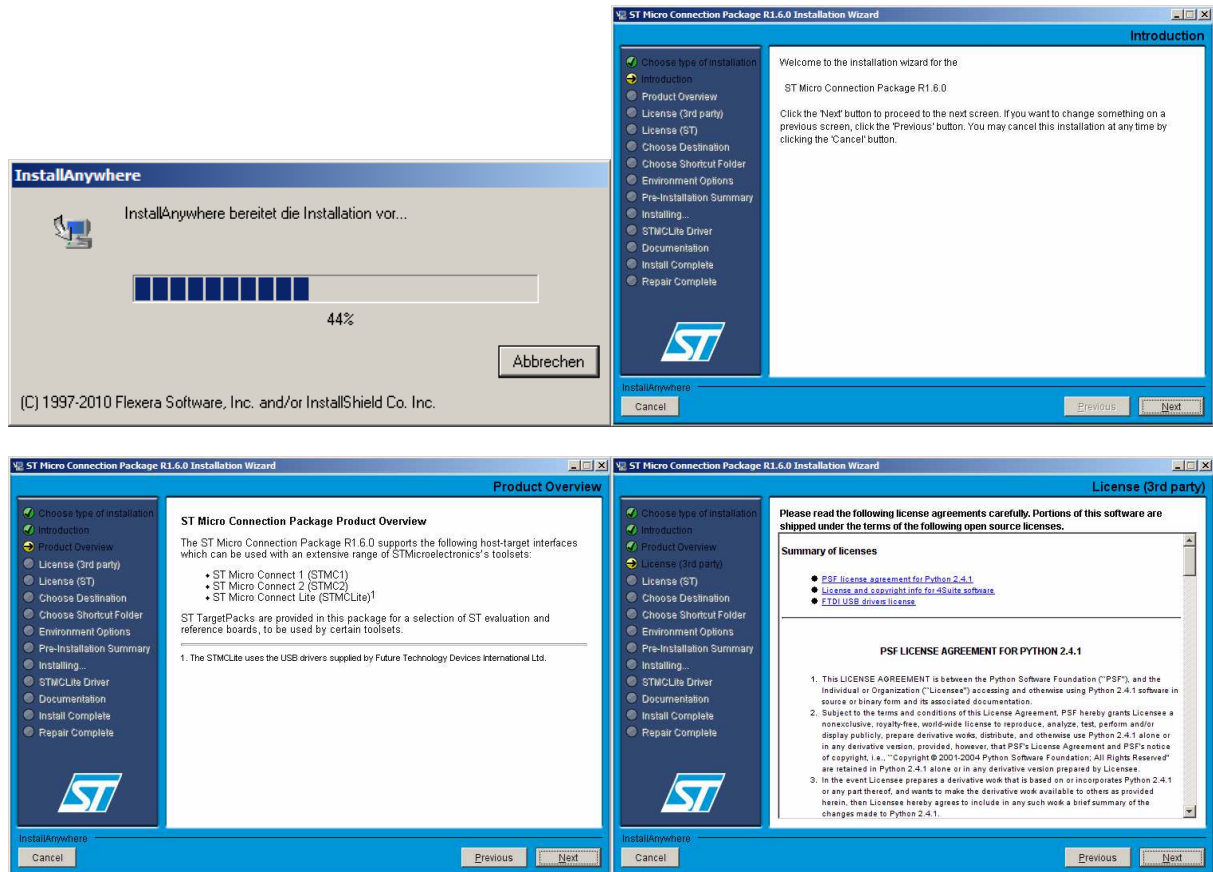


Nach etwas Wartezeit ist die Installation dann beendet.

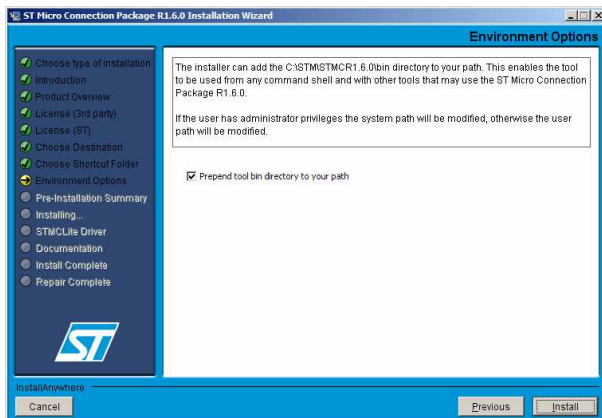
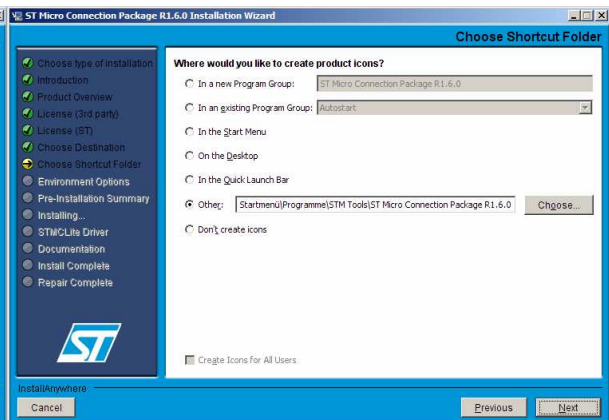
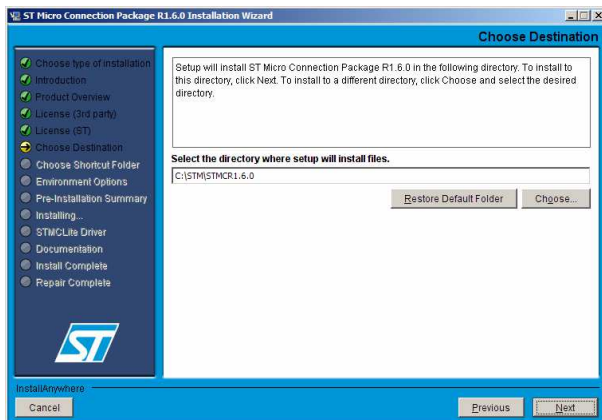
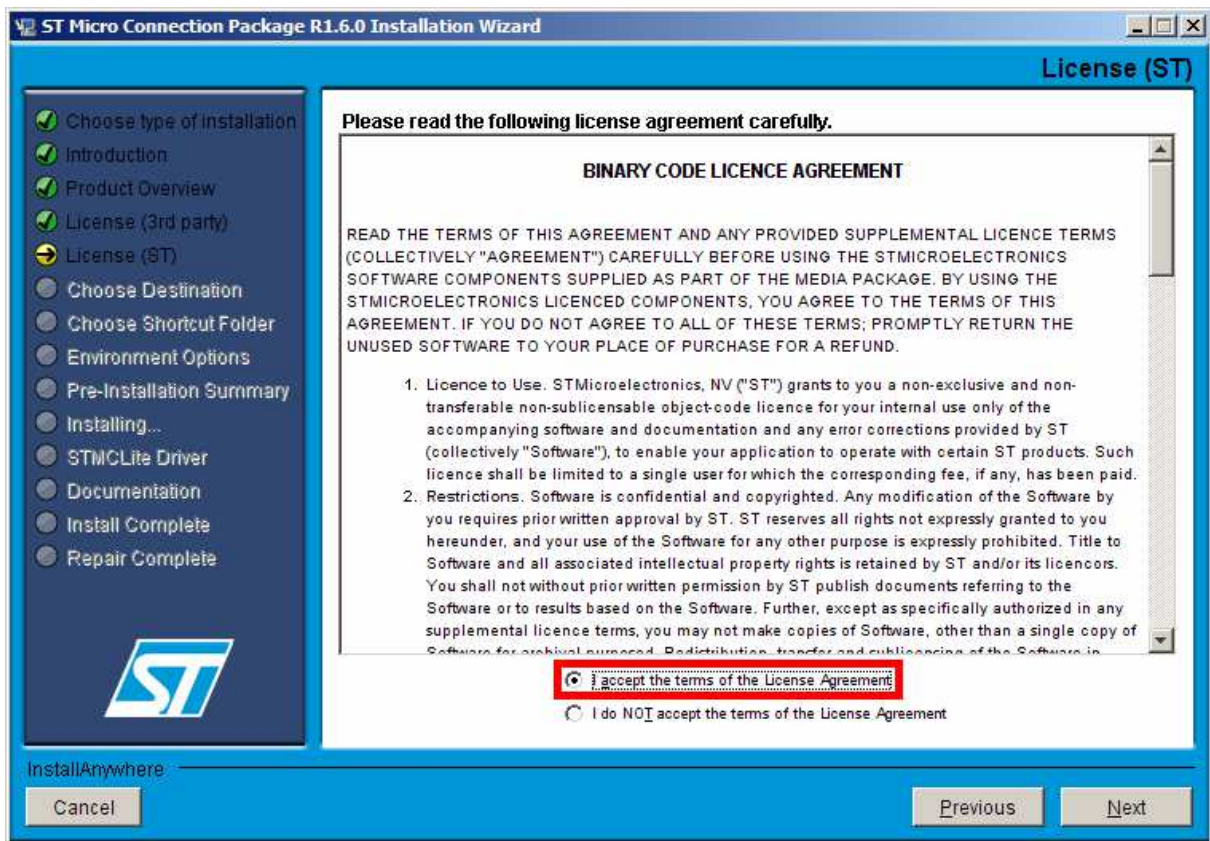


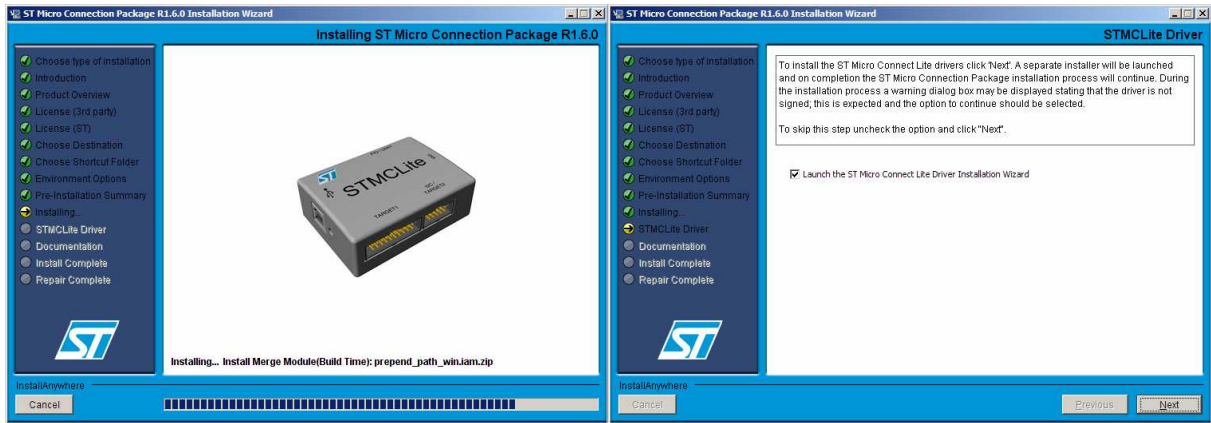
2.2 Das STMicroelectronics "ST Micro Connection Package"

Auf dem FTP-Server von STMicroelectronics findet man im Verzeichnis **stmc2** auch alle bisher veröffentlichten Versionen des „ST Micro Connection Package“ für die „ST Micro Connect“ JTAG-Adapter. Ein „ST Micro Connect Lite“ oder Clone JTAG-Adapter wird mit der Version R1.6.0 unterstützt. Im Unterverzeichnis **R1.6.0** befindet sich dann die notwendige Windows-Installationsdatei **stm-stmc.160-1.6.0-MSWin32-x86.exe** zum Download. Die Installation erfolgt wieder Windows-typisch in ähnlichen Schritten wie im vorherigen Abschnitt auch:

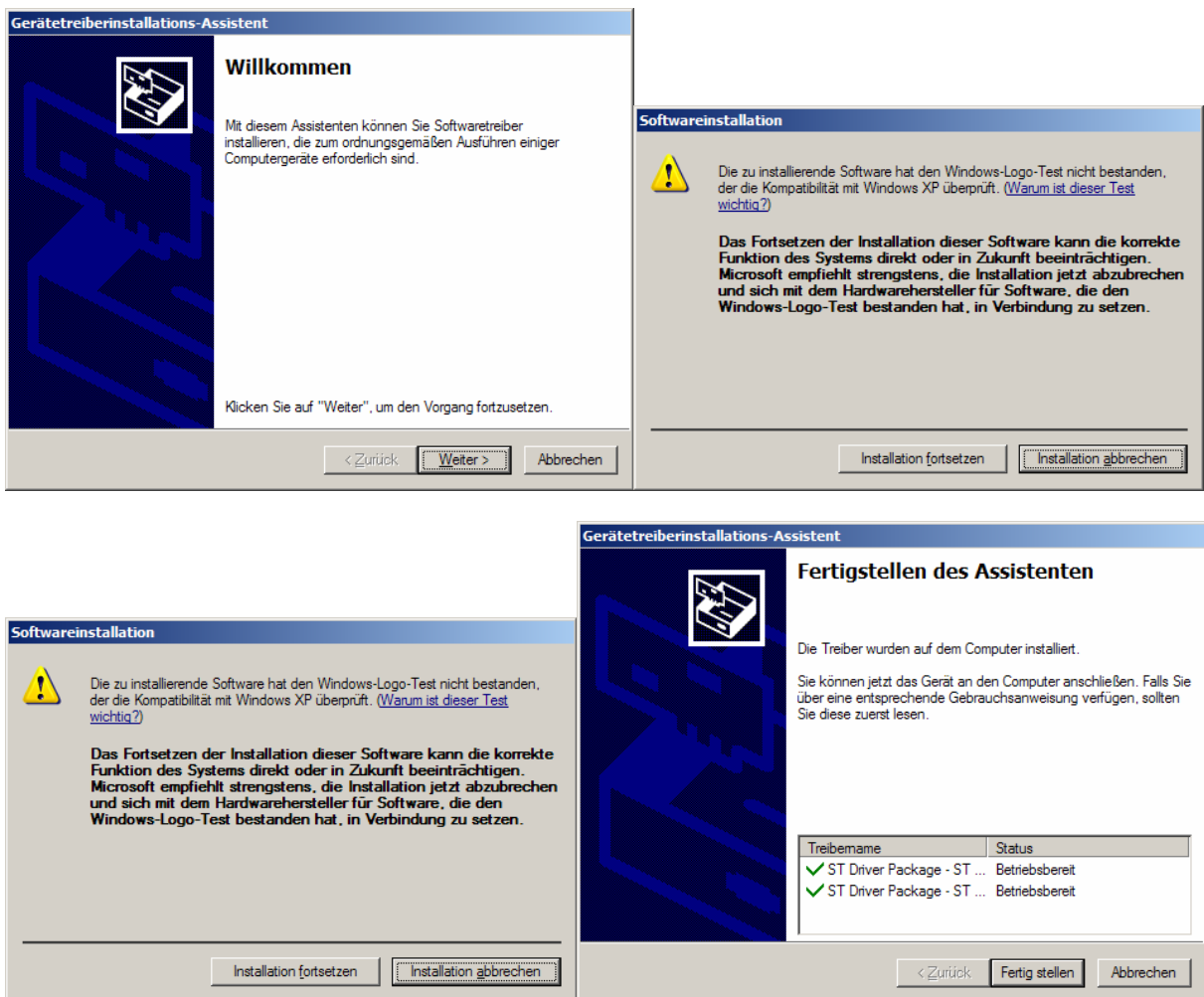


Die License muss man wieder akzeptieren, um fortsetzen zu können:



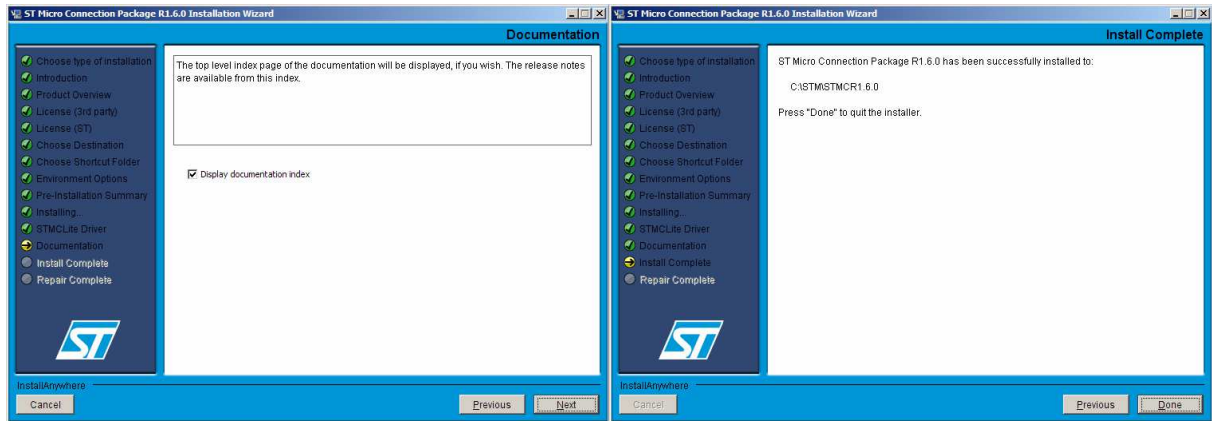


Wenn man im vorherigen Schritt nicht gerade die Installation der Windows-Treiber für den „ST Micro Connect Lite“ Adapter deaktiviert hat, wird der Gerätetreiberinstallations-Assistent von Windows automatisch gestartet und man muss der Installation der unsignierten Treiber zweimal zustimmen:



Danach kann man mit der Installation des “ST Micro Connection Package” fortsetzen und diese abschließen.

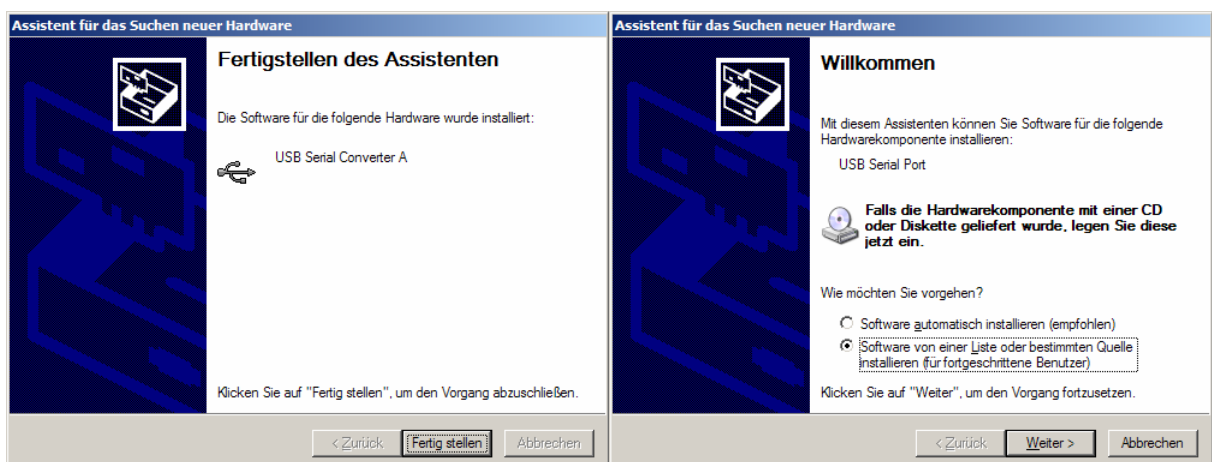
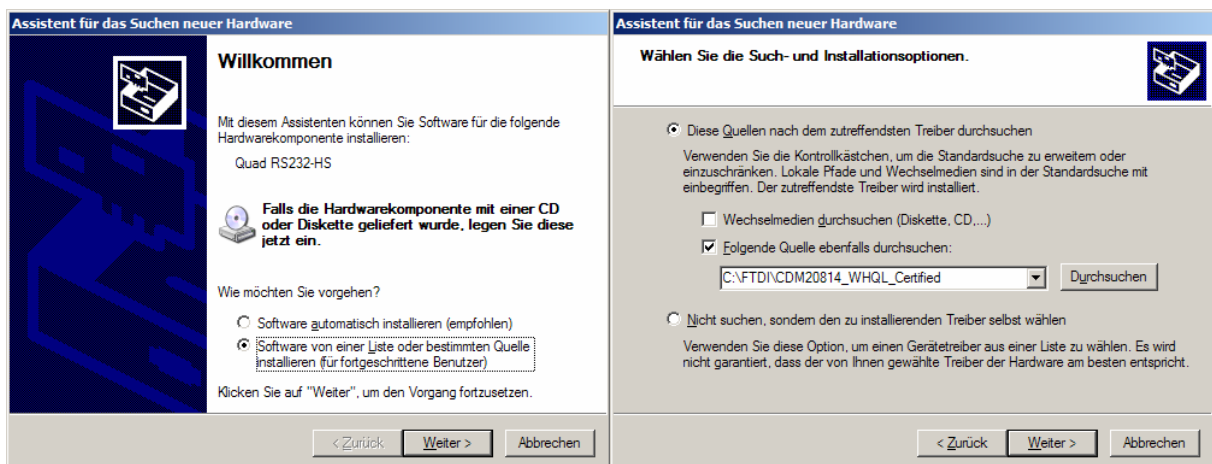


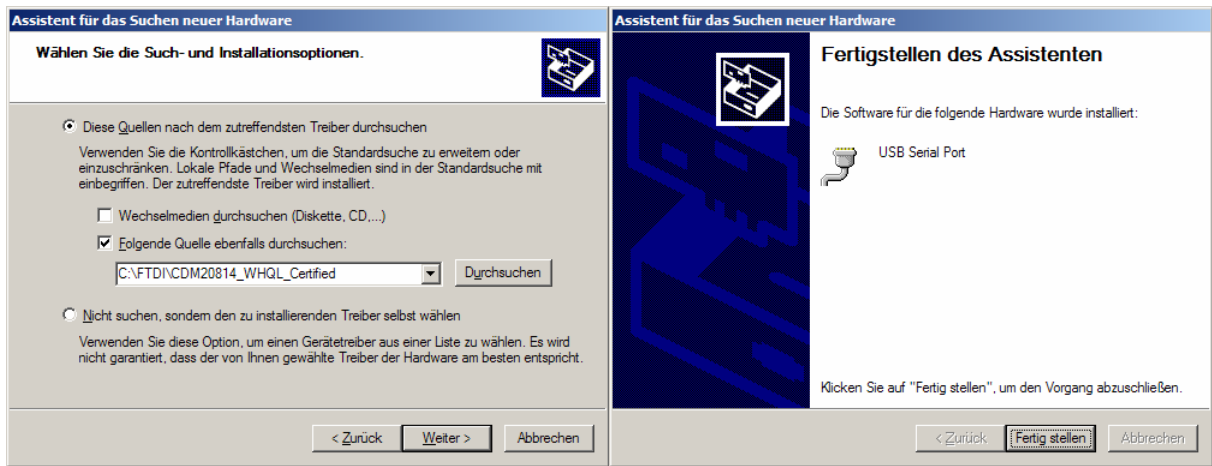


2.3 Die FDTI „MProg 3.5“ Utility

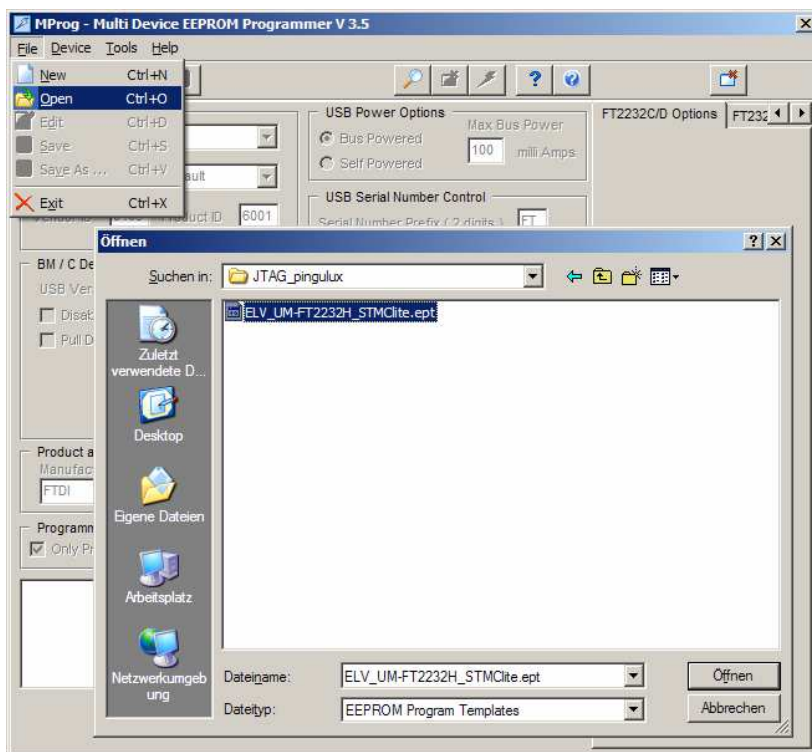
Um das „ELV Highspeed Mini USB Modul UM-FT2232H“ als „ST Micro Connect Lite“ Clone verwenden zu können, muss der EEPROM des Moduls vorher entsprechend programmiert werden, damit die USB-Kennungen (VID, PID und Seriennummer usw.) mit dem übereinstimmen, was die Software von STMicroelectronics erwartet. Zum Programmieren von Adaptionen mit FTDI-Chips gibt es von FTDI selbst entsprechende Tools und für den FT2232H Chip bietet sich da „MProg 3.5“ an.

Wenn das „ELV Highspeed Mini USB Modul UM-FT2232H“ zum ersten Mal an den USB-Port eines PCs angeschlossen wird, müssen erstmal die zugehörigen Treiber von FTDI (kann man bei FTDI immer in der aktuellsten Version downloaden und in ein beliebiges Verzeichnis entpacken) mit Hilfe des dann automatisch startenden Hardware-Installations-Assistens installiert werden:

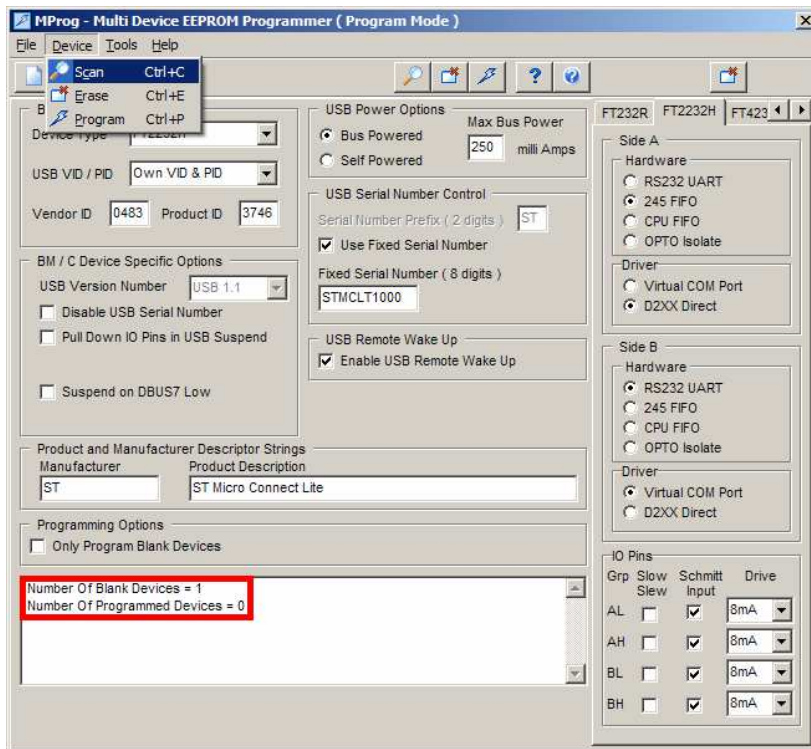




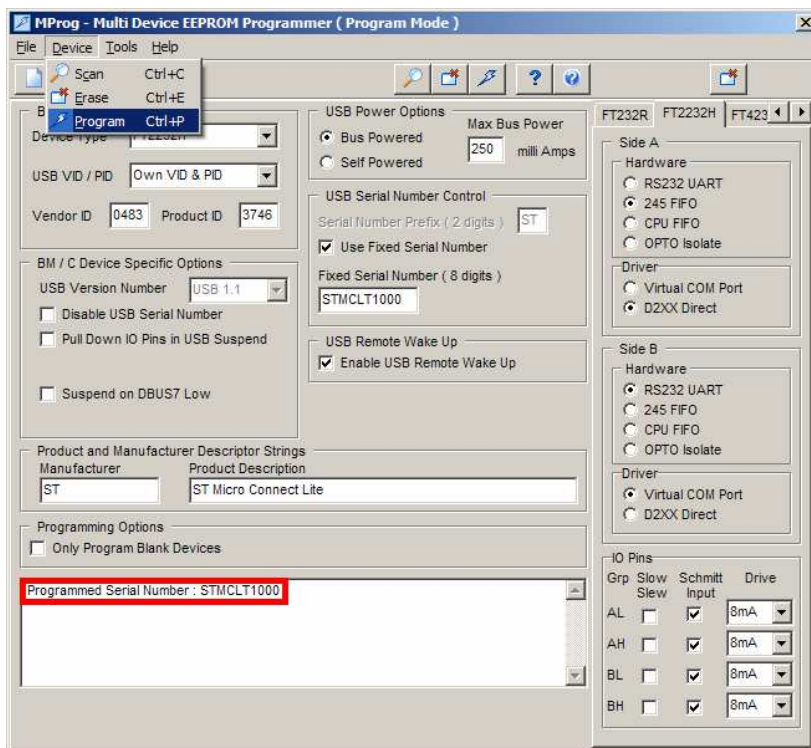
Die „MProg 3.5“ Utility kann man bei FTDI downloaden, in ein beliebiges Verzeichnis entpacken und dort dann direkt starten. Über das Menü „File“ kann man dann mit „Open“ die dieser Beschreibung beigefügte Template-Datei **ELV_UM-FT2232H_STMClite.ept** laden:



Über das Menü „Device“ kann man mit „Scan“ nach einem angeschlossenen FTDI basierenden Adapter suchen lassen und diesen dann auch mit den Template-Daten über „Program“ entsprechend programmieren. Auf keinen Fall sollte man die Template-Daten verändern, wenn man nicht weiß, was man damit bewirkt. Auf jeden Fall sollte man darauf achten, dass nur ein einziges FTDI-Device (unprogrammiert oder auch schon vorher eventuell anders programmiert) zum Programmieren gefunden wurde, also wirklich nur das zu programmierende FTDI-Device am PC angeschlossen ist:



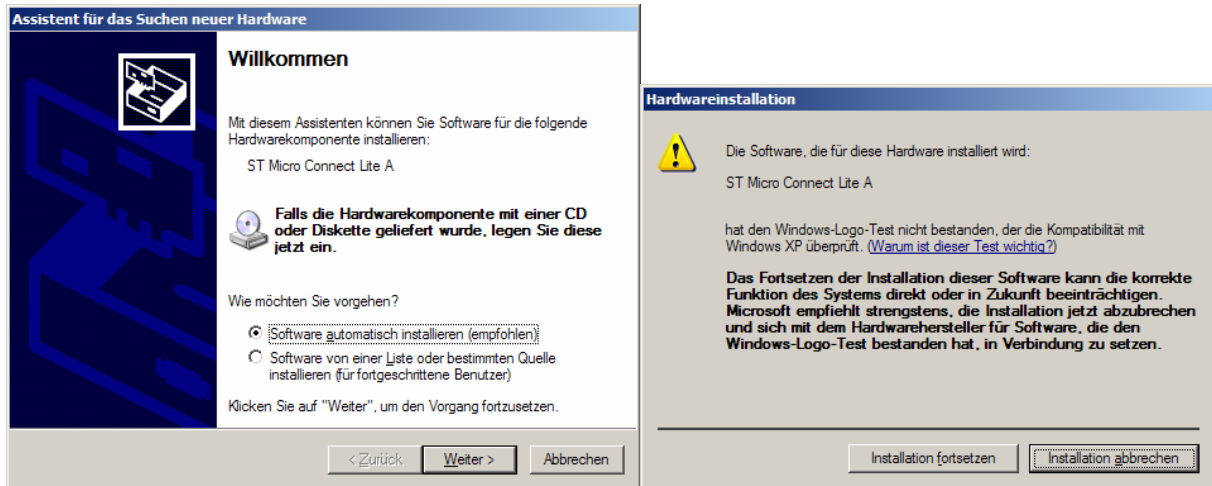
Die erfolgreiche Programmierung wird zum Schluss angezeigt:



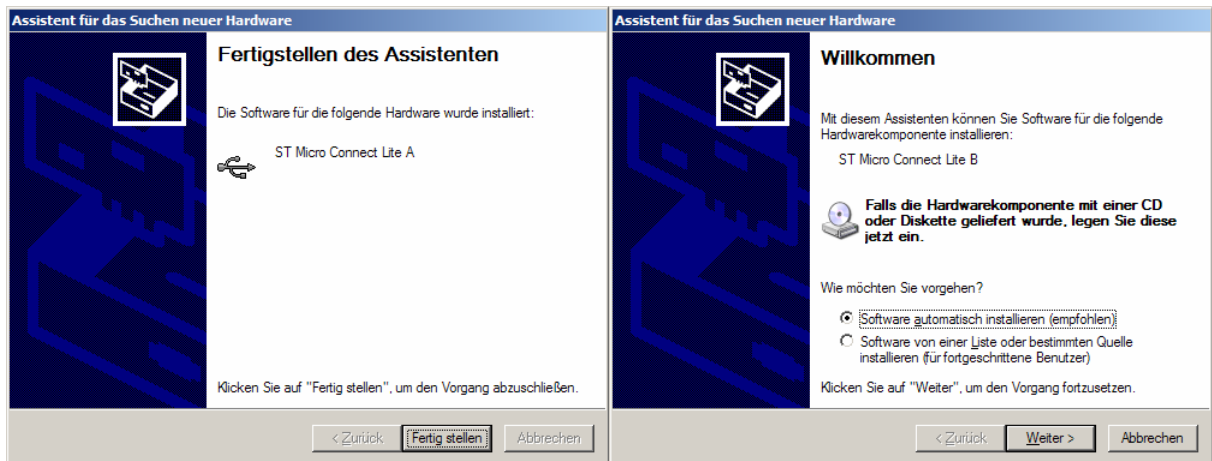
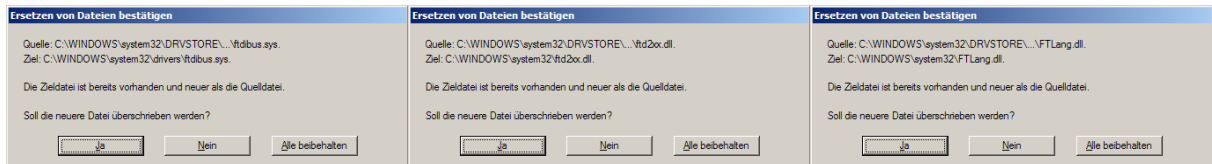
3. JTAG-Flashen des Bootloaders

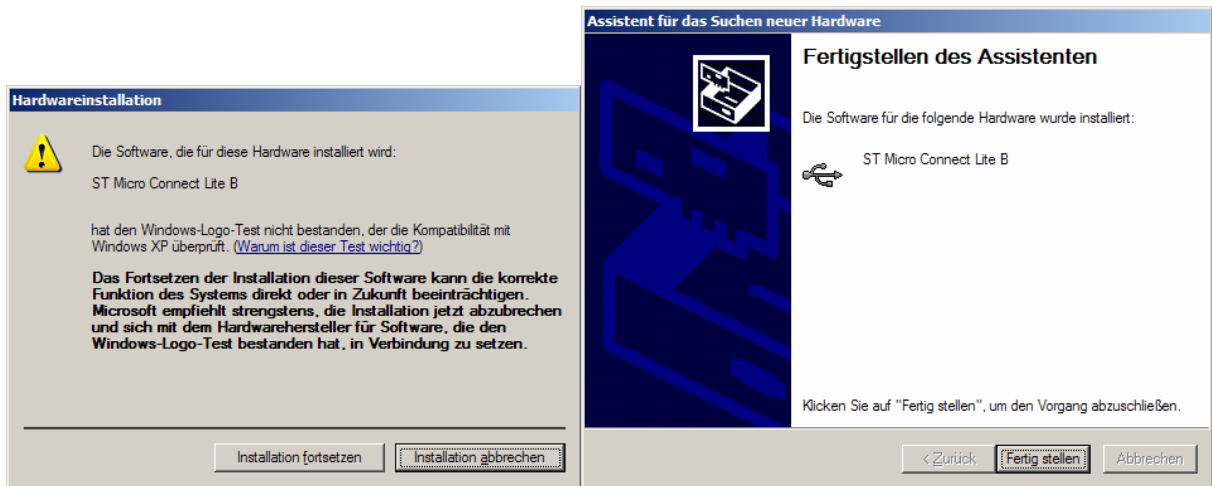
3.1 Installation der Treiber für den "ST Micro Connect Lite" Clone JTAG-Adapter

Nach dem das „ELV Highspeed Mini USB Modul UM-FT232H“ mit Hilfe von MPog 3.5 als „ST Micro Connect Lite“ Clone umprogrammiert wurde und erneut mit dem USB-Port eines PCs verbunden wird, startet wieder automatisch der Hardware-Installations-Assistent zur Installation der Treiber für den "ST Micro Connect Lite" JTAG-Adapter (werden automatisch gefunden, da das "ST Micro Connection Package" einschließlich der Windows-Treiber für den "ST Micro Connect Lite" JTAG-Adapter vorher installiert wurden) und der Installation der nicht signierten Windows-Treiber muss man wieder zustimmen:

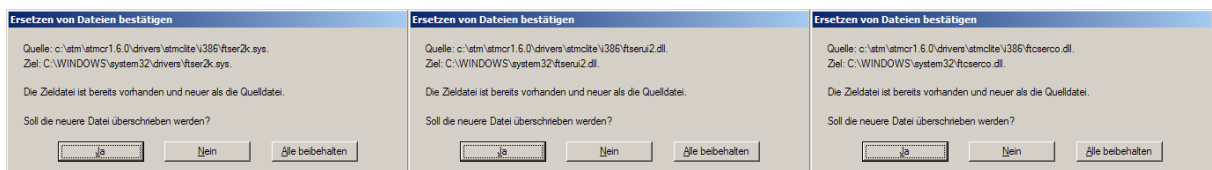
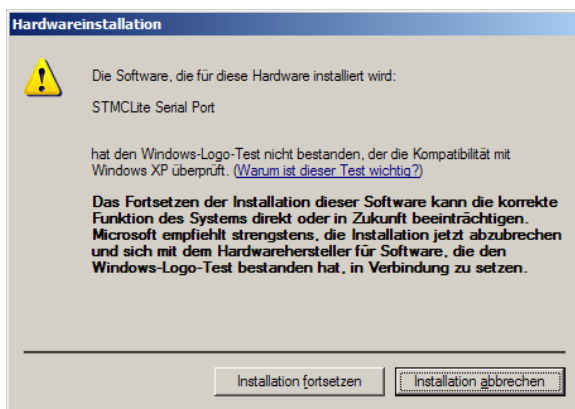
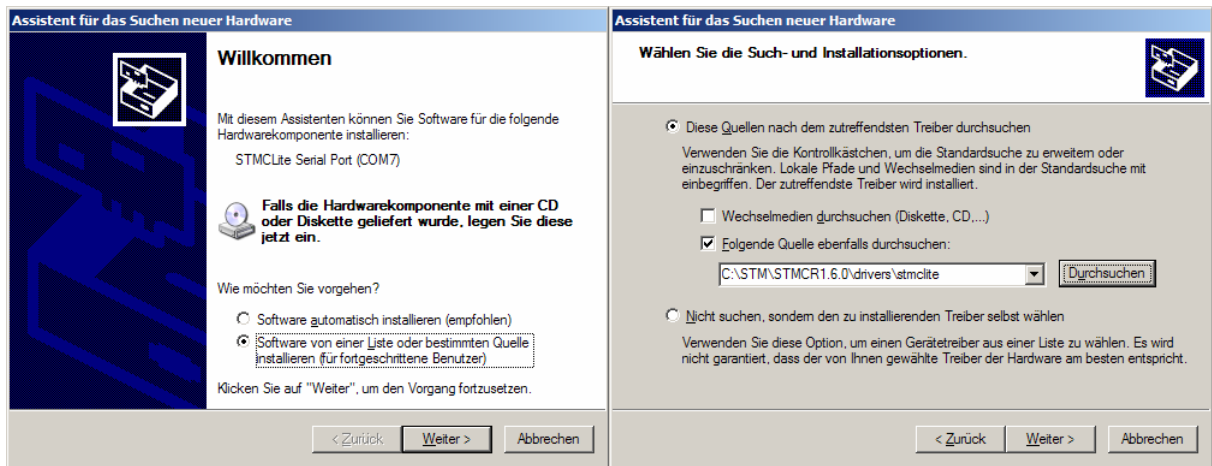


Da die vorher installierten FTDI-Treiber für das noch nicht umprogrammierten „ELV Highspeed Mini USB Modul UM-FT232H“ neueren Datums als die von STMicroelectronics sind, muss/sollte man die neueren FTDI-Dateien durch die älteren von STMicroelectronics ersetzen lassen:



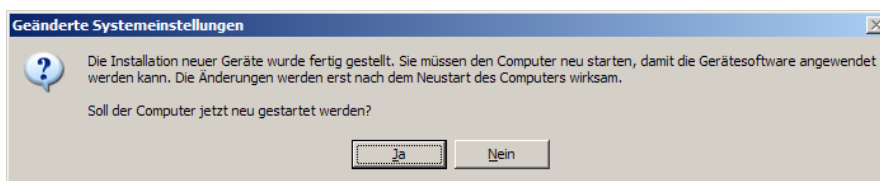


Den passenden Treiber für den virtuellen COM-Port des FT2232H Chips findet der Hardware-Installation-Assistent nicht automatisch und man muss die Treiber von STMicroelectronics, die unter **C:\STM\STMCR1.6.0\drivers\stmclite** installiert wurden, manuell vorgeben, die Installation der nicht signierten Treiber zulassen und neuere Dateien durch die älteren ersetzen lassen:

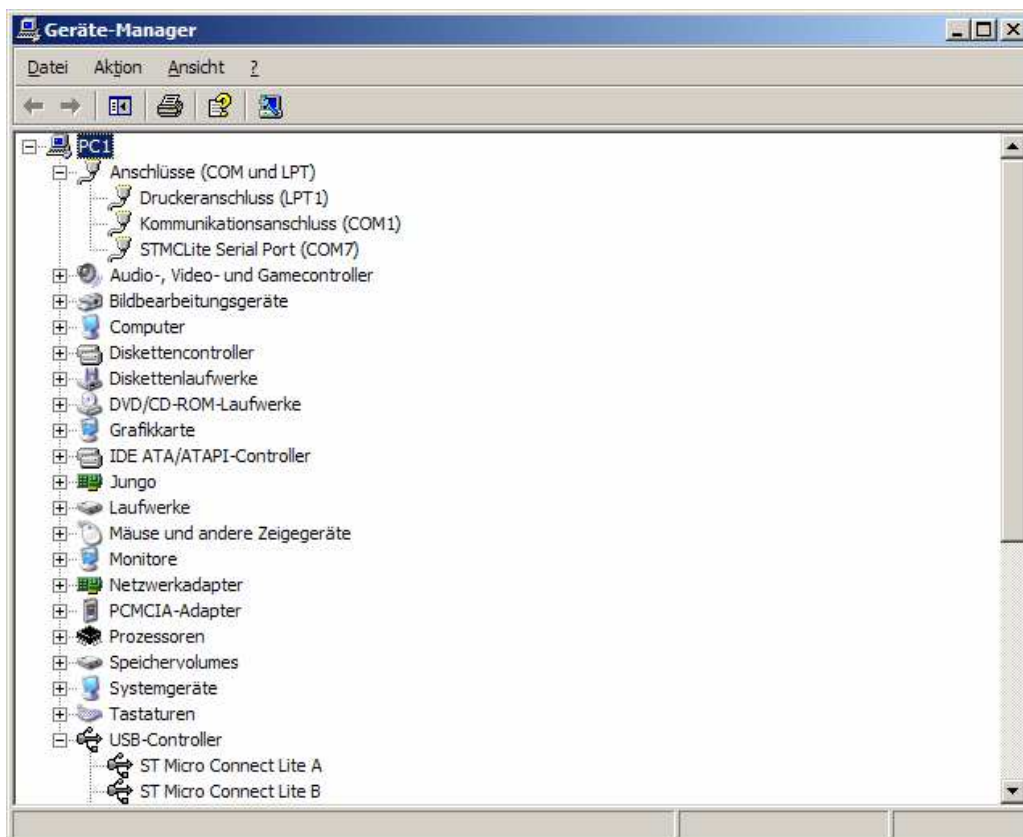




Eventuell ist ein Neustart von Windows notwendig:



Jetzt sollte das „ELV Highspeed Mini USB Modul UM-FT2232H“ als „ST Micro Connect Lite“ JTAG-Adapter im Geräte-Manager von Windows vorhanden sein und auch der zugehörige virtuelle COM-Port:

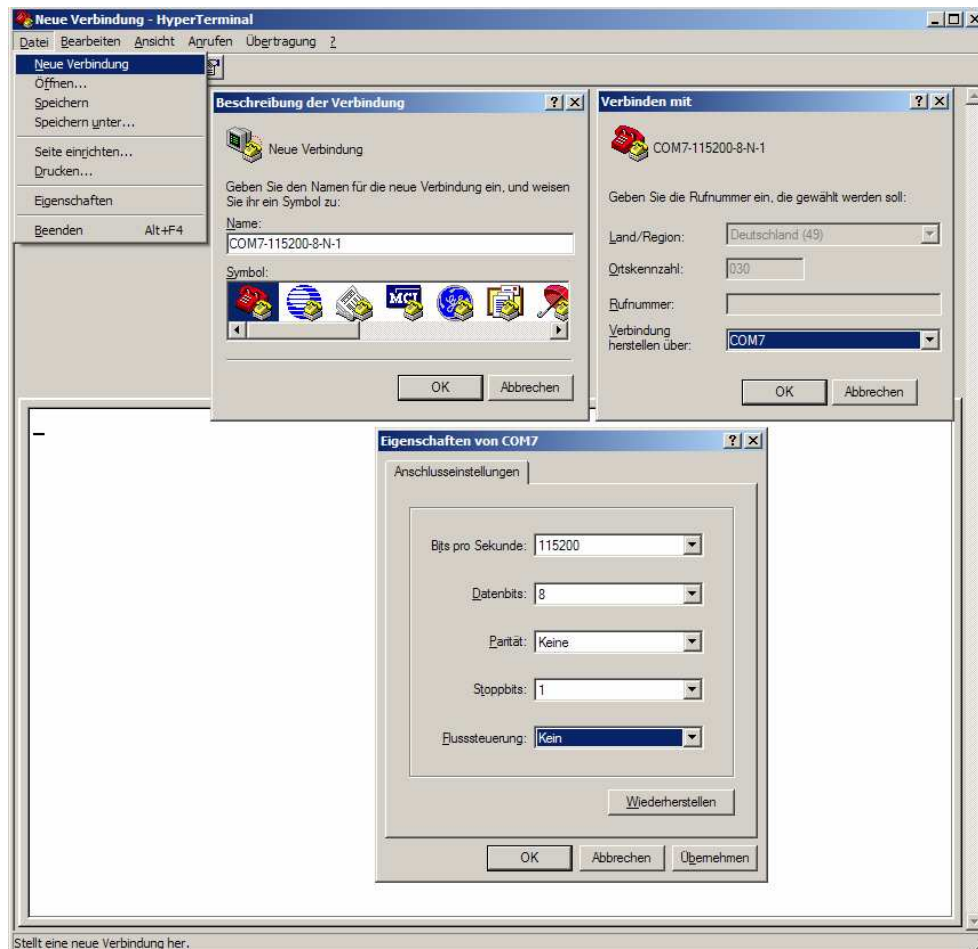


Damit sind alle Voraussetzungen zum JTAG-Flashen geschaffen.



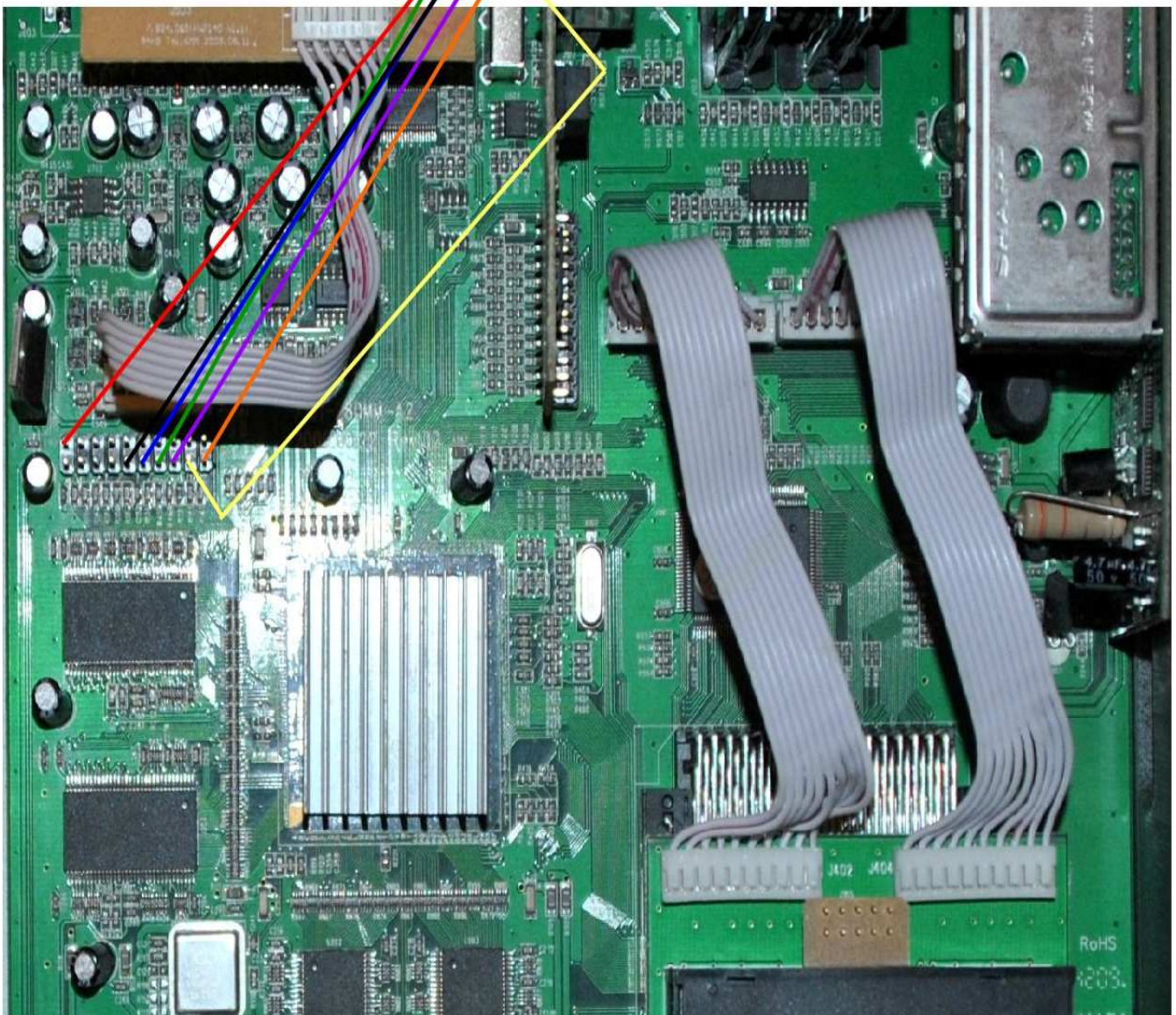
3.2 Serielle RS-232 Daten-Verbindung

Für das JTAG-Flashen des Bootloaders muss der serielle Anschluss des Receivers mit Hilfe eines Nullmodem-Kabels mit einem COM-Port des Computers verbunden werden (ist auch mit dem „ELV TTL - nach RS 232-Umsetzer“ in Verbindung mit dem virtuellen COM-Port des „ELV Highspeed Mini USB Modul UM-FT2232H“ realisierbar) und ein Terminal-Programm wie z.B. Hyperterminal von WindowsXP mit den Übertragungsparametern **115200 Baud**, **8 Datenbits**, **keine Parität**, **1 Stoppbit** und **keine Flussteuerung** für den verwendeten COM-Port gestartet werden, damit man die serielle Verbindung mit dem Receiver nutzen kann:



3.3 JTAG Daten-Verbindung

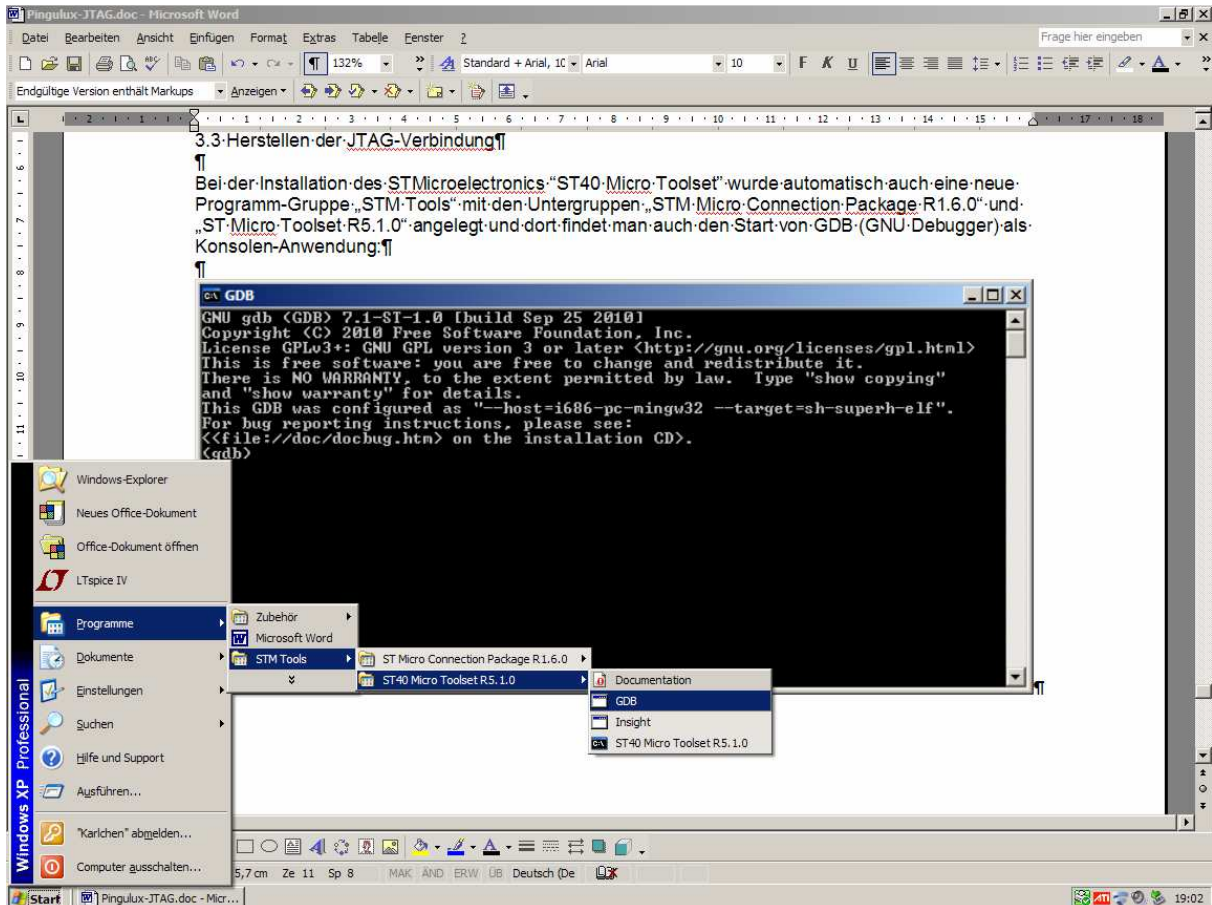
Der Receiver muss noch mit dem „ELV Highspeed Mini USB Modul UM-FT2232H“ als „ST Micro Connect Lite“ JTAG-Adapter verbunden sein, wobei nur die gleichnamigen JTAG-Signale **GND**, **TCK**, **TDO**, **nTRST**, **nSRST**, **TDI** und **TMS** des JTAG-Interfaces vom Receiver nötig sind und die restlichen JTAG-Signale (nASEBRK, TRIGIN und TRIGOUT) werden nicht verwendet:



3.4 Das Flashen des u-boot Bootloaders

Das Flashen des u-boot Bootloaders erfolgt in drei wesentlichen Schritten: Herstellen der JTAG-Kommunikation mit dem Receiver-Chip, laden und starten eines rudimentären Bootloaders im RAM des Receivers und mit diesem dann laden und flashen des eigentlichen Original-Bootloaders.

Bei der Installation des STMicroelectronics "ST40 Micro Toolset" wurde automatisch auch eine neue Programm-Gruppe „STM Tools“ mit den Untergruppen „STM Micro Connection Package R1.6.0“ und „ST Micro Toolset R5.1.0“ angelegt und dort findet man auch den Start von GDB (GNU Debugger) als Konsolen-Anwendung:



Im ersten Schritt wird die JTAG-Kommunikation mit dem Receiver-Chip hergestellt und dafür gibt man in der GDB-Konsole den Befehl

```
sh4tp STMCLT1000A:mb442:st40
```

ein, wobei die genaue Groß- und Kleinschreibung zu beachten ist.

Bei Ausführung des Befehls wird über die JTAG-Signale ein RESET-des Receiver-Chips ausgelöst und sofort die volle Kontrolle der Funktion des Receiver-Chips übernommen, so dass auch ein „zerflashter“ Bootloader nicht gestartet werden kann. Über die JTAG-Verbindung wird der Receiver-Chip vollkommen neu initialisiert, so dass zumindest der Memory-Controller des Receiver-Chips mit dem Flash-Chip für den Bootloader und mit den RAM-Chips voll funktionsfähig ist.

Wichtig hierbei ist jetzt die Eingabe des ersten Befehl's (rot markierte Zeile).

Das als Nächstes benötigtes rudimentäres „u-boot“ speichern wir uns an einem bestimmten Ort ab & merken uns auch dieses Speicherort. Den Pfad zum Speicherort des u-boot's brauchen wir für den nächsten Schritt, damit wir dieses rudimentäre u-boot in den RAM des Receiver's zu laden.

```
GDB
GNU gdb (GDB) 7.1-ST-1.0 [build Sep 25 2010]
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=sh-superh-elf".
For bug reporting instructions, please see:
<<file://doc/docbug.htm> on the installation CD>.
(gdb) sh4tp STMCLT1000A:mb442:st40
The target is assumed to be little endian
The target architecture is assumed to be sh4
mb442_stx7109_cut30 (mb442) connect start - parameters {}
Initialization TCK frequency set to 30000000 Hz
Device id 0x2d42c041
tapmux connect(): boot mode single core setup
tapmux setup to bypass to core st40, channel 1
mb442 initialization start with SoC stx7109_cut30 ...
stx7109_cut30: booted audio companion
stx7109_cut30: booted video companion
TCK frequency set to 100000000 Hz
tapmux complete_connect(): single core setup
mb442 initialization complete
0xa0000000 in ?? ()
(gdb) load C:\Optiboot\750000\sh-u-boot.elf
Loading section .text, size 0x1c4a8 lma 0x87f00000
Loading section .rodata, size 0x1268 lma 0x87f1c4a8
Loading section .rodata.str1.4, size 0x7024 lma 0x87f1d710
Loading section .data, size 0xaf0 lma 0x87f24734
Loading section .u_boot_cmd, size 0x620 lma 0x87f25224
Start address 0x87f00000, load size 153668
Transfer rate: 237 KB/sec, 30733 bytes/write.
(gdb) continue
Continuing.
```



Im nächsten Schritt wird ein rudimentäres u-boot als Bootloader über die JTAG-Verbindung in den RAM des Receivers geladen. Dieses rudimentäre u-boot ist als **u-boot.o** Datei im Paket dieser Anleitung enthalten. Dazu gibt man in der GDB-Konsole den Befehl:

```
load C:\\\\Opticum9600HD\\\\u-boot.o
```

ein.

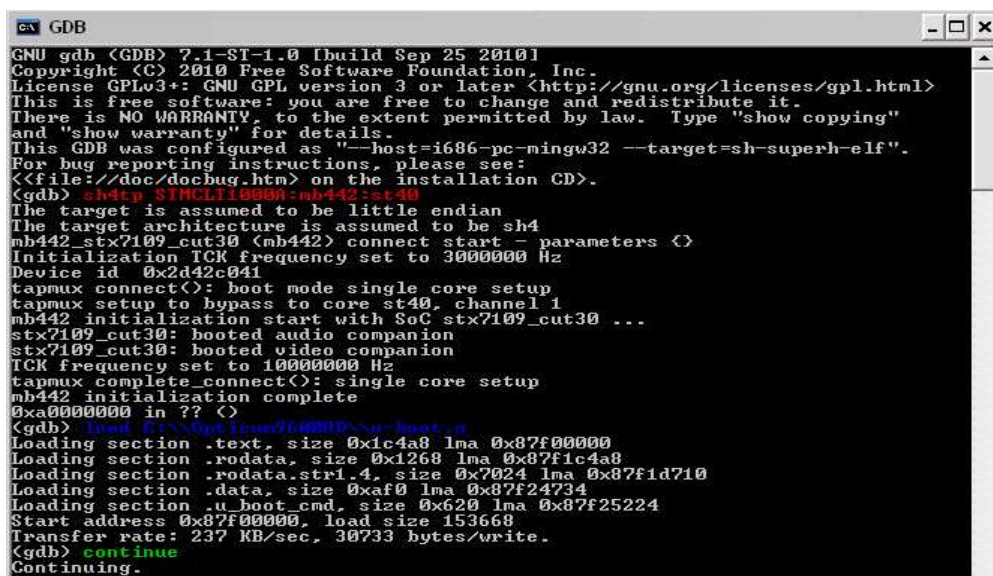
Der Pfad zur **u-boot.o** Datei muss natürlich dem Ort entsprechen, wo sich die entpackte Datei befindet. Man beachte, dass der Windows-Backslash in der Pfadangabe doppelt geschrieben werden muss, da GDB eigentlich eine Linux-Anwendung ist, die nur für Windows portiert wurde.

Nun muss man nur noch das rudimentäre u-boot im RAM des Receivers mit dem GDB-Befehl

```
continue
```

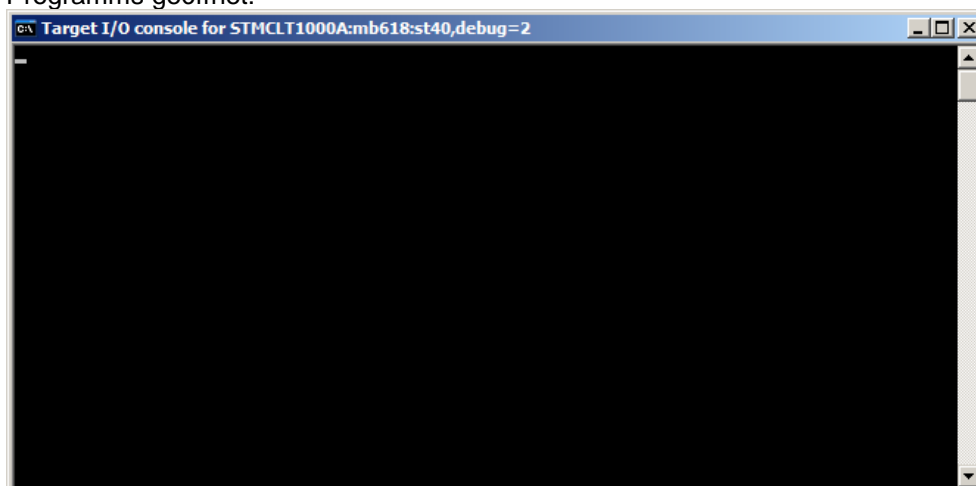
starten:

Blau markierte Befehlszeile & grün.



```
GNU gdb (GDB) 7.1-SI-1.0 [build Sep 25 2010]
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-mingw32 --target=sh-superh-elf".
For bug reporting instructions, please see:
<<file://doc/docbug.htm> on the installation CD>.
(gdb) sh4tp STMCL11000A:mb442:st40
The target is assumed to be little endian
The target architecture is assumed to be sh4
mb442_stx7109_cut30 (mb442) connect start - parameters <
Initialization TCK frequency set to 3000000 Hz
Device id 0x2d42c041
tapmux connect(): boot mode single core setup
tapmux setup to bypass to core st40, channel 1
mb442 initialization start with SoC stx7109_cut30 ...
stx7109_cut30: booted audio companion
stx7109_cut30: booted video companion
TCK frequency set to 10000000 Hz
tapmux complete_connect(): single core setup
mb442 initialization complete
0xa0000000 in ?? <
(gdb) load C:\\\\Opticum9600HD\\\\u-boot.o
Loading section .text, size 0x1c4a8 lma 0x87f00000
Loading section .rodata, size 0x1268 lma 0x87f1c4a8
Loading section .rodata.str1.4, size 0x7024 lma 0x87f1d710
Loading section .data, size 0xaf0 lma 0x87f24734
Loading section .u_boot.cmd, size 0x620 lma 0x87f25224
Start address 0x87f00000, load size 153668
Transfer rate: 237 KB/sec, 30733 bytes/write.
(gdb) continue
Continuing.
```

Dabei wird ein weiteres Konsolen-Fenster für eventuelle Ein-/Ausgaben des im RAM gestarteten Programms geöffnet:



Da u-boot für Ein-/Ausgaben die serielle Verbindung benutzt, können die beiden Konsolen-Fenster minimiert werden, dürfen aber auf keinen Fall geschlossen werden. Jetzt sollte man im Hyperterminal-Fenster den erfolgreichen Start des rudimentären u-boot im RAM des Receivers anhand der Ausgaben von u-boot sehen bzw. verfolgen können:

```

COM6-115200-8-N-1 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
[Icons]
è•B0002000002c97
Board: STb7100-Reference (MB442) [29-bit mode]
U-Boot 1.3.1 (Jun 14 2012 - 18:14:38) - stm24_0057
DRAM: 64 MiB
NOR: 8 MiB
*** Warning - bad CRC, using default environment
In: serial
Out: serial
Err: serial
MB442>
Verbinden 00:53:22 Auto-Erkenn. 115200 8-N-1 RF GROSS NUM Aufzeichnen Druckerecho

```

Mit dem u-boot Befehl

loady

kann man nun das vollständige u-boot als serielle Übertragung der beigefügten **u-boot.bin** Datei mit dem Ymodem-Protokoll in den RAM des Receivers laden:

Dieser nachfolgende Screenshot entspricht jetzt nicht dem des Opticum`s , aber es geht hier um die Verfahrensweise des „Datei senden“. Man muss sich hier natürlich das Passende für den Opticum auswählen.

```

COM7-115200-8-N-1 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
[Icons]
Datei senden...
Datei empfangen...
Text aufzeichnen...
Textdatei senden...
Am Drucker aufzeichnen
Datei senden
Ordner: F:\JTAG_pingulux
Dateiname: F:\JTAG_pingulux\u-boot bin [Durchsuchen...]
Protokoll: Ymodem
Senden Schließen Abbrechen
Ymodem Dateiversand für COM7-115200-8-N-1
Senden: F:\JTAG_pingulux\u-boot bin
Paket: 55 Fehlerprüfung: CRC Dateigröße: 512 KB
Wiederholungen: 0 Wiederholungen insgesamt: 0 Dateien: 1 von 1
Letzter Fehler:
Datei: **** 50K von 512K
Dauer bisher: 00:00:10 Restdauer: 00:01:32 Durchsatz: 5120 cps
Abbrechen gpa/bsp
Board: STx7111-Mboard (MB618) [29-bit mode]
info: Disregarding any EPLD
U-Boot 1.3.1 (May 29 2012 - 20:15:43)
DRAM: 128 MiB
NOR: 8 MiB
*** Warning - bad CRC, using default environment
In: serial
Out: serial
Err: serial
MB618> loady
## Ready for binary (ymodem) download to 0x8C000000 at 115200 bps...
CC
Sendet eine Datei zum Remotesystem.

```



Nachdem die **u-boot.bin** Datei erfolgreich übertragen wurde muss deren Inhalt noch als Bootloader geflasht werden. Dafür enthält das rudimentäre u-boot zwei Script-Befehle, die man mit dem u-boot run-Befehl nacheinander starten muss. Im ersten Schritt wird mit:

Protect off bank 1

zunächst der Schreibschutz der nötigen Sektoren des Flash-Chips deaktiviert. Im zweiten Schritt erfolgt dann mit:

Erase bank 1

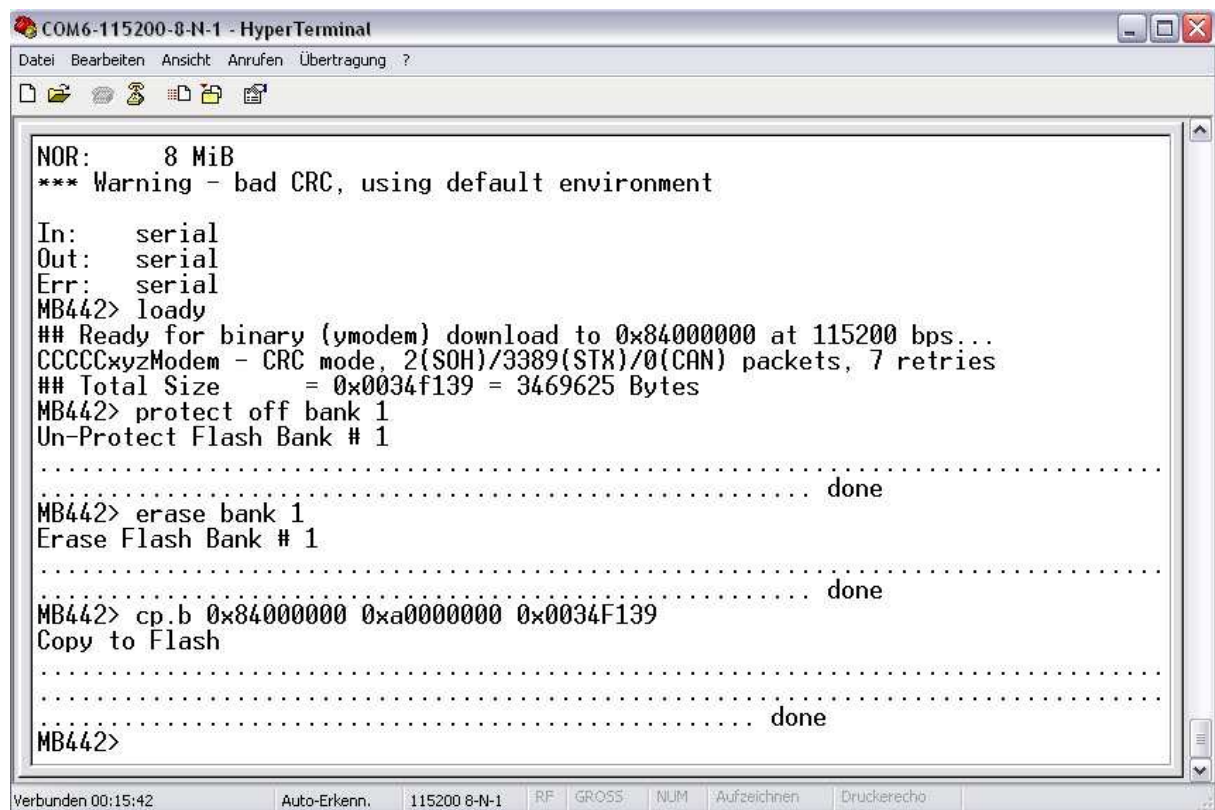
erst das Löschen der nötigen Sektoren des Flash-Chips, dann das Flashen des Bootloaders und zum Schluss wird wieder der Schreibschutz der Sektoren des Flash-Chips aktiviert:

cp.b 0x84000000 0xa0000000 0x0001bcd

Mit diesem Befehl wird dann erst der komplette Bootloader in den RAM des Receivers geschrieben.

Auf diese gleiche Art & Weise kann man anstatt des Bootloaders auch gleich die Originalfirmware auf den Receiver übertragen. Der einzige Unterschied dabei ist hier die letzte Befehlszeile. Da ja die Originalfirmware größer ist, von der Datei her, als der Bootloader, muss sich da natürlich der Endwert ändern. Würde dann z.B. so aussehen:

cp.b 0x84000000 0xa0000000 0x0034F139



```
COM6-115200-8-N-1 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
NOR:      8 MiB
*** Warning - bad CRC, using default environment

In:      serial
Out:     serial
Err:     serial
MB442> loady
## Ready for binary (ymodem) download to 0x84000000 at 115200 bps...
CCCCxyzModem - CRC mode, 2(SOH)/3389(STX)/0(CAN) packets, 7 retries
## Total Size      = 0x0034f139 = 3469625 Bytes
MB442> protect off bank 1
Un-Protect Flash Bank # 1
..... done
MB442> erase bank 1
Erase Flash Bank # 1
..... done
MB442> cp.b 0x84000000 0xa0000000 0x0034F139
Copy to Flash
..... done
MB442>
```



Fertig!

Jetzt kann man die beiden Konsolen-Fenster schließen, den Receiver aus- und wieder einschalten und sollte im Hyperterminal den normalen Bootvorgang beobachten können. Auch der Receiver sollte nun wieder ganz normal booten.

Nun kann man wieder seine eventuell eigenen, gesicherten Daten per USB auspielen.

Viel Erfolg

Daniel & Rimini at Nachtfalke-Forum

Nochmals, hier an dieser Stelle ein „GROSSES DANKE SCHÖN“ an Rimini, der hier ein super Vorarbeit geleistet hat, ohne die ich das hier hätte nicht machen können.

Nachtrag:

Die passende „u-boot.o“ , den Bootloader & auch die Originalfirmware werde ich noch zu einem Paket packen und in die Database laden.

