

JTAG-Flash für Edision argus Pingulux Receiver

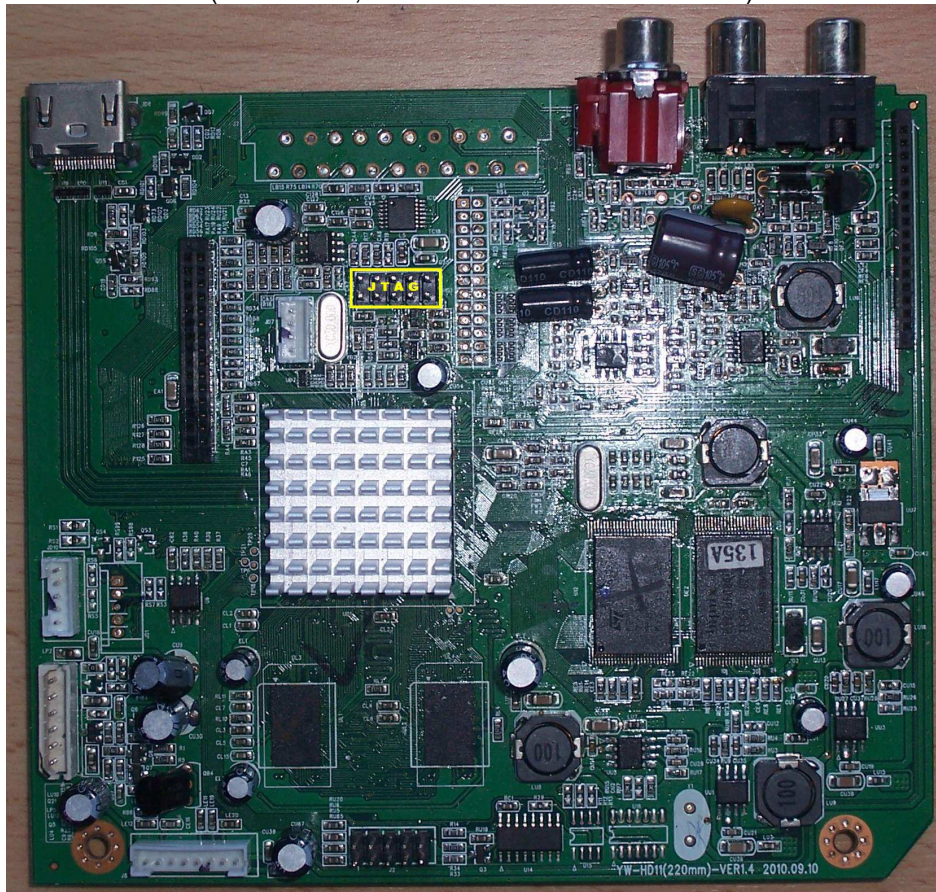
0. Vorwort

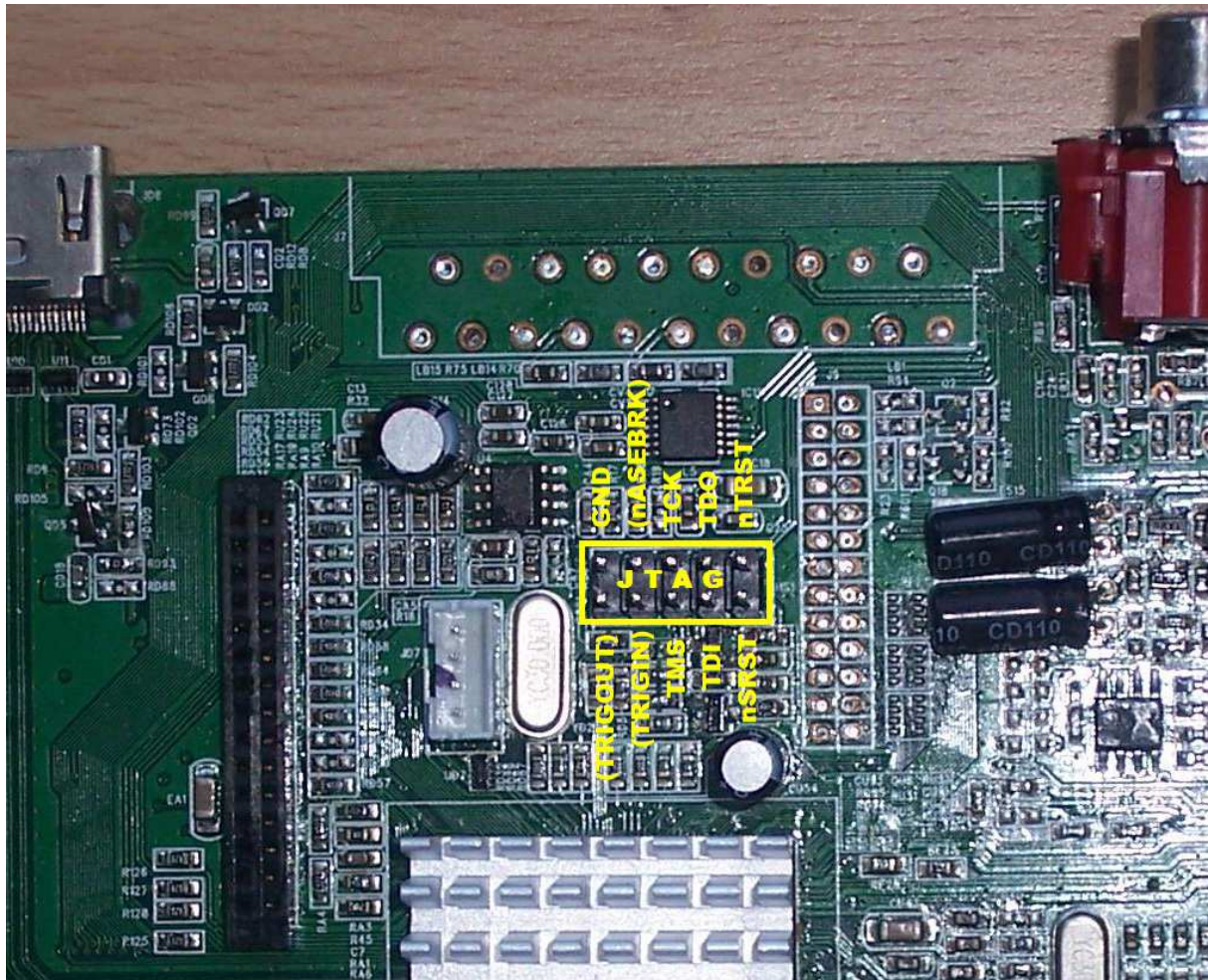
Die Edision argus Pingulux Receiver gehören in die Klasse der „SPARK-Receiver“ von Fulan und basieren auf einem STi7111 Receiver-Chip von STMicroelectronics. Kern dieser Receiver-Chips ist eine ST40 Core-CPU für die allgemeinen Receiver-Funktionen, die ihren Ursprung wiederum in einer Hitachi SH4 Core-CPU hat, kombiniert mit zwei ST231 Core-CPUs, die als freiprogrammierbare Video- und Audio-Codecs von der Receiver-Firmware verwendet werden. 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 FDTI 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 Edision argus Pingulux

Receiver-Platine (ohne Tuner, Ethernet-Modul und Cardreader)

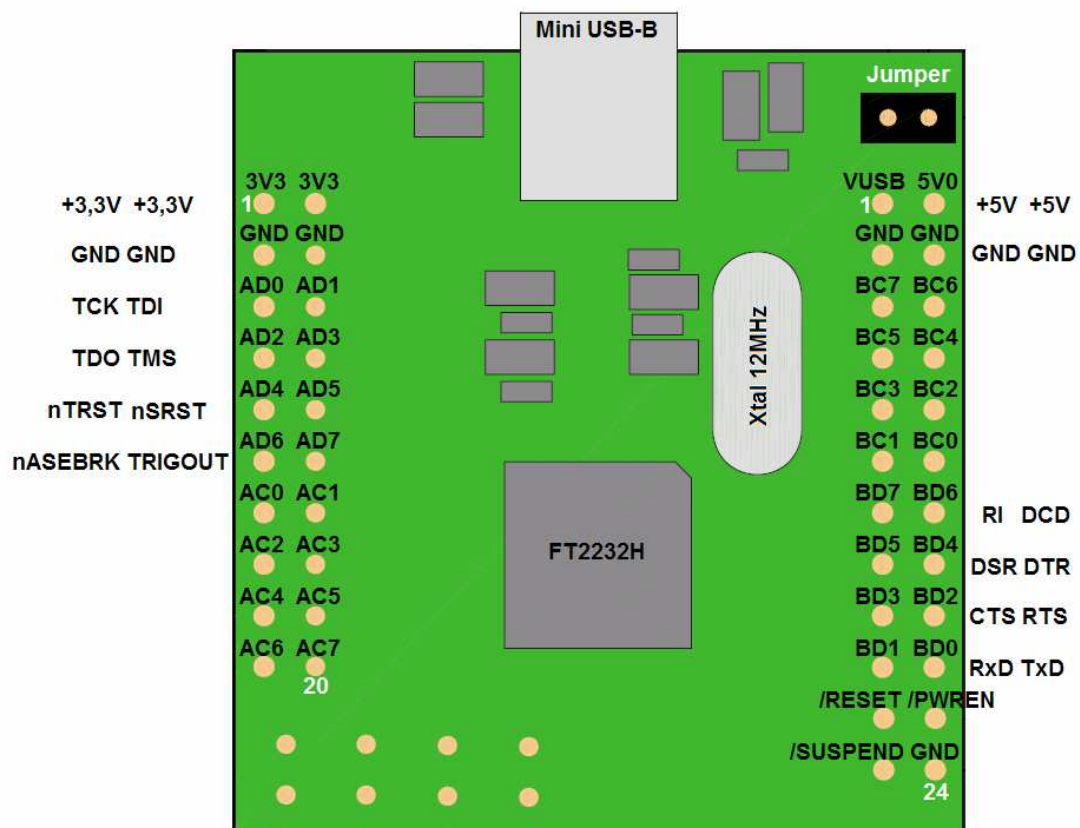
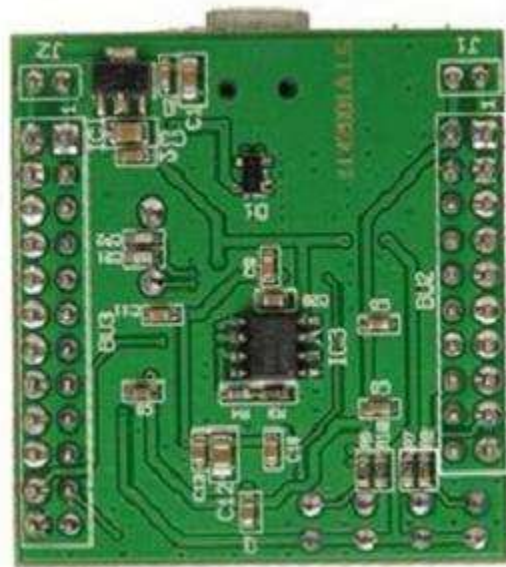
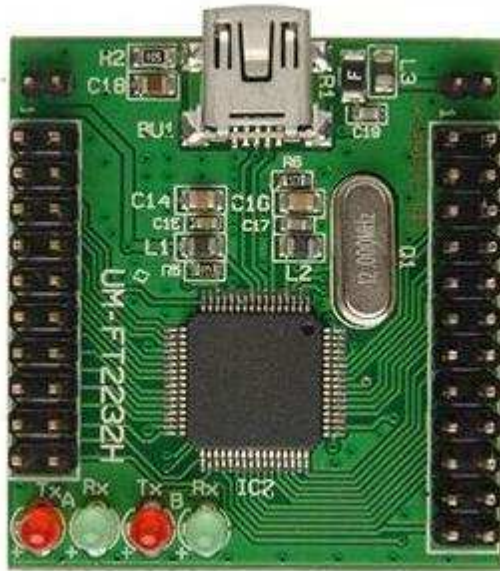




1.2. JTAG-Adapter

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 von der ELV Elektronik AG als „ELV Highspeed Mini USB Modul UM-FT2232H, Komplettbausatz“ (Artikel-Nr.: 68-09 93 47). 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 tolerantan 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“ (Artikel-Nr.: 68-03 84 39), der sogar den Vorteil hat, dass er eine 9-polige Buchse für die RS-232 Signale besitzt und man kein zusätzliches Nullmodem-Kabel benötigt.





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 Micro Toolset“ (ab R5.1.0) in Verbindung mit dem „ST Micro Connection Package“ von STMicroelectronics unter **WindowsXP** (ein 32Bit Windows 7 sollte auch verwendbar sein, aber kein 64Bit Windows) 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 die nötigen Tools und Programme auf dem FT-Server vollkommen frei zugänglich gemacht:

<ftp://ftp.stlinux.com/pub/tools/products/>

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 vorbereite 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:

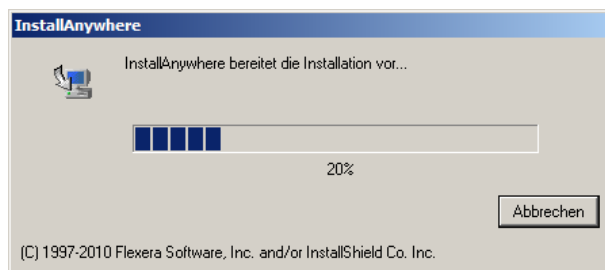
1. Man muss die „ST License“ akzeptieren, damit man bei der Installation fortsetzen kann.
2. Es macht für die Hobby-Nutzung keinen Sinn, die Installation zu registrieren und muss/sollte deshalb die vorgegebene Auswahl zur Registrierung deaktivieren.

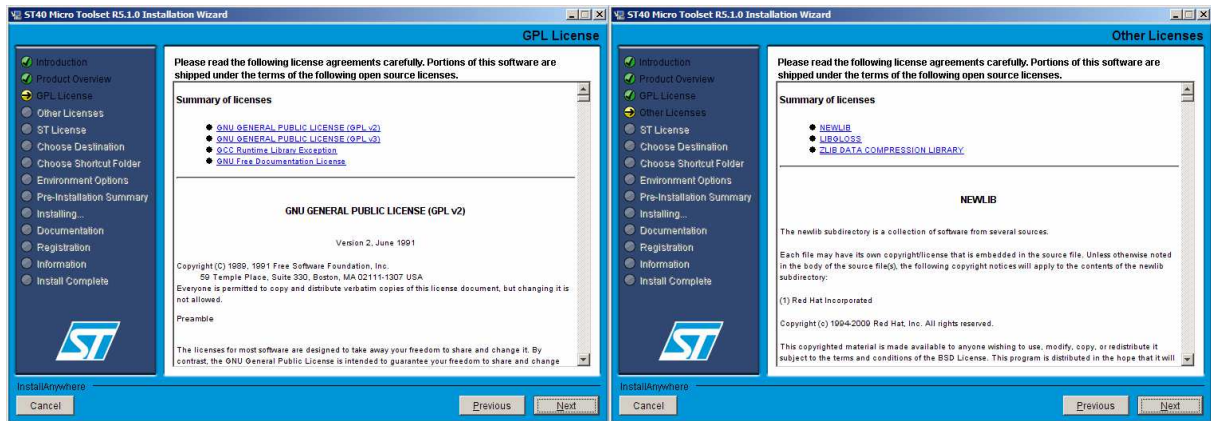
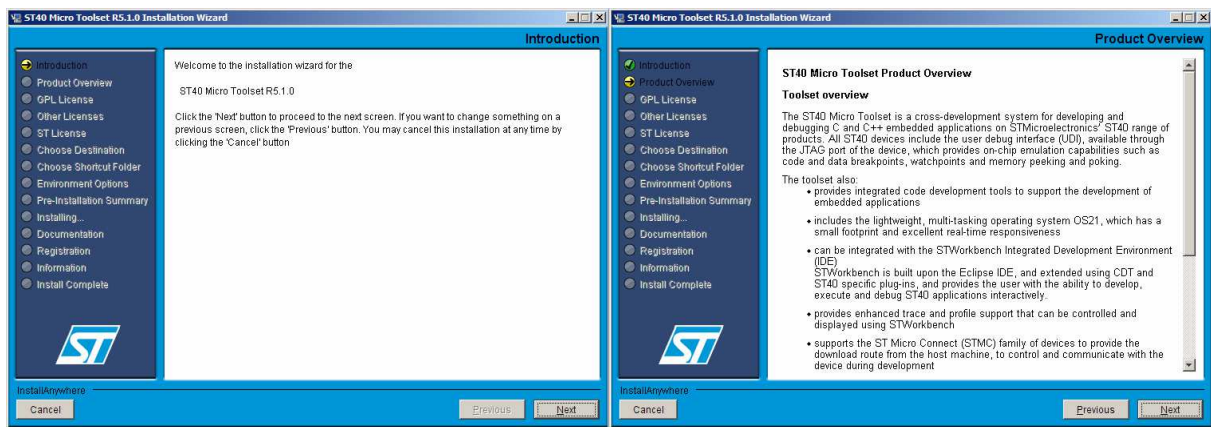
Ob man sich den Index für die installierten Dokumentationen vor der Registrierung anzeigen lässt oder nicht, kann man selber entscheiden. Darauf hat man nach abgeschlossener Installation auch später jederzeit über das „Programm-Menü“ von Windows einen Zugriff.

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.

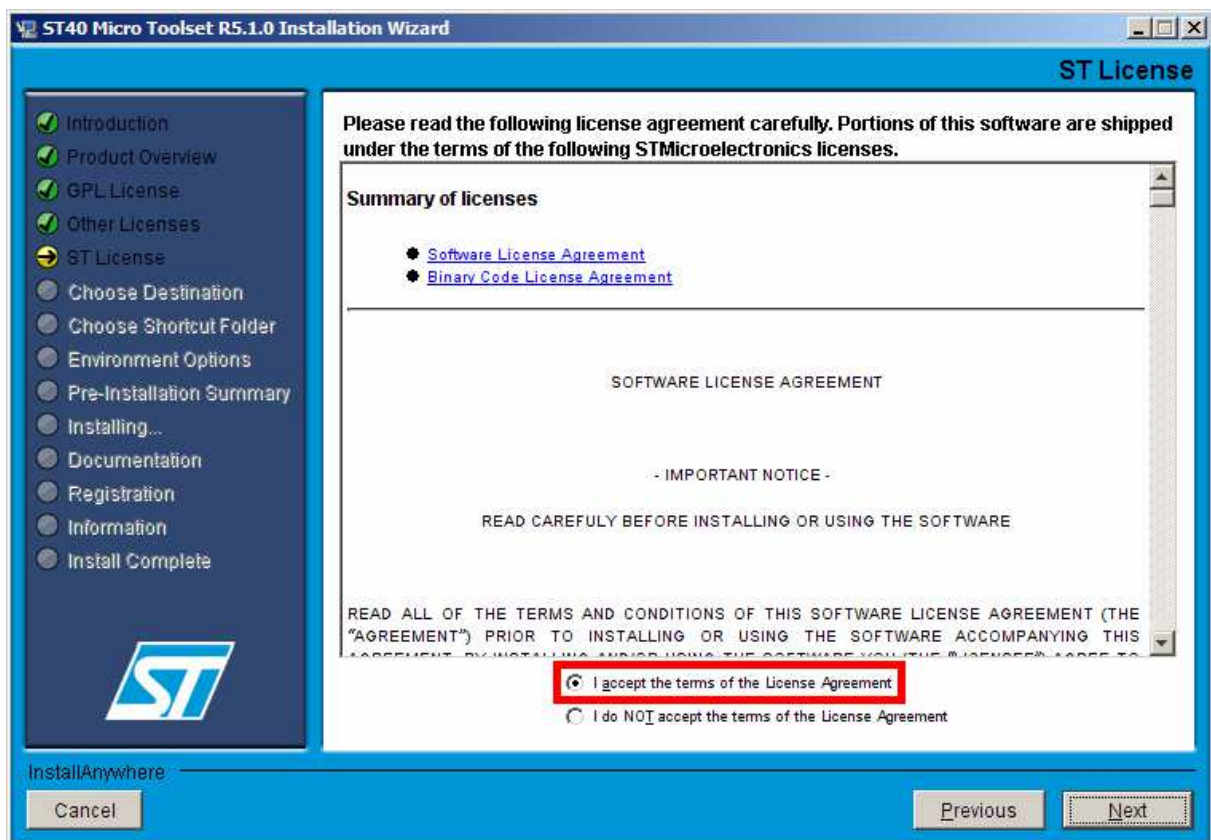
Der Abschluss der Installation dauert je nach Leistungsfähigkeit des verwendeten PCs ein paar Sekunden. Also warten und nicht ungeduldig sein.

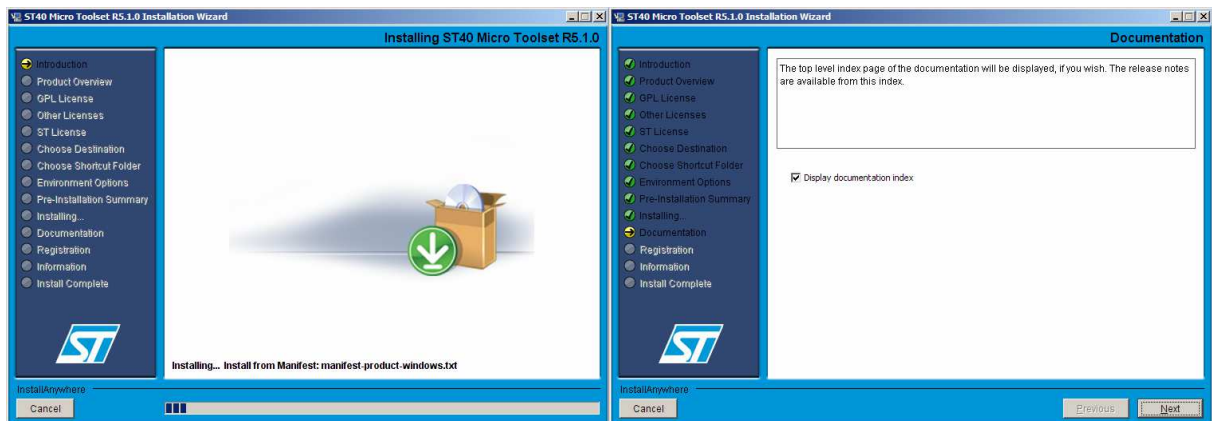
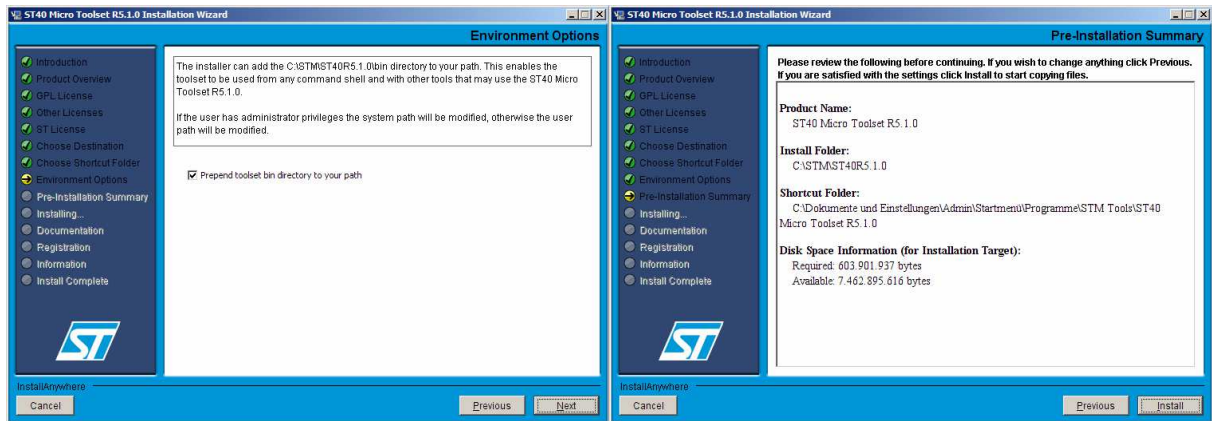
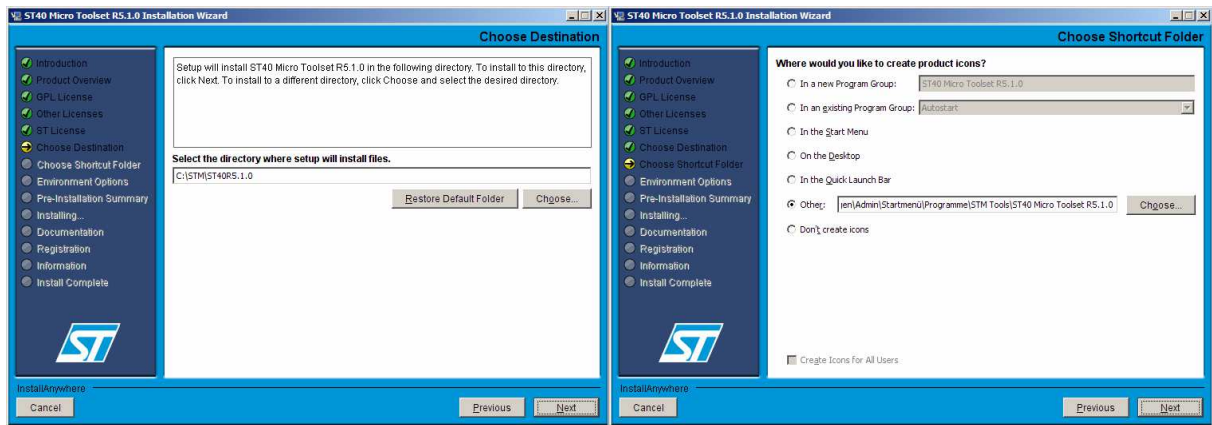
Los geht es:



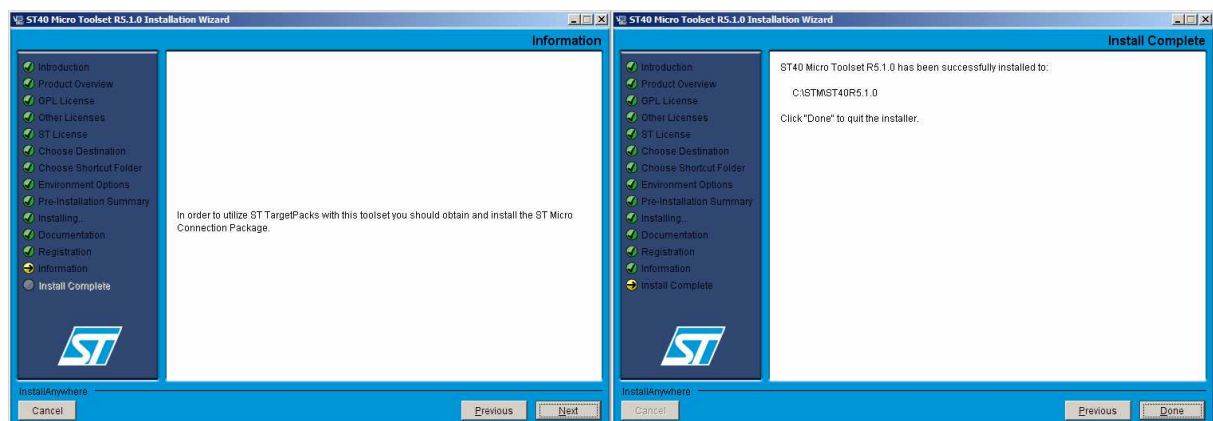
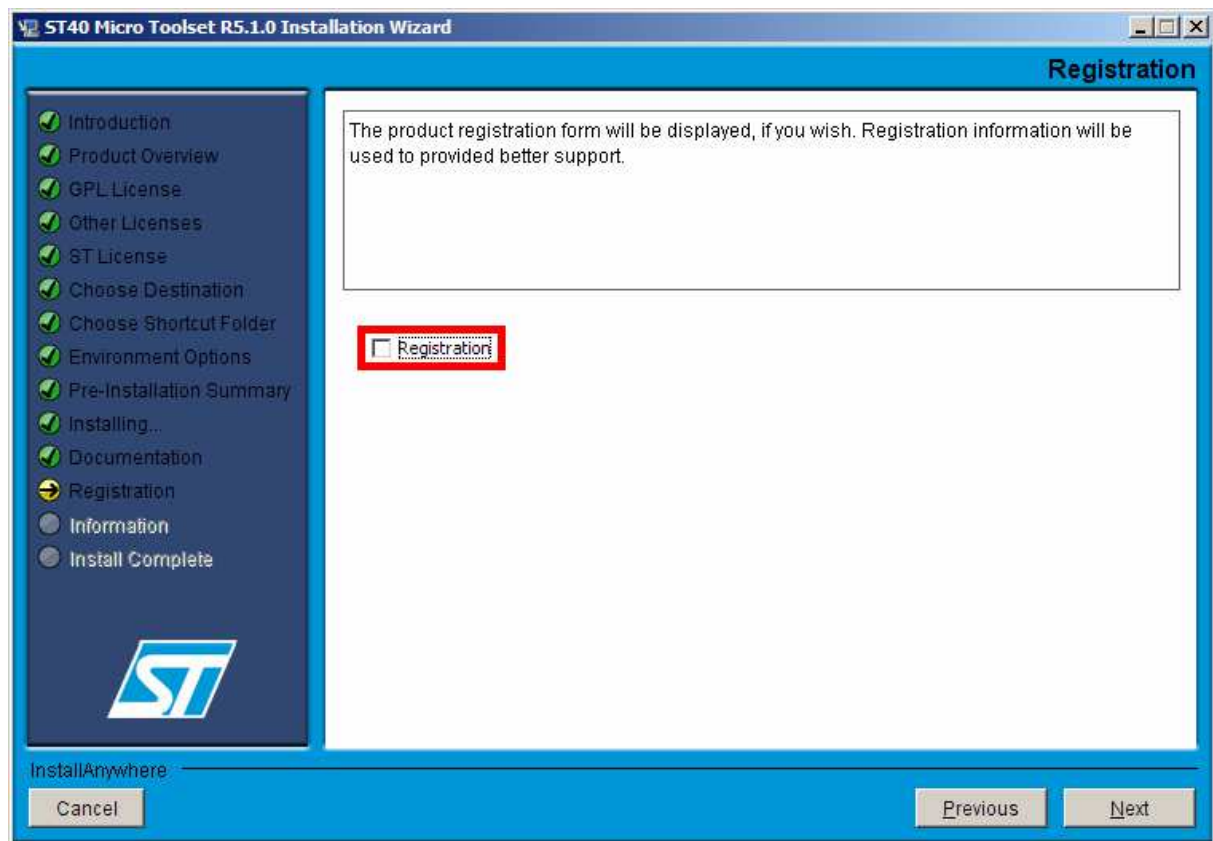


ST License muss man akzeptieren, um fortsetzen zu können:





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

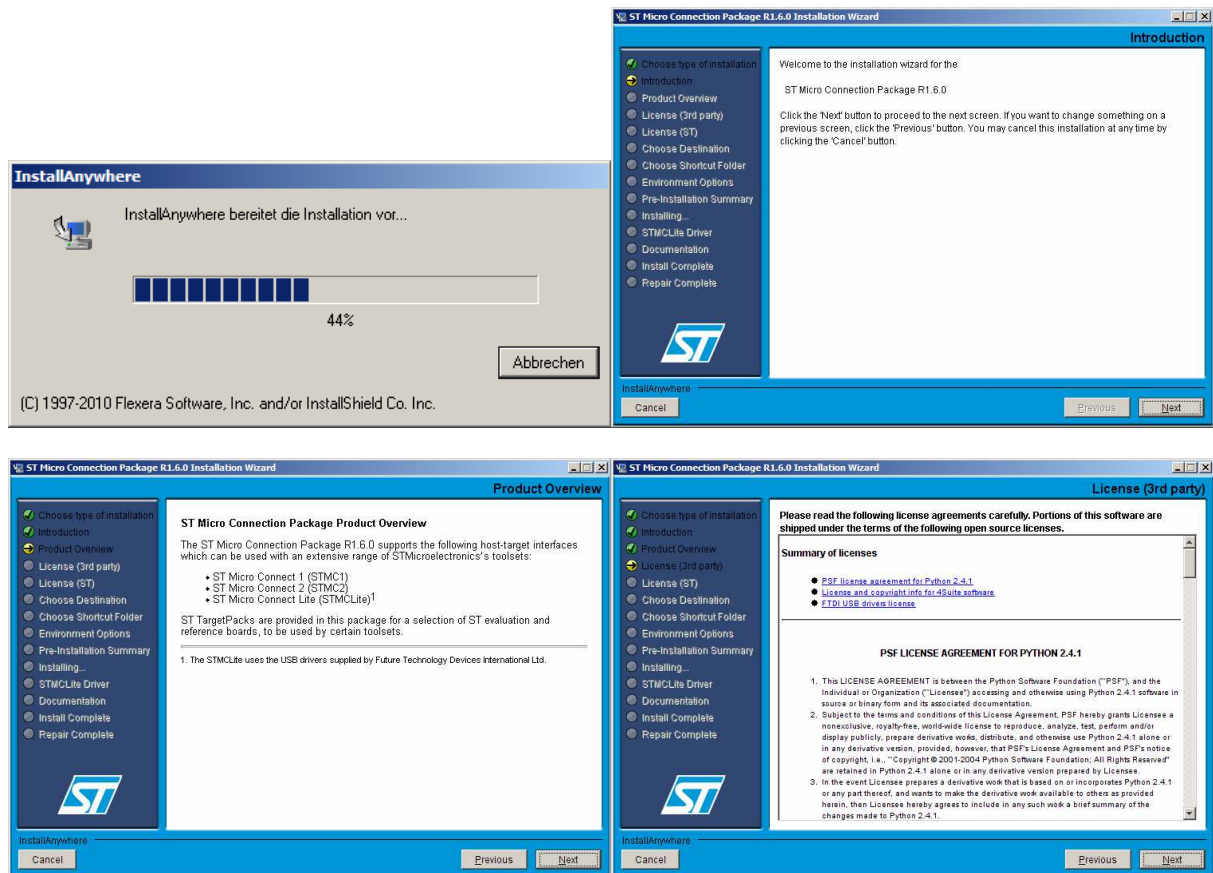


Nach etwas Wartezeit ist die Installation dann beendet.

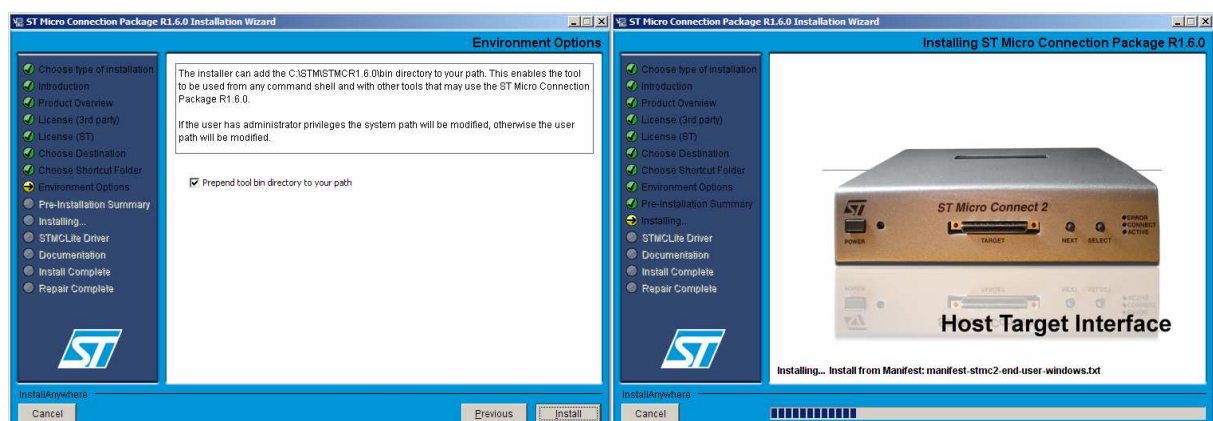
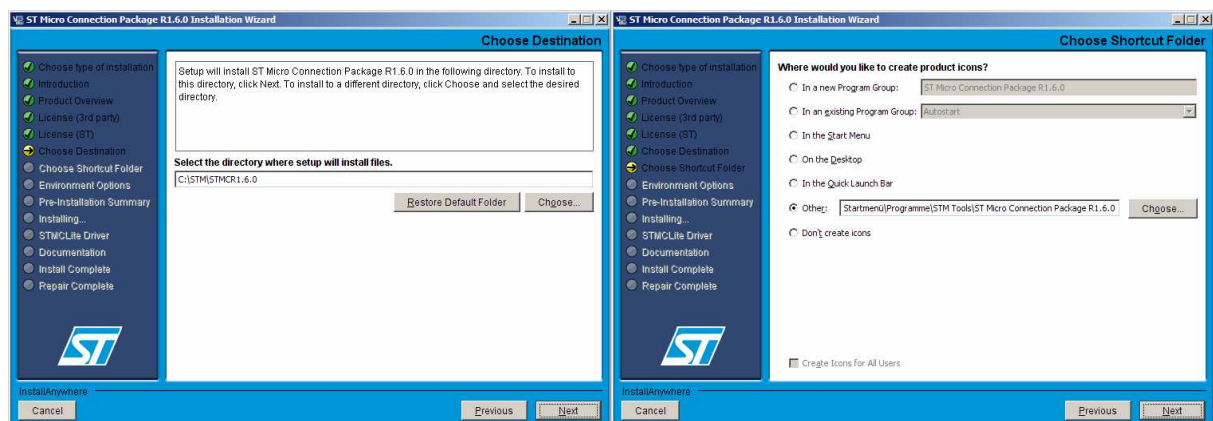
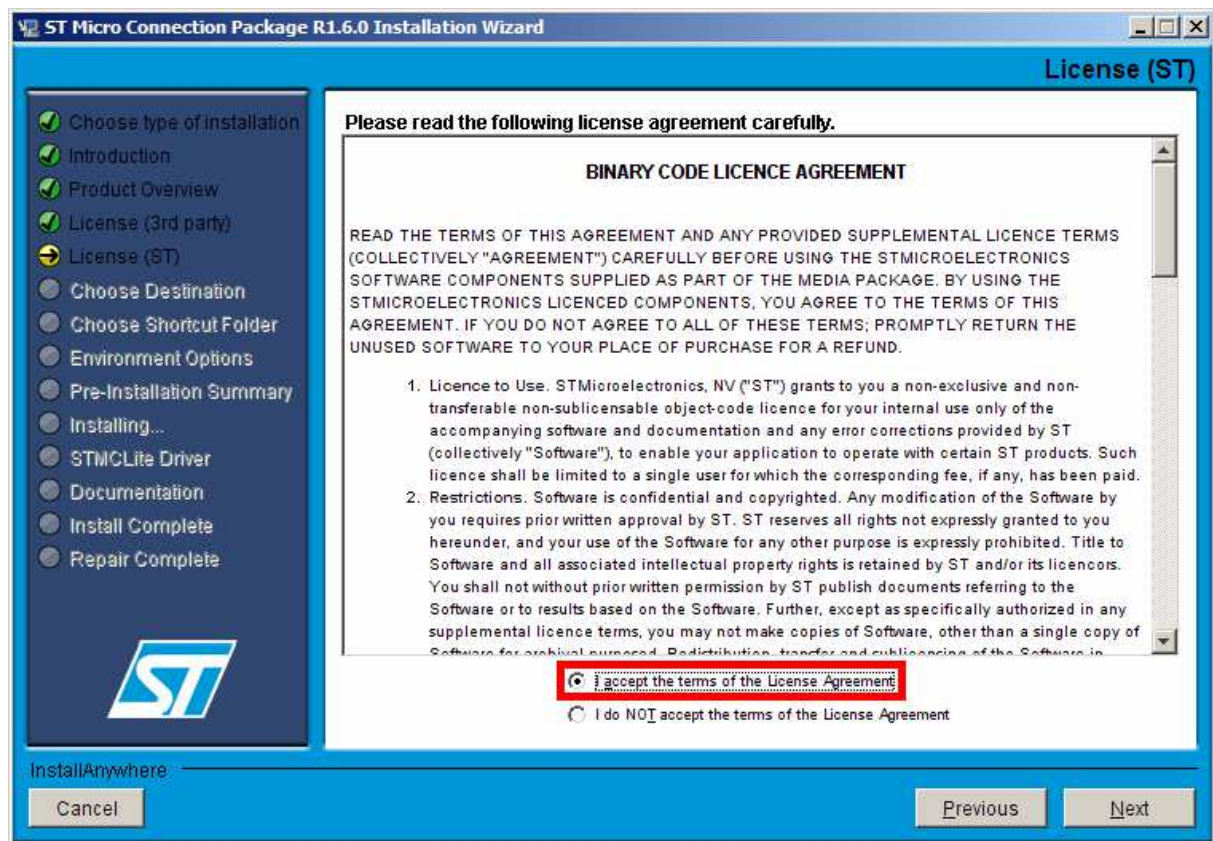


2.2 Das STMicroelectronics "ST Micro Connection Package"

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:

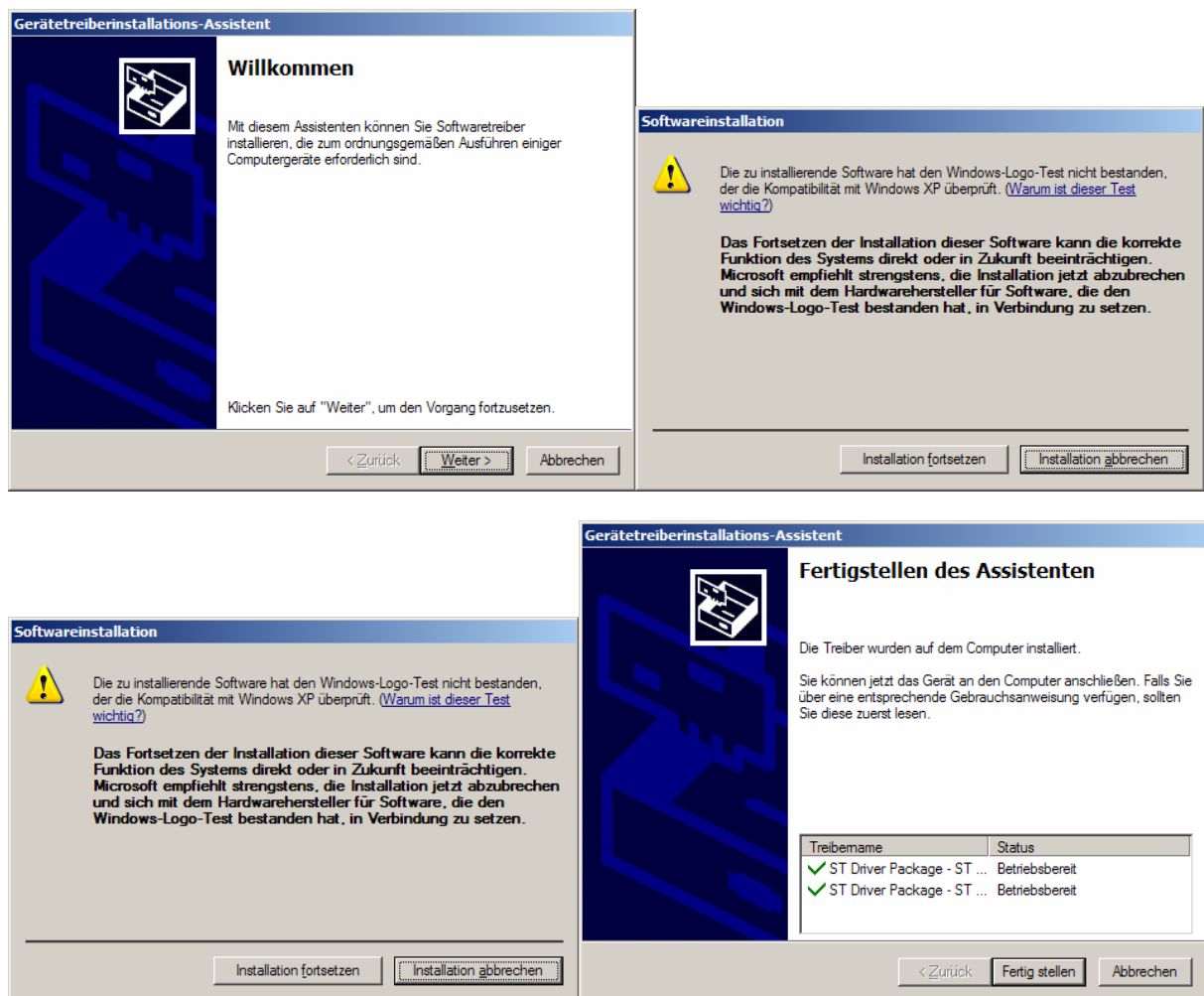


Die License muss man wieder akzeptieren, um fortsetzen zu können:



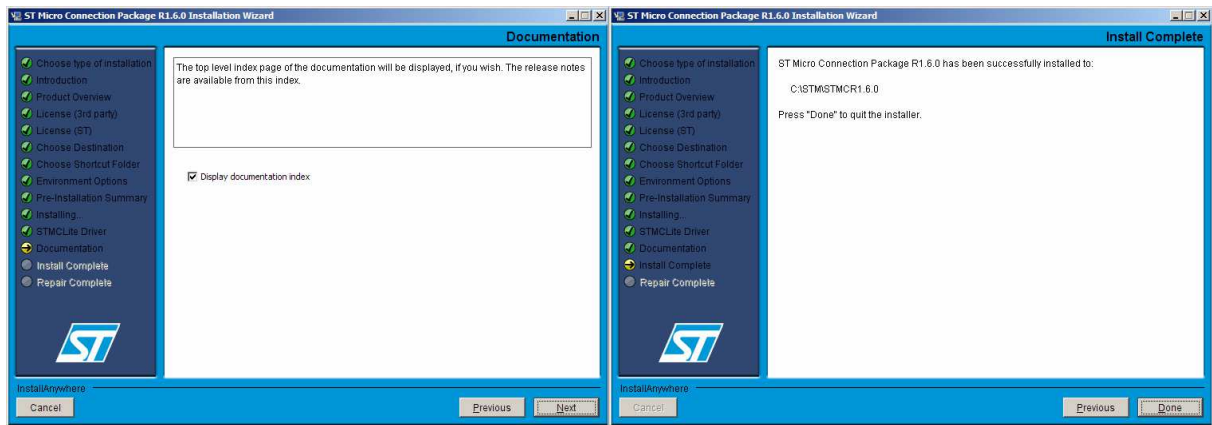


Wenn man im vorherigen Schritt nicht gerade die Installation der Windows-Treiber für den „ST Micro Connect Lite“ Adapter deaktiviert hat, wird der Gerätetreiberinstallations-Assistent von Windows automatisch gestartet und man muss der Installation der unsignierten Treiber zweimal zustimmen:



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

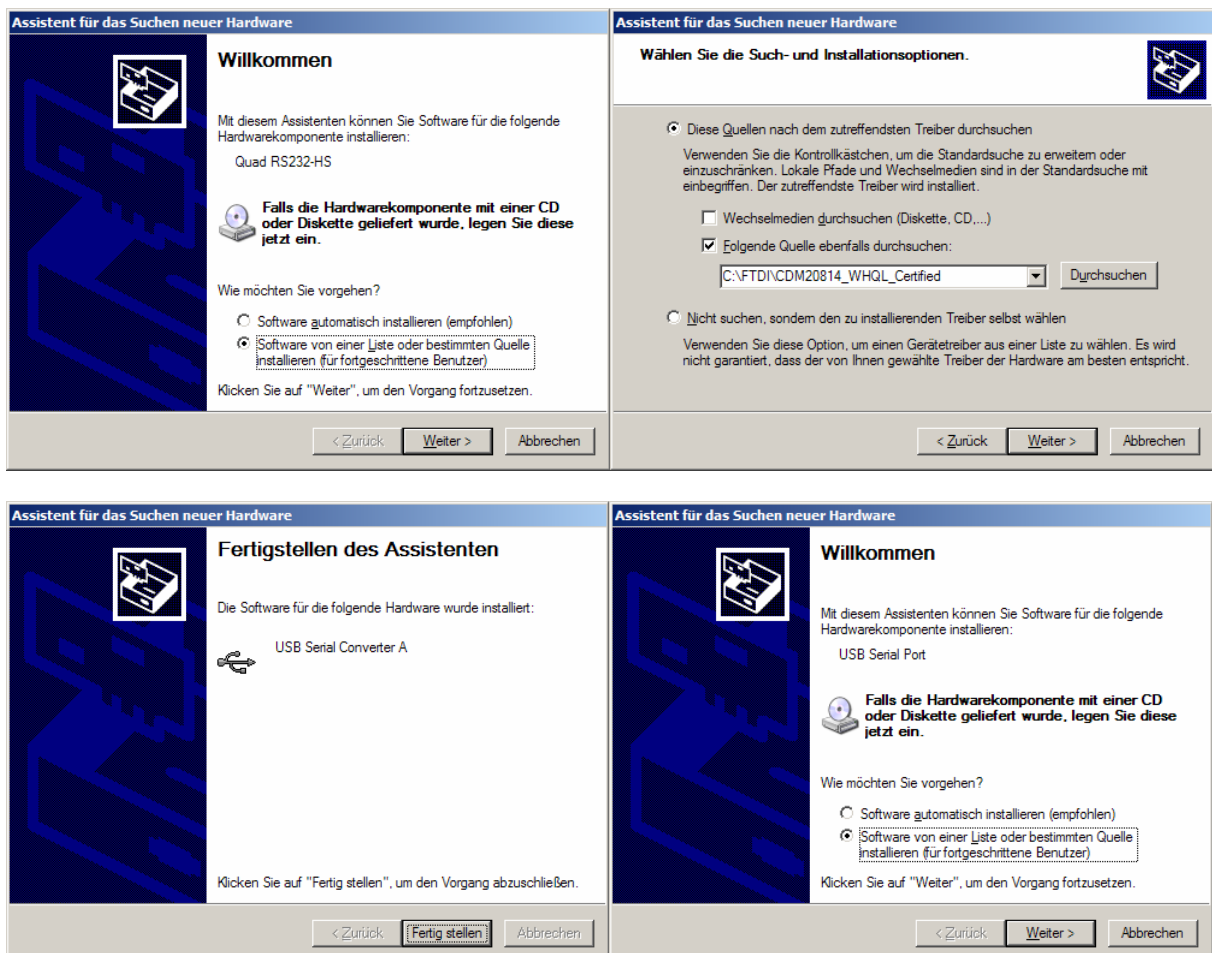


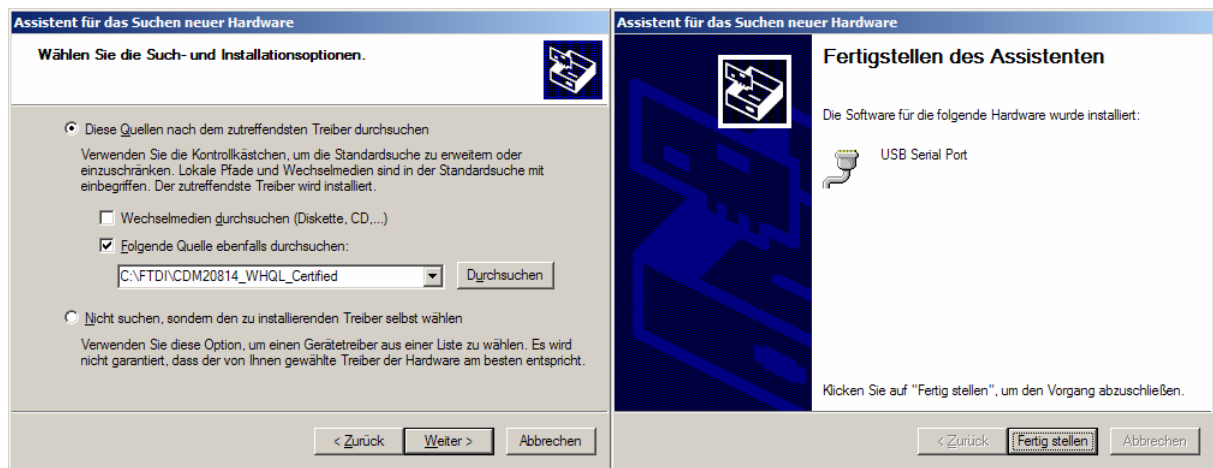


2.3 Die FDTI „MProg 3.5“ Utility

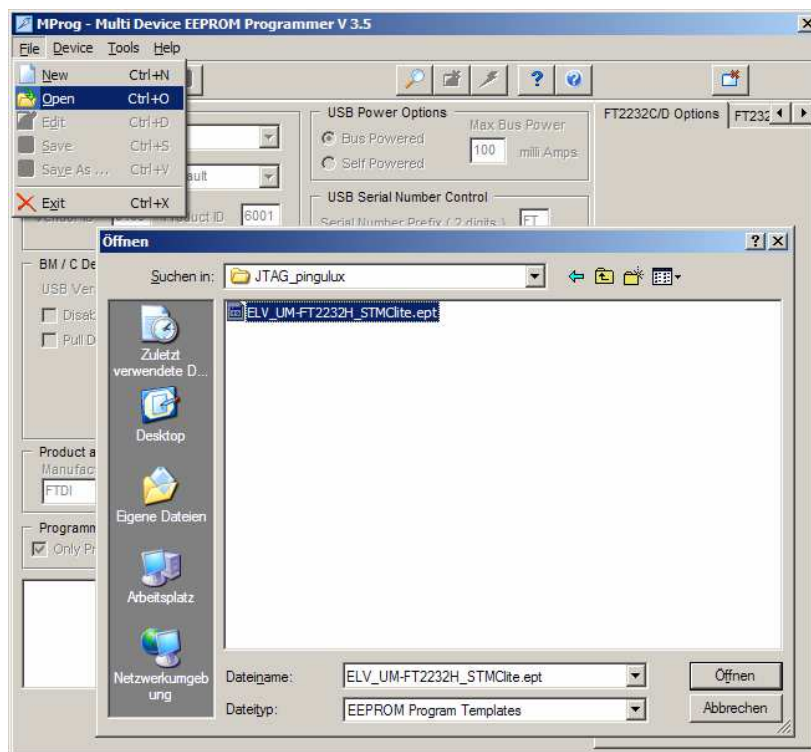
Um das „ELV Highspeed Mini USB Modul UM-FT232H“ als „ST Micro Connect Lite“ Clone verwenden zu können, muss der EEPROM des Moduls vorher entsprechend programmiert werden, damit die USB-Kennungen (VID, PID und Seriennummer usw.) mit dem übereinstimmen, was die Software von STMicroelectronics erwartet. Zum Programmieren von Adaptern mit FTDI-Chips gibt es von FTDI selbst entsprechende Tools und für den FT232H Chip bietet sich da **MProg 3.5** an.

Wenn das „ELV Highspeed Mini USB Modul UM-FT232H“ zum ersten Mal an den USB-Port eines PCs angeschlossen wird, müssen erstmal die zugehörigen Treiber von FTDI (**kann man bei FDTI immer in der aktuellsten Version downloaden und in ein beliebiges Verzeichnis entpacken**) mit Hilfe des dann automatisch startenden Hardware-Installations-Assistens installiert werden:

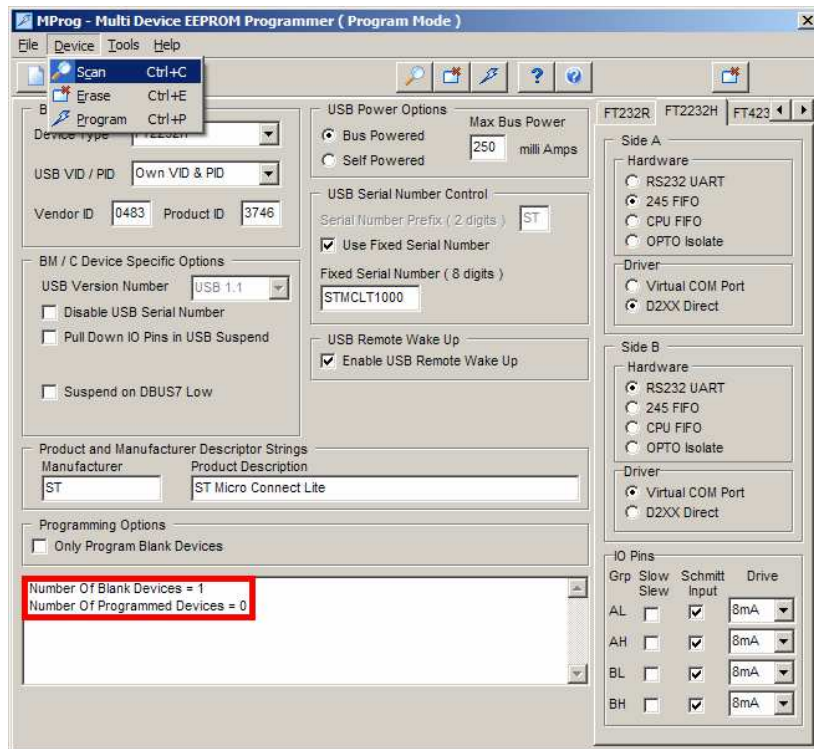




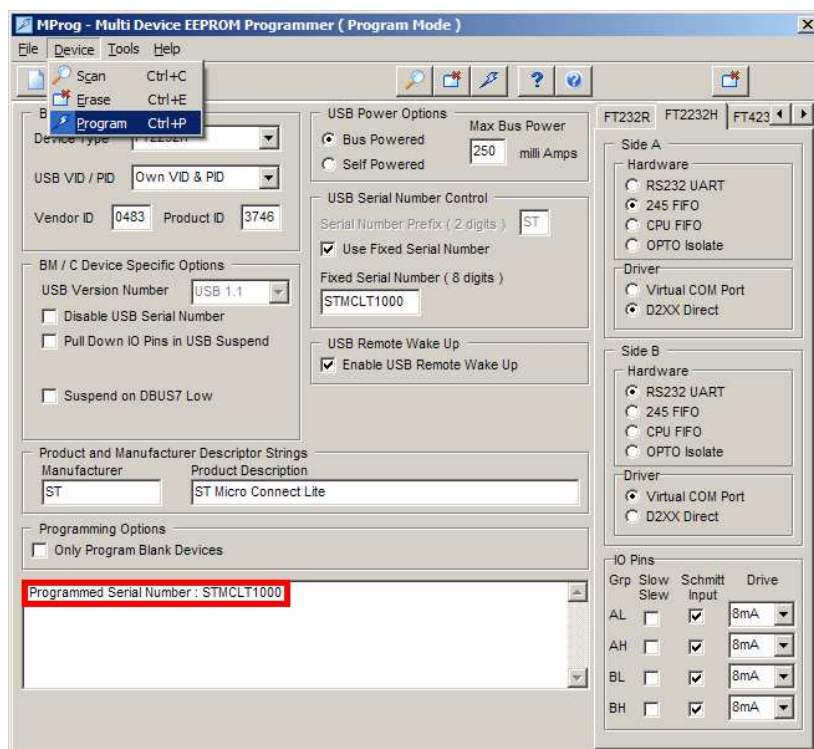
Die „MProg 3.5“ Utility kann man bei FTDI downloaden, in ein beliebiges Verzeichnis entpacken und dort dann direkt starten. Über das Menü „File“ kann man dann mit „Open“ die dieser Beschreibung beigefügte Template-Datei [ELV_UM-FT2232H_STMClite.ept](#) laden:



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:



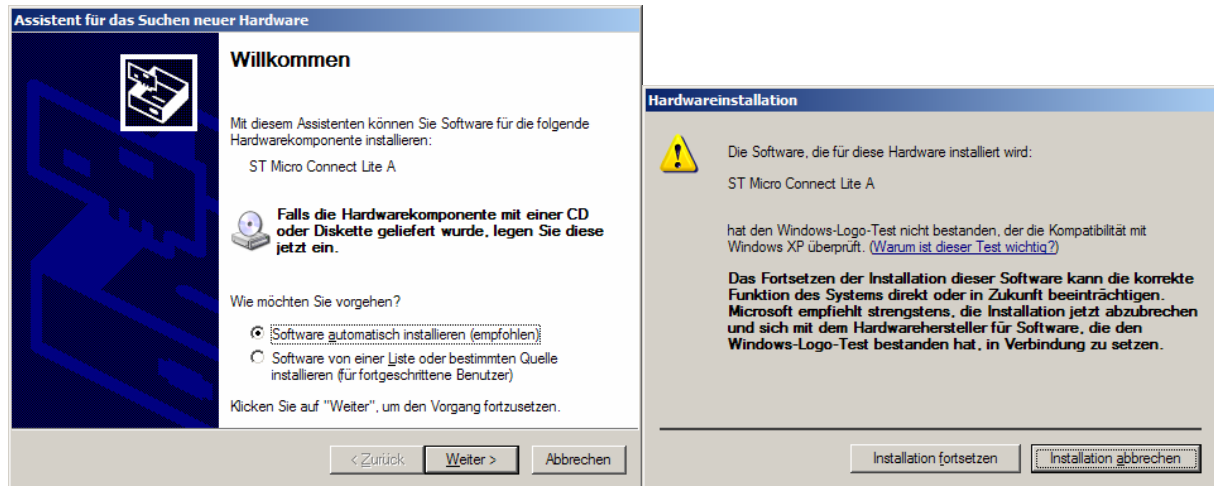
Die erfolgreiche Programmierung wird zum Schluss angezeigt:



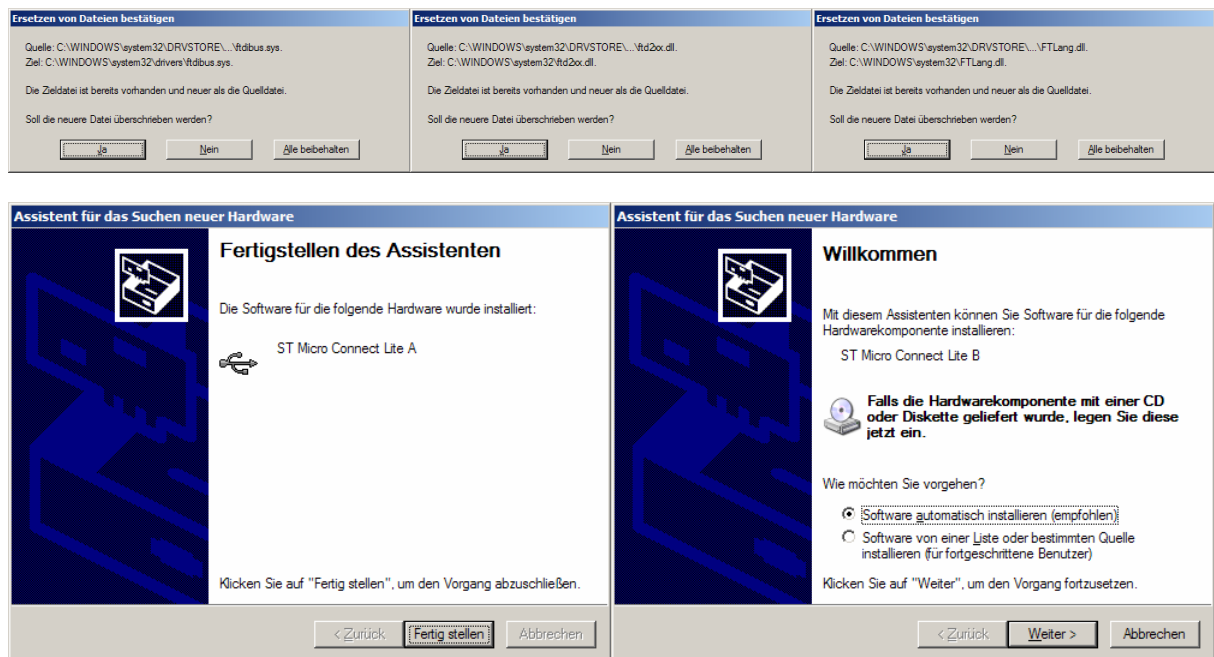
3. JTAG-Flashen des Bootloaders

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:

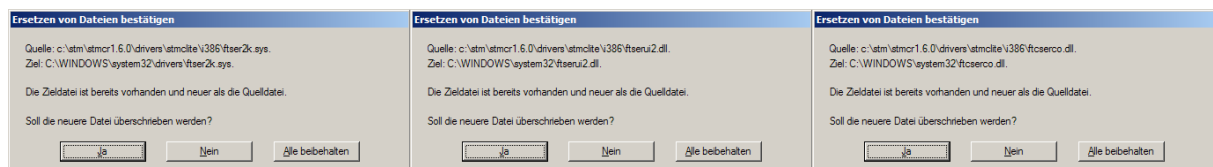
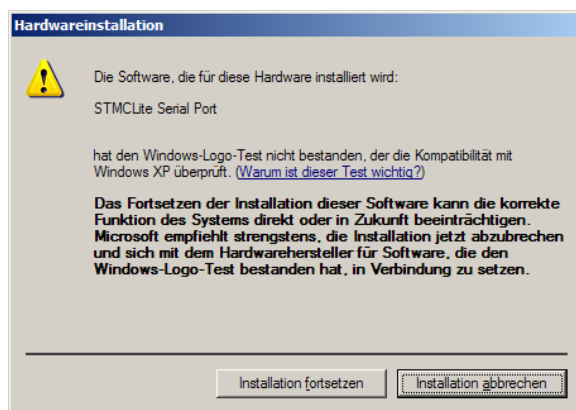
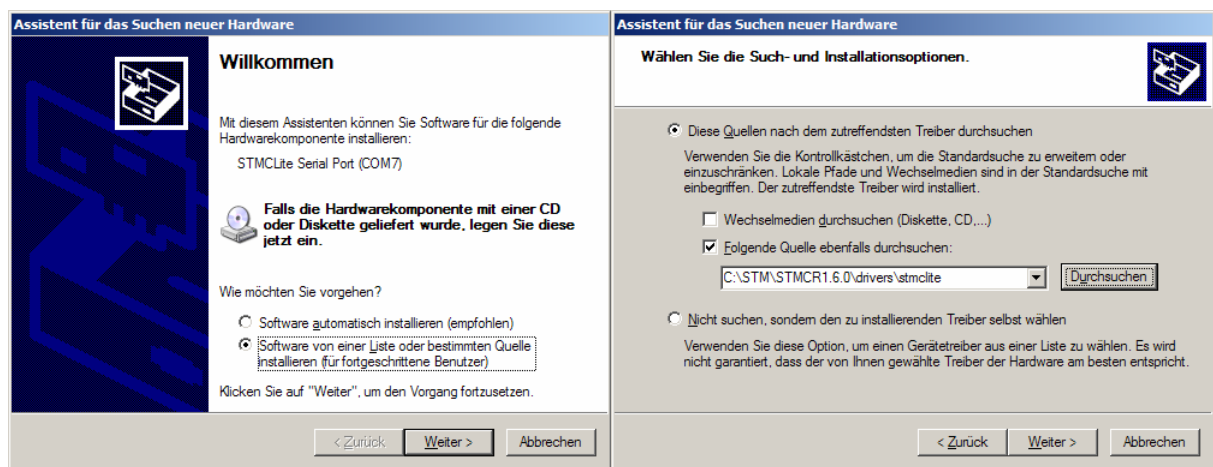


Da die vorher installierten FTDI-Treiber für das noch nicht umprogrammierten „ELV Highspeed Mini USB Modul UM-FT232H“ neueren Datums als die von STMicroelectronics sind, muss/sollte man die neueren FTDI-Dateien durch die älteren von STMicroelectronics ersetzen lassen:



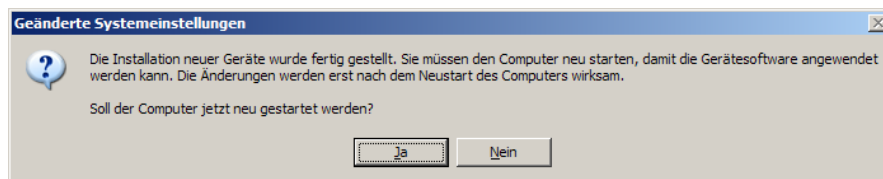


Den passenden Treiber für den virtuellen COM-Port des FT2232H Chips findet der Hardware-Installation-Assistent nicht automatisch und man muss die Treiber von STMicroelectronics, die unter **C:\STM\STMCR1.6.0\drivers\stmclite** installiert wurden, manuell vorgeben, die Installation der nicht signierten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

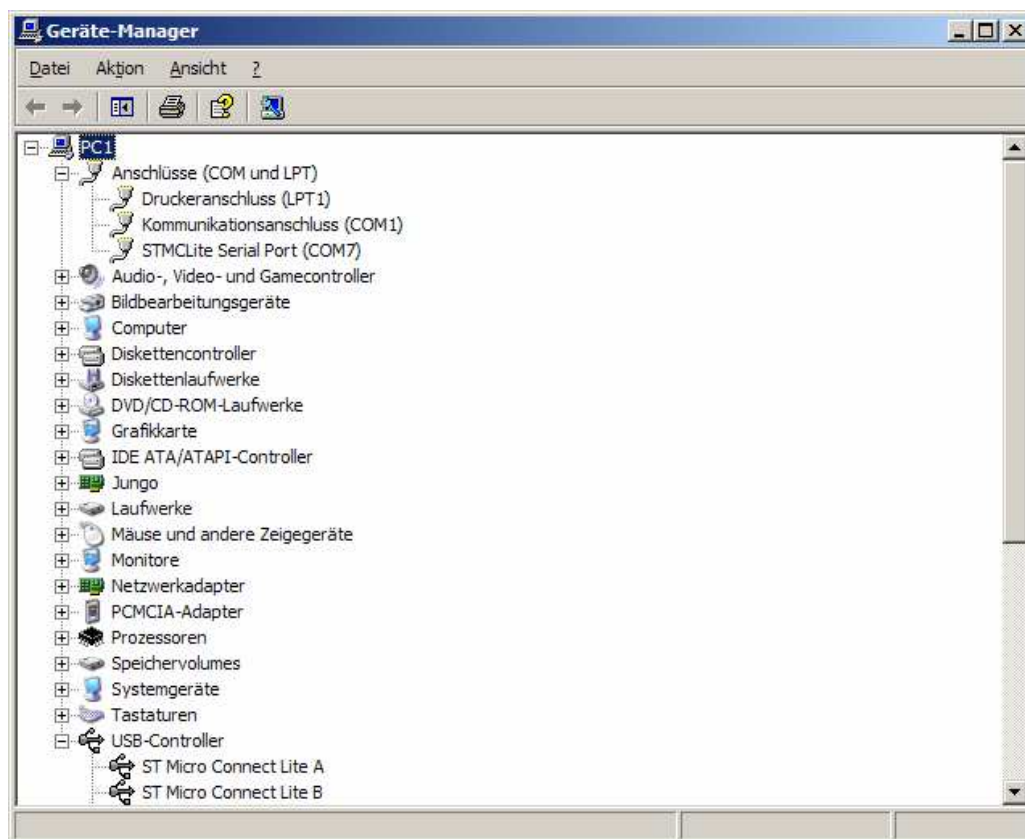




Eventuell ist ein Neustart von Windows notwendig:



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:

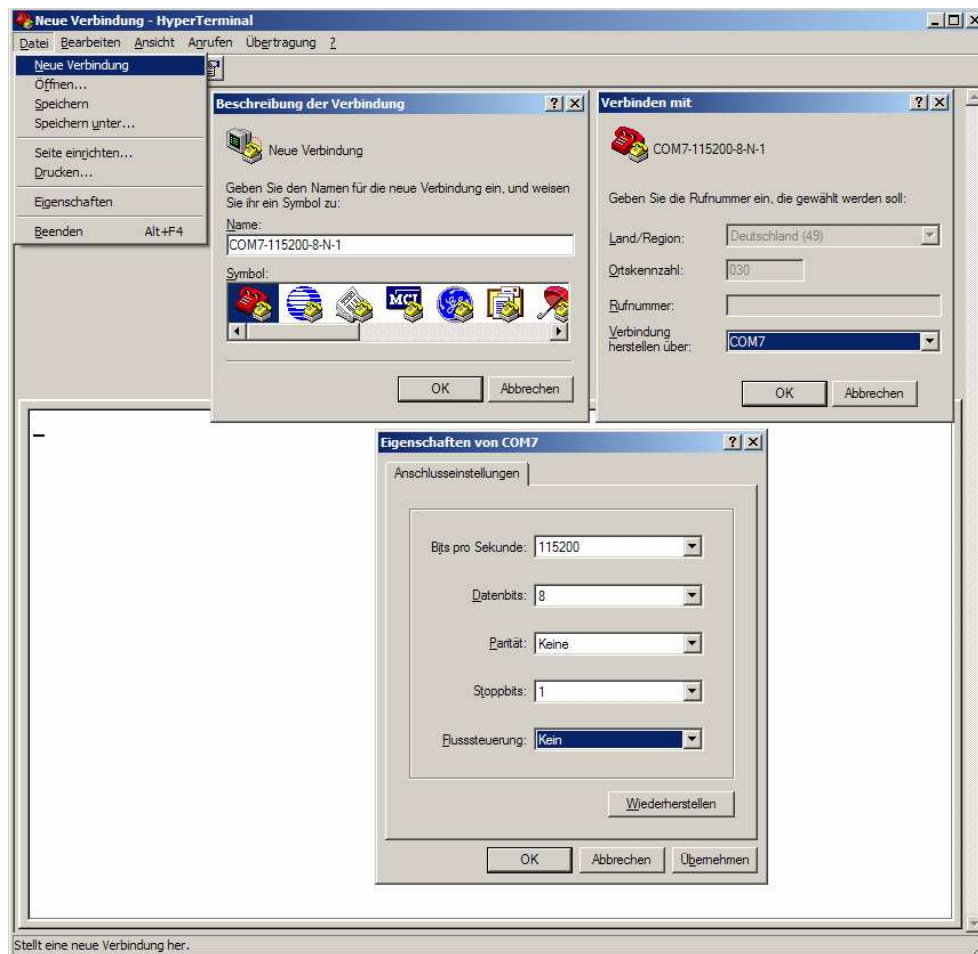


Damit sind alle Voraussetzungen zum JTAG-Flashen geschaffen.



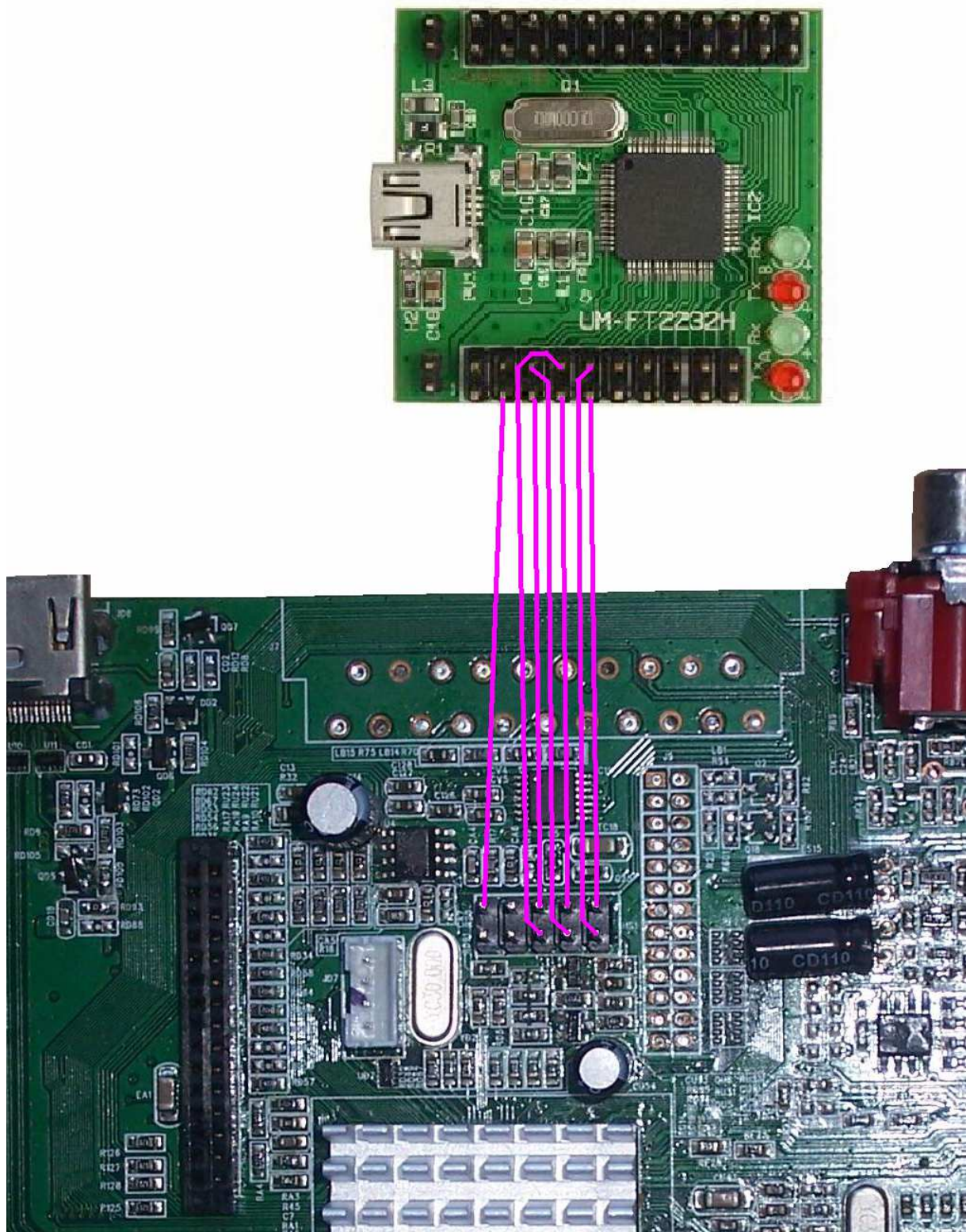
3.2 Serielle RS-232 Daten-Verbindung

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-FT232H“ realisierbar) und ein Terminal-Programm wie z.B. Hyperterminal von WindowsXP mit den Übertragungsparametern **115200 Baud**, **8 Datenbits**, **keine Parität**, **1 Stoppbit** und **keine Flusssteuerung** für den verwendeten COM-Port gestartet werden, damit man die serielle Verbindung mit dem Receiver nutzen kann:



3.3 JTAG Daten-Verbindung

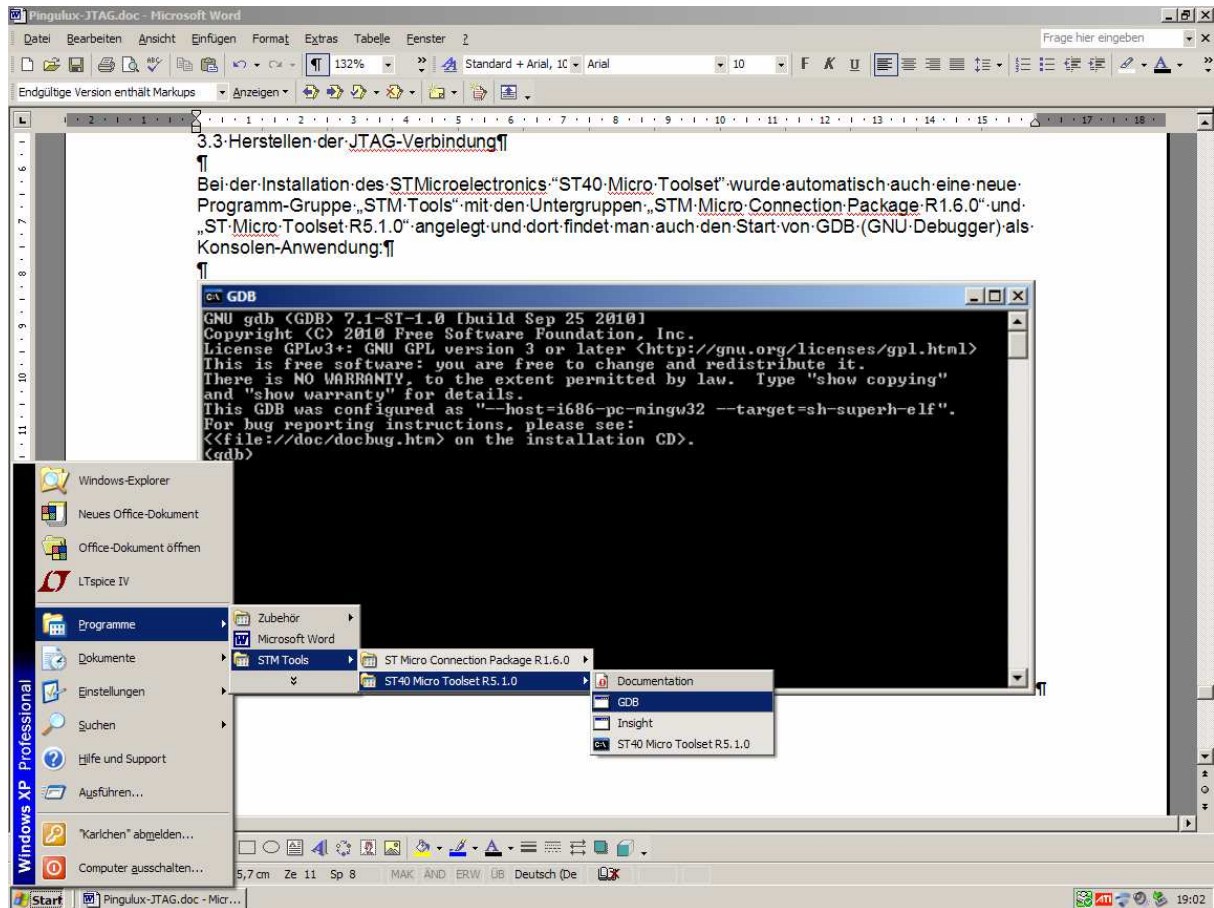
Der Receiver muss noch mit dem „ELV Highspeed Mini USB Modul UM-FT232H“ als „ST Micro Connect Lite“ JTAG-Adapter verbunden sein, wobei nur die gleichnamigen JTAG-Signale **GND**, **TCK**, **TDO**, **nTRST**, **nSRST**, **TDI** und **TMS** des JTAG-Interfaces vom Receiver nötig sind und die restlichen 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:mb618:st40,debug=2
```

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. Danach wird dann auch noch gleich ein Test des RAM-Speichers ausgeführt (die rot markierten Zeilen) und dessen Ergebnis sollte man besonders beachten:

```
GDB
(gdb) sh4tp STMCLT1000A:mb618:st40,debug=2
The target is assumed to be little endian
The target architecture is assumed to be sh4
mb618 connect start - parameters <'debug': '2'>
Initialization TCK frequency set to 3000000 Hz
Device id 0x2d43b041
tapmux connect(): boot mode single core setup
tapmux setup to bypass to core st40, channel 1
mb618 initialization start ...
mb618_setup - parameters <'debug': '2', 'tapmux_bypass_init': u'st40', 'reset_lo
w_period': 360000>
Chip infos
  Device ID = 0x2D43B041 ==> STi7111 cut 3.0
  Mode pins = 0x00002128 ==> [1:0] ClockgenA ref: SYS0_CLKIN/FE OSC
                             [10:9] Boot mode . : ST40 first
                             [17:16] Boot device: NOR flash
                             [14] Boot port size: 16-bits
                             [11] Nand page size: 512 bytes

In do_ic_wa_mode()
Clock frequencies
  CKGA_PLL0 HS/LS = 900.0/450.0 MHz
  CKGA_PLL1      = 800.0 MHz
  ST40 ICK       = 450.0 MHz
  LX DMU         = 450.0 MHz
  LX AUD         = 450.0 MHz
  FDMA0         = 400.0 MHz
  FDMA1         = 400.0 MHz
  STNOC          = 400.0 MHz
  IC_BDISP_200   = 200.0 MHz
  IC_DISP_200    = 200.0 MHz
  IC_IS_200      = 200.0 MHz
  IC_COMPO_200   = 200.0 MHz
  IC_IF_200      = 200.0 MHz
  IC_IF_100      = 100.0 MHz
  DISP_PIPE_200  = 200.0 MHz
  BLIT_PROC      = 266.7 MHz
  IC_DELTA_200   = 266.7 MHz
  ETHERNET       = 25.0 MHz
  PCI            = 66.7 MHz
  EMI MASTER     = 100.0 MHz
  LMI2X          = 800.0 MHz
Info: STA03=0x101FB4F6 => DLL1=0x7B, DLL2=0x1F6
Done poke_sysconf_cfg55: lmi16bits 0, filter 5, dummy 0

Info: DQSx=494 <0x1EE>, DQS valid offset=0x0
Info: LMI2X=800 Mhz => period=2.5ns, refresh val=2864
Info: MIM0=0xb30017b
Info: BANDWIDTH limiter ACTIVE <Frame=160, Limit=7>
Info: STR0=0xcc2db41b, STR1=0x2202d6
Info: SDRAM1_0J=0x14000a20
Info: ddr size for lmi_sdrax: 128

Info: ddr_size_func => called <function poke_lmi at 0x025DCAF0><C>
Info: EPLD IDENT=0x0
Board identification: TOO OLD EPLD !!
Testing DDR memory <size=128MB, inc=16MB>
-> writing ...
-> checking ...
epld_configure complete
Info: correcting interco reset setup
chip_init complete
stx7111: booted video companion
stx7111: booted audio companion
TCK frequency set to 10000000 Hz
tapmux complete_connect(): single core setup
mb618 initialization complete
0xa0000000 in ?? <
(gdb) -
```



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 F:\\JTAG_pingulux\\u-boot.o
```

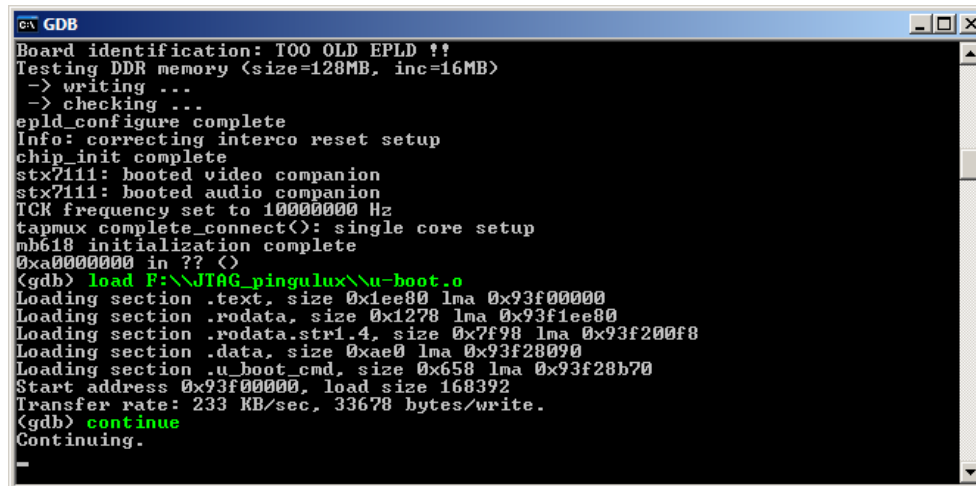
ein.

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

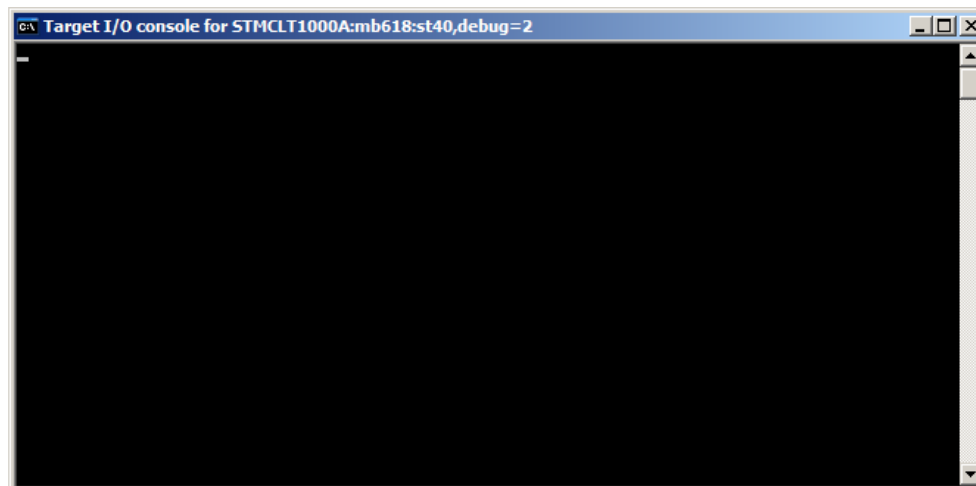
```
continue
```

starten:



```
C:\ GDB
Board identification: T00 OLD EPLD !!
Testing DDR memory (size=128MB, inc=16MB)
-> writing ...
-> checking ...
epld_configure complete
Info: correcting interco reset setup
chip_init complete
stx7111: booted video companion
stx7111: booted audio companion
ICK frequency set to 100000000 Hz
tapmux complete_connect(): single core setup
mb618 initialization complete
0xa0000000 in ?? (<)
(gdb) load F:\\JTAG_pingulux\\u-boot.o
Loading section .text, size 0x1ee80 lma 0x93f00000
Loading section .rodata, size 0x1278 lma 0x93f1ee80
Loading section .rodata.str1.4, size 0x7f98 lma 0x93f200f8
Loading section .data, size 0xae0 lma 0x93f28090
Loading section .u_boot_cmd, size 0x658 lma 0x93f28b70
Start address 0x93f00000, load size 168392
Transfer rate: 233 KB/sec, 33678 bytes/write.
(gdb) continue
Continuing.
```

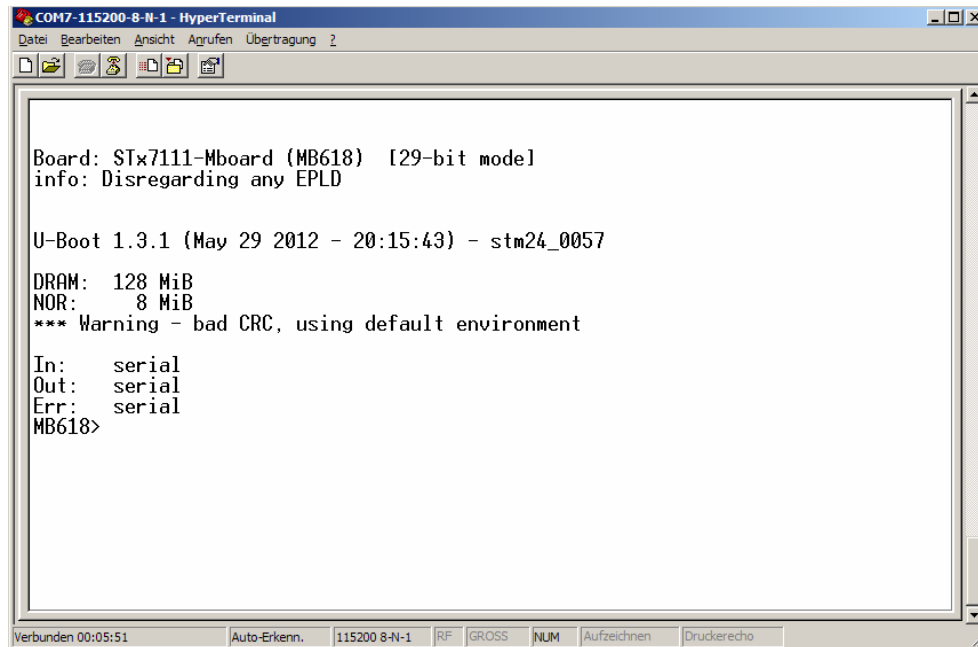
Dabei wird ein weiteres Konsolen-Fenster für eventuelle Ein-/Ausgaben des im RAM gestarteten Programms geöffnet:



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:



```
Board: STx7111-Mboard (MB618) [29-bit model]
info: Disregarding any EPLD

U-Boot 1.3.1 (May 29 2012 - 20:15:43) - stm24_0057

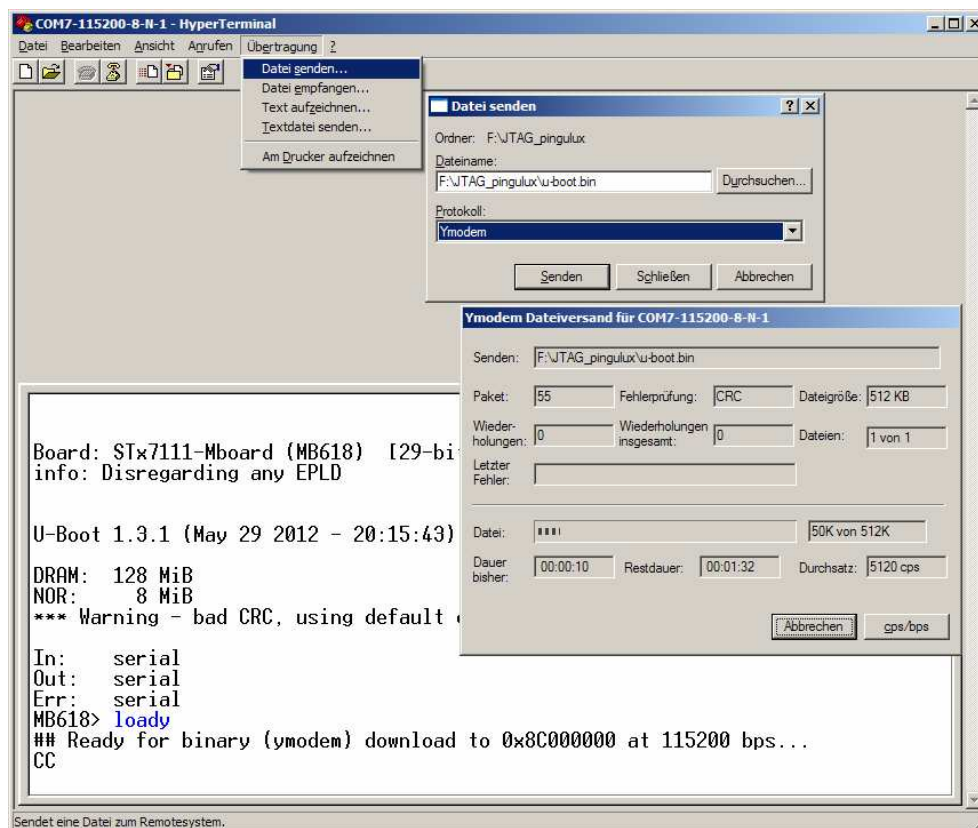
DRAM: 128 MiB
NOR: 8 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
MB618>
```

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:



```
Board: STx7111-Mboard (MB618) [29-bit model]
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
```



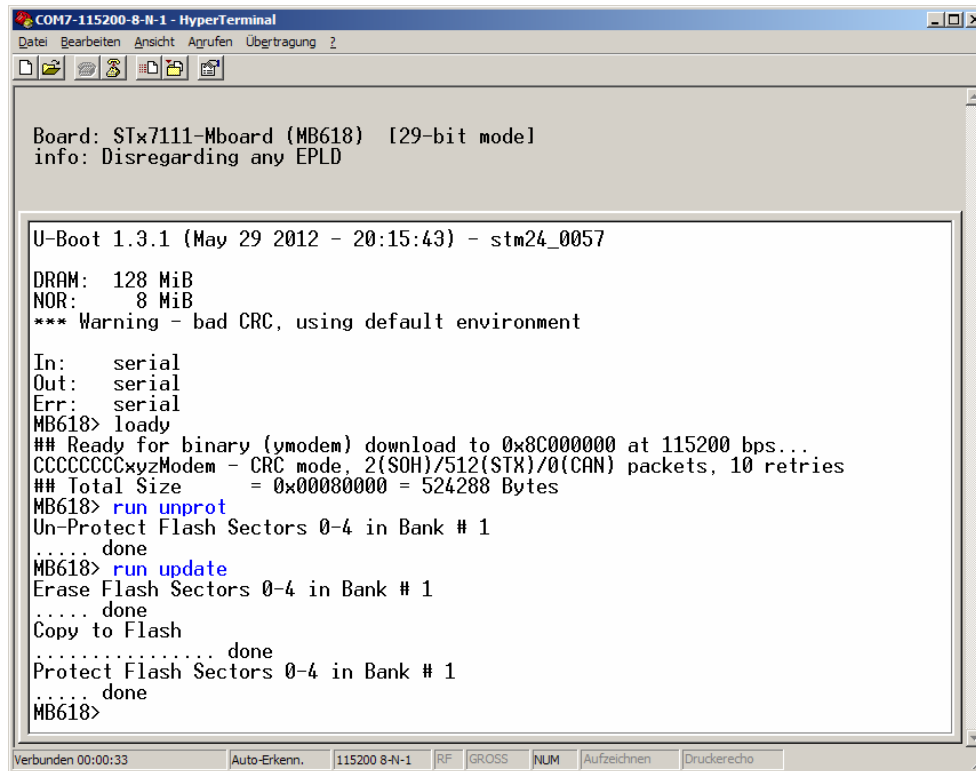
Nachdem die **u-boot.bin** Datei erfolgreich übertragen wurde muss deren Inhalt noch als Bootloader geflasht werden. Dafür enthält das rudimentäre u-boot zwei Script-Befehle, die man mit dem u-boot run-Befehl nacheinander starten muss. Im ersten Schritt wird mit:

run unprot

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

run update

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:



```
COM7-115200-8-N-1 - HyperTerminal
Datei Bearbeiten Ansicht Agrufen Übertragung 2

Board: STx7111-Mboard (MB618) [29-bit model]
info: Disregarding any EPLD

U-Boot 1.3.1 (May 29 2012 - 20:15:43) - stm24_0057

DRAM: 128 MiB
NOR: 8 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
MB618> loady
## Ready for binary (vmodem) download to 0x8C000000 at 115200 bps...
CCCCCCCxyzModem - CRC mode, 2(SOH)/512(STX)/0(CAN) packets, 10 retries
## Total Size = 0x00080000 = 524288 Bytes
MB618> run unprot
Un-Protect Flash Sectors 0-4 in Bank # 1
..... done
MB618> run update
Erase Flash Sectors 0-4 in Bank # 1
..... done
Copy to Flash
..... done
Protect Flash Sectors 0-4 in Bank # 1
..... done
MB618>
```

Fertig!

Jetzt kann man die beiden Konsolen-Fenster schließen, den Receiver aus- und wieder einschalten und sollte im Hyperterminal den normalen Bootvorgang beobachten können. In den meisten Fällen ist das Linux-System noch vollständig vorhanden und startet ganz normal. Ansonsten muss man mit Hyperterminal oder mit einem USB-Speichermedium das Linux-System komplett neu installieren.

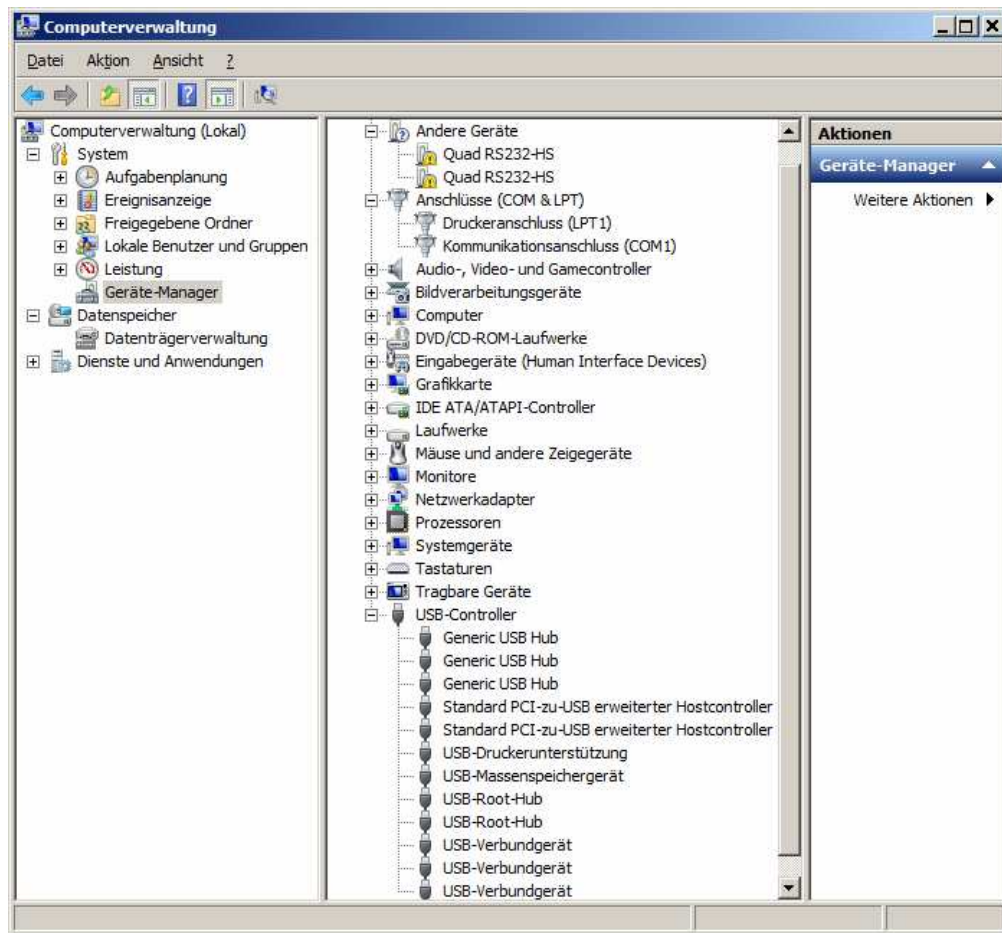
Viel Erfolg

Rimini at Nachtfalke.biz



Installation der Treiber für das USB-Modul von ELV unter Windows 7/32Bit

Wenn man das noch nicht programmierte USB-Modul das erste mal anschließt und auch noch keine Treiber von FTDI installiert wurden, Tauchen im Geräte-Manager zwei unbekannte Geräte auf:



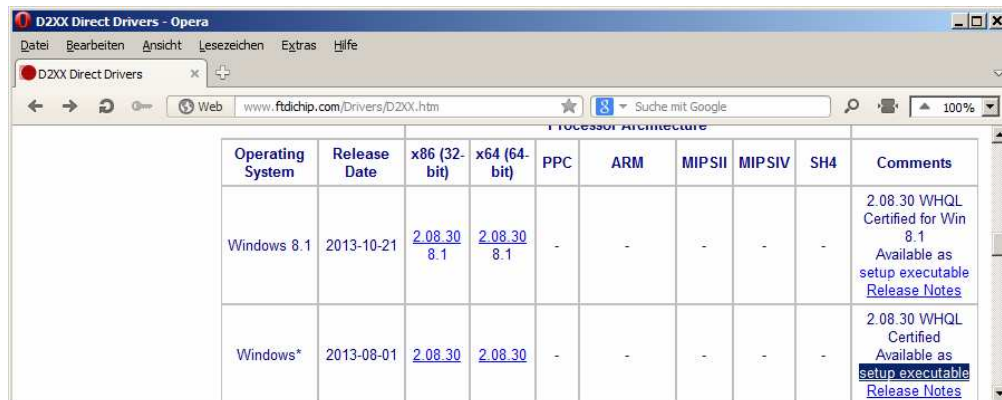
Nicht wundern, ich verwende z.Z. ein FT4232H-Modul, das vier Ports hat und das deshalb Quad RS232-HS heißt. Da der FT232H des ELV-Moduls nur zwei Ports hat, hab ich der Übersichtlichkeit die zwei zusätzlichen Ports des FT4232H im Geräte Manager gelöscht.



Download der D2XX-Treiber von FTDI:

<http://www.ftdichip.com/Drivers/D2XX.htm>

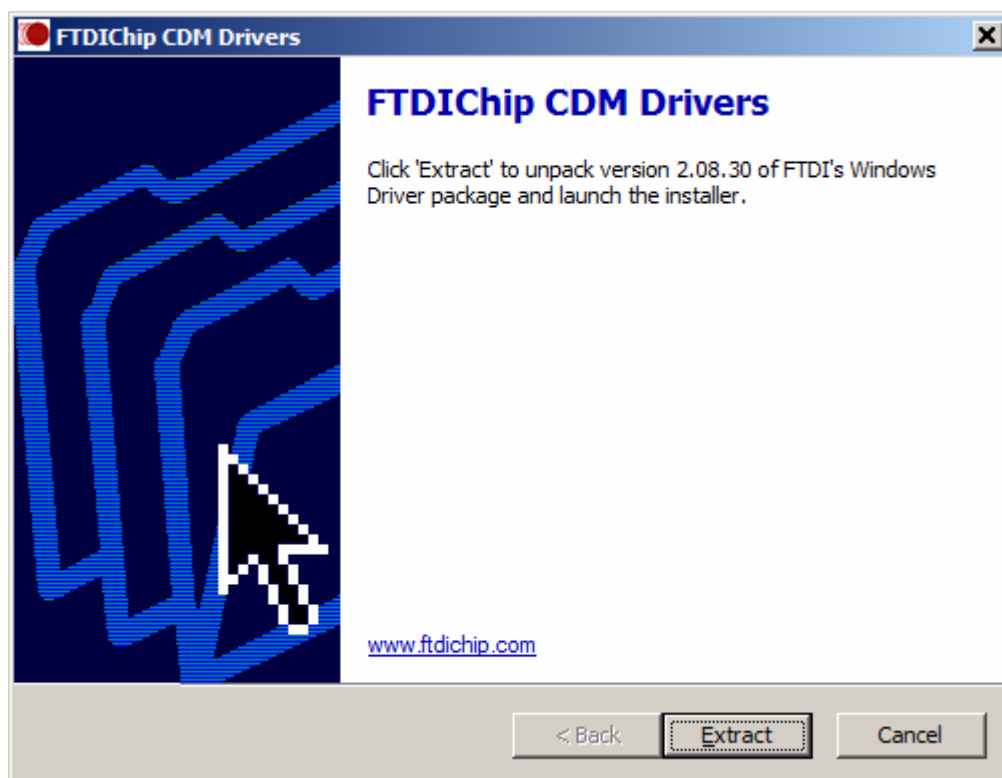
und dort findet man dann auch ein automatisches Installer-Programm:



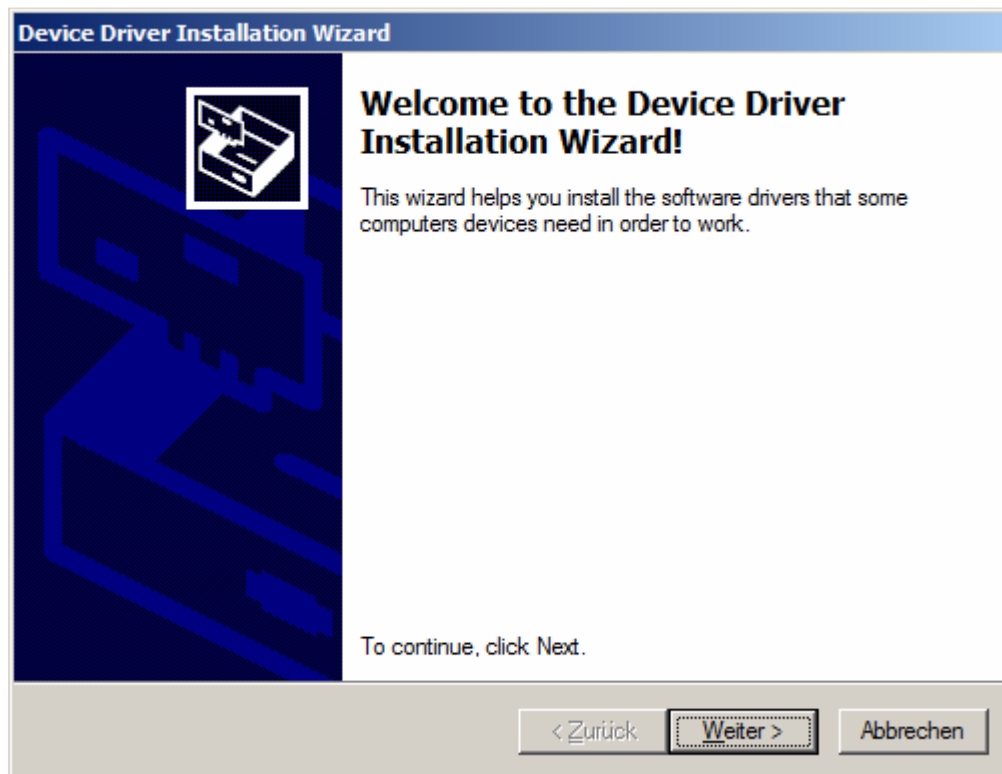
Das Installer-Programm downloaden und ausführen lassen.

Zunächst fragt die eventuell Benutzerkontensteuerung von Windows 7, ob man dieses Programm wirklich ausführen will, weil es Veränderungen am Windows-System vornimmt. Dem muss man natürlich zustimmen, sonst werden die Treiber nicht installiert.

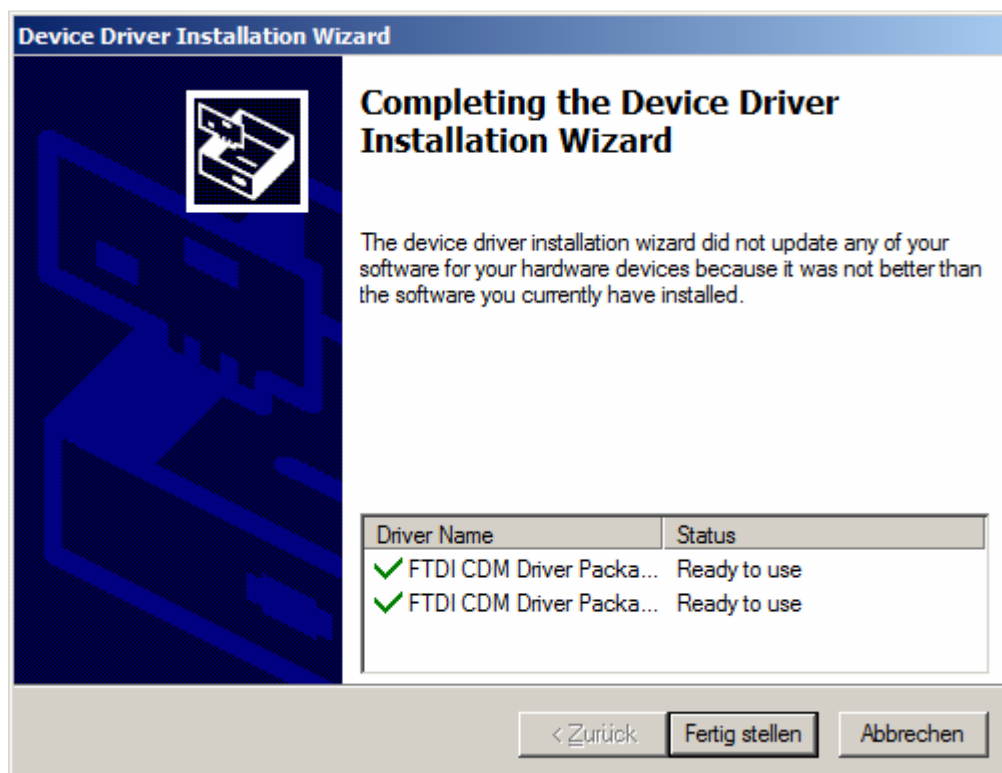
Zunächst müssen die nötigen Dateien dann automatisch extrahiert werden:



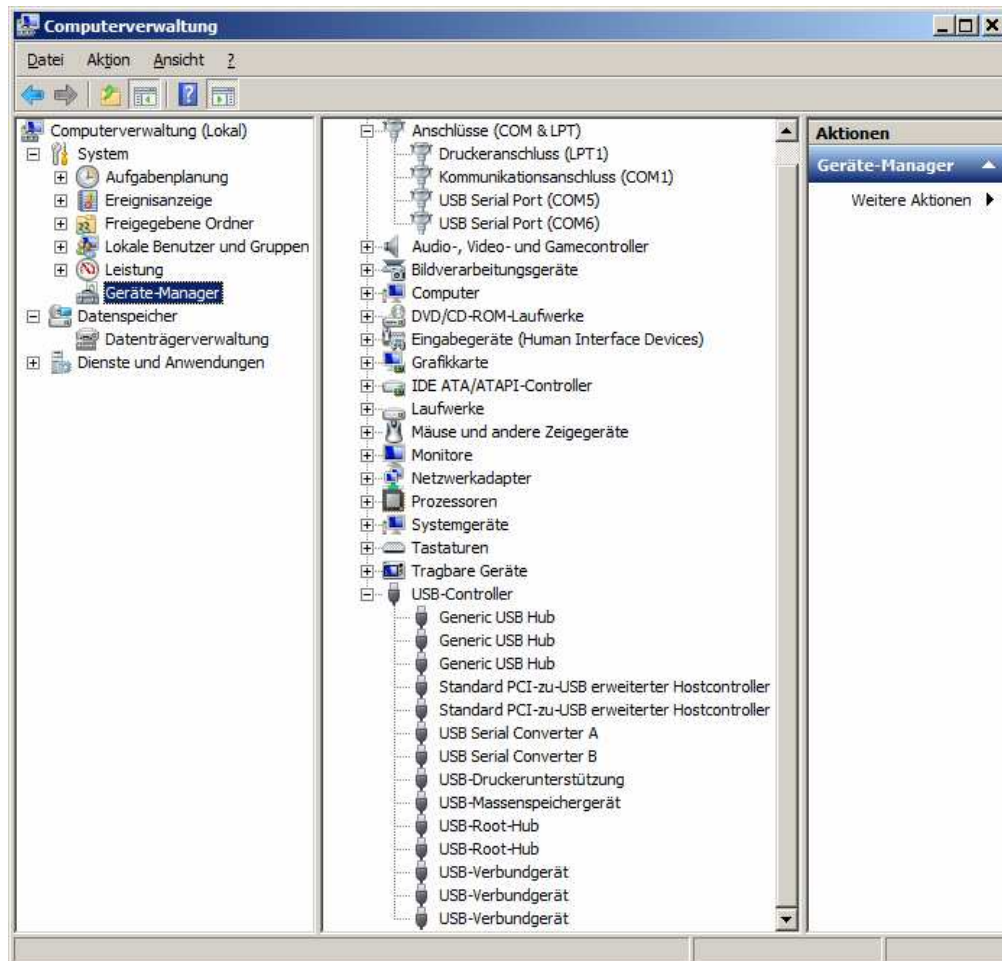
Ehe man mit der eigentlichen Treiber-Installation fortfahren kann:



Vor dem Fertigstellen der Treiber-Installation bekommt man noch angezeigt, welche Treiber installiert wurden:



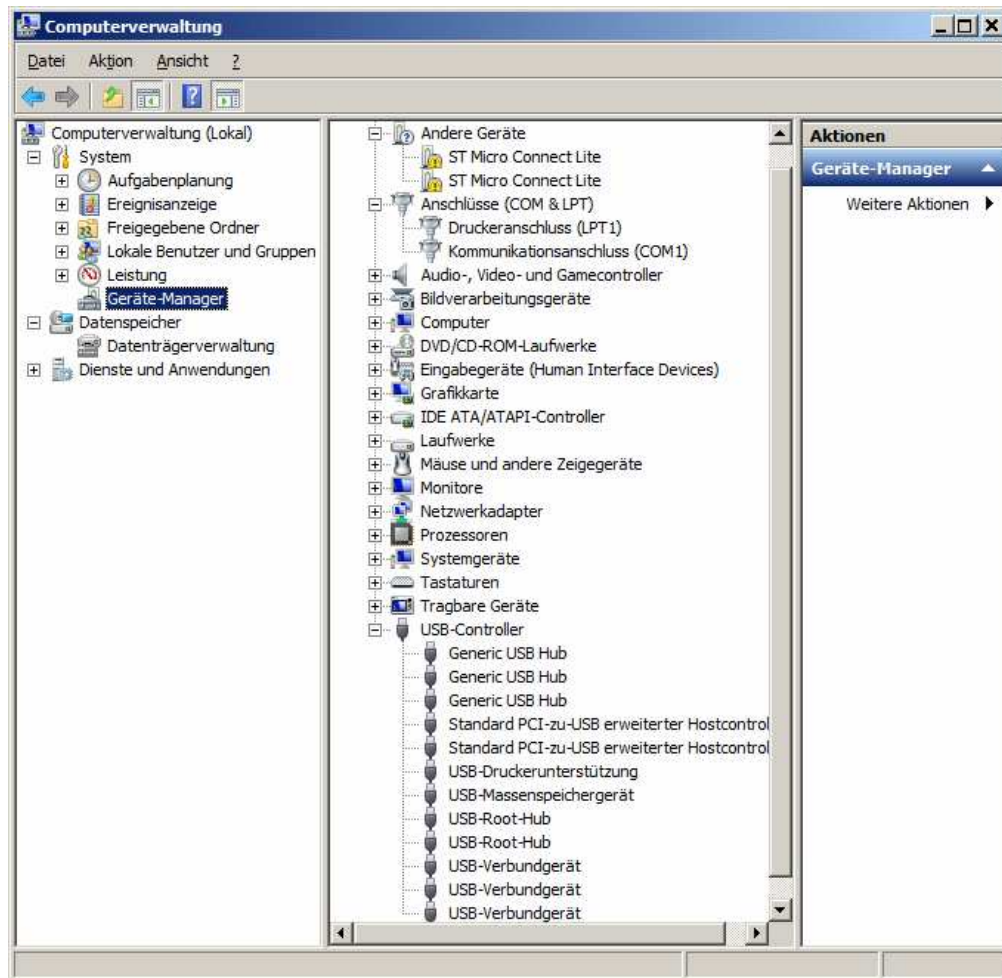
Wenn man jetzt im Geräte-Manager nachsieht, sollte das ELV UM2232H-Modul mit zwei USB Serial Ports (COMx) und als USB Serial Converter A bzw. B angezeigt werden:



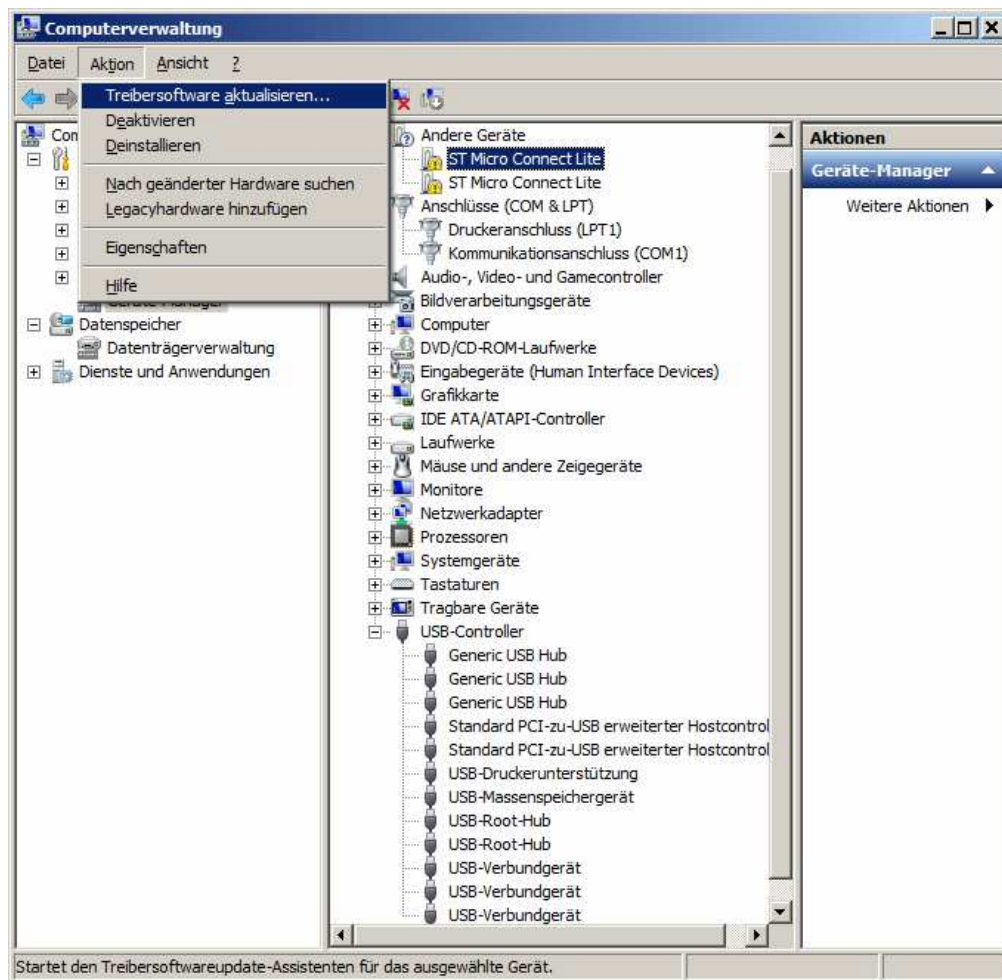
Nun kann man mit Mprog 3.5 das ELV UM-FT2232H Modul zum ST Micro Connect Lite Modul umprogrammieren.



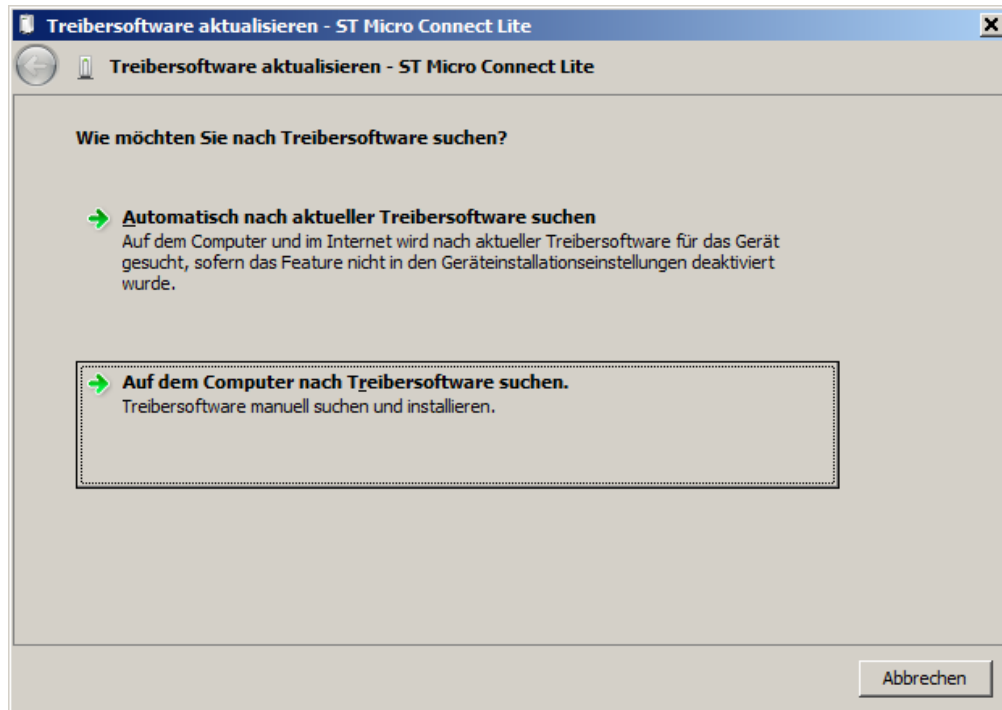
Nach der Umprogrammierung und einmal das USB-Modul entfernen und wieder anstecken, taucht es jetzt im Geräte-Manager zweimal als unbekanntes ST Micro Connect Lite Gerät auf, solange man noch nicht die Treiber von STmicroelectronic installiert hat:



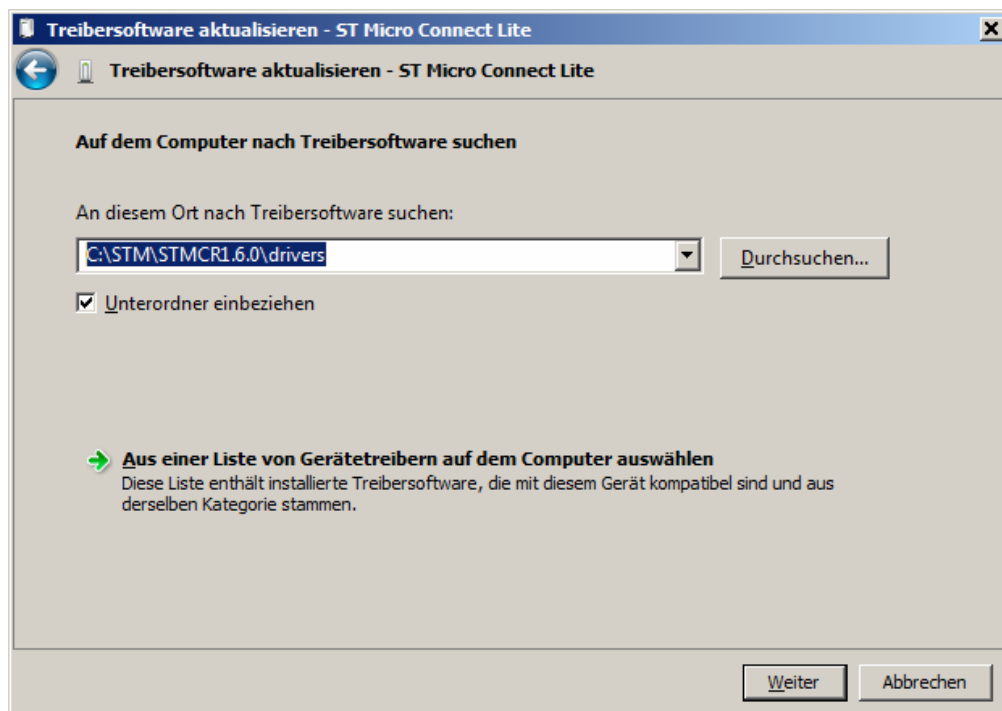
Die Treiber kann man über Aktion im Geräte-Manager installieren lassen:



Man muss die Treiber manuell suchen lassen:



und das Verzeichnis mit den Treibern (C:\STM\STMCR1.6.0\drivers) suchen und dessen Unterverzeichnisse bei der anschließenden Suche nach den passenden Treibern einbeziehen lassen:



Da es keine signierten Treiber sind, fragt die Windows-Sicherheit nach, ob man die Treiber wirklich installieren will:



Nach erfolgreicher Installation der Treiber und einmal das USB-Modul entfernen und wieder anstecken, sollte dann der STMCLite Serial Port (COMx) und die beiden ST Micro Connect Lite A bzw. B Geräte richtig erkannt angezeigt werden:

