

Amadeus

µC based USB-programmer
for Windows Computers
for selected PIC and AVR
controllers

(c) 2004 by Bernhard Michelis

Amadeus is a free project. You can freely copy and redistribute it, if no fee is charged for use, copying or distribution, with the exception, that no part of this project is allowed to be a part of a commercial product without the written permission of the author. That means not as a bonus to a book or a part of a commercial website nor anything else.

DISCLAIMER

AMADEUS IS PROVIDED TO YOU "AS IS," WITHOUT WARRANTY. THERE IS NO WARRANTY FOR THE PROGRAM, ITS DOCUMENTATION NOR ITS HARDWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM, ITS DOCUMENTATION AND ITS HARDWARE IS WITH YOU. SHOULD ANY PART OF THIS PROJECT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT THE AUTHOR WILL BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR SUSTAINED LOSSES BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAM), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

... or simplified:

“Whatever the future will bring, you are responsible. No *Risk* no *Fun!*”

Amadeus ist ein freies Projekt. Du darfst es frei kopieren und weiterverteilen, wenn weder für den Gebrauch, das kopieren noch die Verteilung eine Gebühr erhoben wird. Kein Teil des Projektes darf als kostenlose Beigabe zu einem kommerziellen Produkt verteilt werden, ohne die schriftliche Genehmigung des Autors. Diese gilt insbesondere für Bücher und CDs sowie Webseiten, auf denen Werbung gemacht wird.

HAFTUNGSAUSSCHLUSS

AMADEUS SOFTWARE, DOKUMENTATION UND HARDWAREBESCHREIBUNG WERDEN SO WIE SIE SIND ("AS IS"), OHNE JEGLICHE GARANTIE UND HAFTUNG ZUR VERFÜGUNG GESTELLT. ALLE SICH AUS DER NUTZUNG ERGEBENDEN RISIKEN OBLIGEN DEM BENUTZER. DIESES SCHLIESST VERLETZUNGEN DER RECHTE DRITTER EIN.

UNTER KEINEN UMSTÄNDEN IST DER AUTOR DES PROJEKTES FÜR SCHÄDEN JEDLICHER ART HAFTBAR. DER NUTZER VERPFLICHTET SICH, ALLE SCHÄDEN AUF EIGENEN RECHNUNG ZU BEHEBEN.

DIESES GILT AUCH DANN, WENN DER AUTOR ÜBER MISSTÄNDE INFORMIERT WURDE.

...oder einfach:

“Was die Zukunft auch bringt, Du (der Nutzer) bist schuld. Kein Risiko, Kein Vergnügen!”

Table of Contents

1 General Information/Allgemeine Informationen.....	4
1.1 Supported Devices/Unterstützte MCUs.....	5
2 Putting the programmer into operation/Inbetriebnahme.....	6
3 Amadeus Software Description/Beschreibung.....	9
3.1 Basics/Grundlagen.....	9
3.1.1 File Selection/Dateiauswahl.....	9
3.1.2 Identify-Button.....	10
3.1.3 Get Configuration From-Button.....	10
3.1.4 Program, Read, Verify.....	10
3.1.5 RAM/EEPROM Read Reset.....	11
3.1.6 Connection and Power Switching/Schalten der Versorgungsspannung.....	11
3.2 Debugging Features/Fehlersuch Möglichkeiten.....	12
3.2.1 How Do I Use this feature/Was fang ich damit an.....	12
3.2.2 Suggestion/Empfehlung.....	13
3.3 ATtiny 11/12.....	14
3.3.1 How To Connect/Programmierschnittstelle.....	14
3.3.2 Programming/Programmieren.....	14
3.3.3 Additionlly Features/Zusätzliches.....	15
3.4 ATmega 8/16/32/64/128.....	15
3.4.1 How To Connect/Programmierschnittstelle.....	15
3.4.2 Programming/Programmierung.....	16
3.4.3 Factory Reset / Reanimate Function.....	18
3.4.4 Oscillator Calibration Byte/Kalibriungsbytes.....	18
3.4.5 EEPROM Programming/Programmierung.....	19
3.5 PIC12F 629/675 / PIC16F 630/676.....	19
3.5.1 How To Connect/Programmierschnittstelle.....	20
3.5.2 Programming/Programmierung.....	20
3.5.3 Calibration Recovery/Kalibrationswiederherstellung.....	21
3.6 PIC18F 242/248/252/258/442/448/452/458.....	22
3.6.1 How To Connect/Programmierschnittstelle.....	22
3.6.2 Programming/Programmierung.....	23
3.6.3 Mini Prommer Window.....	24
4 Amadeus Programmer Hardware.....	25
4.1 Electrical Characteristics/Elektrische Eigenschaften.....	25
4.1.1 Serial Programming Interface.....	25
4.2 Schematics.....	27
4.3 Partlist.....	28
4.4 Assembly.....	30
4.5 Layout.....	31

1 General Information/Allgemeine Informationen

This project is based on a Microchip PIC18F242/252 microcontroller and Future Technology's FT232BM USB-to-Serial converter.

Currently, the software provides four programming algorithms for selected PIC and AVR controllers.

Dieses Projekt basiert auf einem PIC18F242/252 Mikrokontroller und dem FT232BM USB-Serial-Wandler von Future Technologys.

Derzeit gibt es vier Programmieralgorithmen für verschiedene PIC und AVR Mikrokontroller.

1.1 Supported Devices/Unterstützte MCUs

μC	Information
ATtiny11	Not Tested - High-voltage serial programming
ATtiny12	Tested - High-voltage serial programming only
ATmega48	Not Tested*
ATmega8	Not Tested*
ATmega88	Not Tested*
ATmega8515	Not Tested
ATmega8535	Not Tested
ATmega16	Not Tested
ATmega162	Not Tested
ATmega168	Not Tested*
ATmega169	Not Tested
ATmega32	Not Tested
ATmega325	Not Tested*
ATmega3250	Not Tested*
ATmega329	Not Tested*
ATmega3290	Not Tested*
ATmega64	Tested
ATmega645	Not Tested*
ATmega6450	Not Tested*
ATmega649	Not Tested*
ATmega6490	Not Tested*
ATmega128	Not Tested
PIC12F629	Not Tested
PIC12F675	Tested
PIC16F630	Not Tested
PIC16F676	Not Tested
PIC18F242	Tested
PIC18F248	Not Tested
PIC18F252	Tested
PIC18F258	Not Tested
PIC18F442	Test
PIC18F448	Tested
PIC18F452	Tested
PIC18F458	Not Tested

Table 1: List of Supported MCUs

* Have a look at the accompanying “ATmega_Setup” file.

* Bitte auch die Begleitdatei “ATmega-Setup” beachten.

2 Putting the programmer into operation/Inbetriebnahme

This chapter describes what to do, to make Amadeus work. Maybe it is a good idea to glance over the entire document at first.

Dieses Kapitel beschreibt, wie die Programmer Hardware in Betrieb genommen wird.

First of all, you need to order all necessary components. Find a list of all required components in the partlist on page 28.

Als erstes sollten alle notwendigen Komponenten bestellt werden. Eine Liste gibt es auf Seite 28.

Next you need to produce your own PCB. There's an external file, named "Amadeus_PCB.gif" (single side layout), with a higher resolution than the graphics in this document. The schematics can be found on page 26. Feel free to create your own layout, if you like. But be aware that at least the USB controller is only available as SMD package.

Als nächstes gilt es, sich eine Leiterplatte zu fertigen. Die Begleitdatei "Amadeus_PCB.gif" enthält ein Layout in hoher Auflösung. Den Schaltplan findest Du auf auf Seite 26, sowie den Bestückungsplan auf Seite 29.

Then you have to assemble your PCB. See chapter "Assembly" on page 29 for details.

When the hardware is ready to work you have to do some programming.

Nachdem die Hardware aufgebaut ist, muß die Schaltung installiert und programmiert werden.

Download two file from the internet. First of all the D2XX driver package for the FT232BM (www.ftdichip.com/Drivers/FT232-FT245Drivers.htm) and a tool named MProg (www.ftdichip.com/Resources/Utilities.htm#MProg). Both can be found on the webside of Future Technology's.

Wir brauchen den D2XX-Treiber für den FT232BM (www.ftdichip.com/Drivers/FT232-FT245Drivers.htm), sowie ein Programmierool mit dem Namen MProg.exe (www.ftdichip.com/Resources/Utilities.htm#MProg). Beides gibt es auf der Seite von Future Technologys.

Connect the programmer hardware to your computer. Windows should find a new USB device. Install only the D2XX driver not the VCP driver.

Verbinde die Programmer Hardware mit Deinem Computer. Windows sollte ein neues USB-Gerät finden. Installiere nur den D2XX-Treiber und nicht den VCP-Treiber.

After rebooting, start **MProg**. Load the programming package "Amadeus.ept" provided with this documentation and program it into the USB-EEPROM.

*Nach dem Neustart **MProg** starten. Lade die Begleitdatei "Amadeus.ept" in MProg und programmiere sie in die Schaltung.*

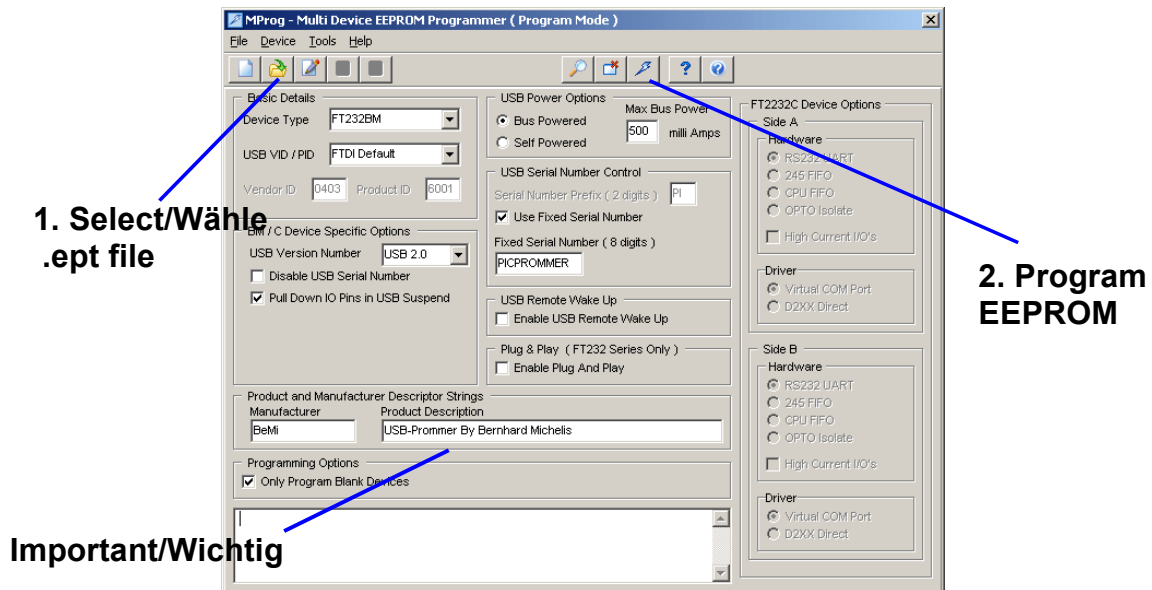


Illustration 1: MProg UI

Disconnect the programmer from the computer and reconnect it. Probably Windows will find a new device, because we reprogrammed the USB-Controller. If you are requested to install a driver, install the **D2XX** driver again.

Den Programmer nach dem Programmieren vom Computer trennen und neu einstecken. Wahrscheinlich wird Windows jetzt ein neues USB-Device finden, da der USB-Chip umprogrammiert wurde.

Now it's time to start **Amadeus**, the programming interface software provided with this documentation. When the message box below appears, the programmer cannot be found. If the EEPROM programming worked before, the EEPROM values might not be programmed correctly. See above, the **Product Description** must be written exactly this way.

*Jetzt ist es Zeit Amadeus, die Benutzersoftware zur Hardware, zu starten. Sollte die folgende Meldung erscheinen, ist wahrscheinlich etwas bei der EEPROM Programmierung schief gegangen. Darauf achten, daß das Feld **Product Description** genauso, wie auf dem Bild, geschrieben ist.*

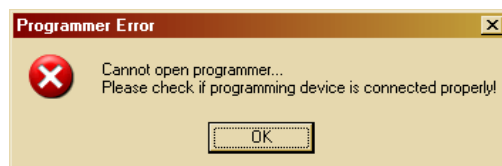


Illustration 2: Error Message if no hardware was found

If everything works correctly, you get the following Message, telling you, that the MCU is not programmed.

Nachdem der USB-Controller nun korrekt programmiert ist, erscheint folgende Meldung, die uns sagt, daß die MCU im Programmer noch keine Firmware hat.

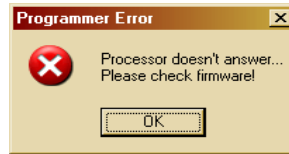


Illustration 3: Firmware is missing in the programmer

The last step, before you can use your new programmer is to program the firmware. Therefore go to the PIC18 programming interface.

Der letzte Schritt, bevor Du den Programmer einsetzen kannst, ist die Programmierung der Firmware.

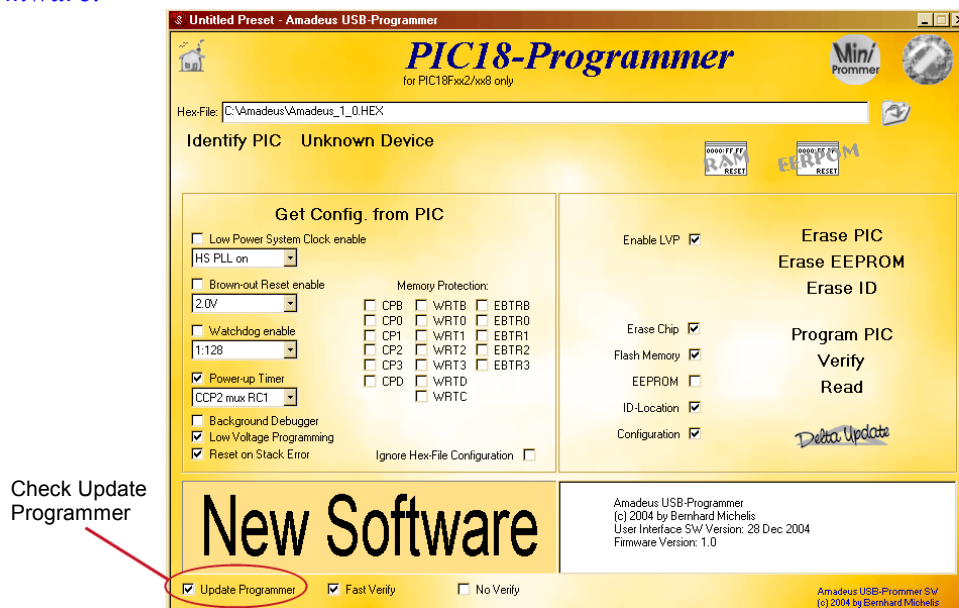


Illustration 4: PIC18 interface is also for firmware updates

Select the firmware file *Amadeus.hex* and check **Update Programmer**. Everything, except the path, should look like on the picture. Click on **Identify PIC**. If the programming interface is working correctly, the software realizes the PIC18F242 (or 252). When this is not working, check your soldering.

Wähle die Firmware-Datei Amadeus.hex und setze das Häkchen bei Update Programmer. Klick auf Identify PIC. Wenn alles in Ordnung ist, sollte ein PIC18F242 (oder 252) erkannt werden. Wenn nicht, bitte den Aufbau nochmal kontrollieren.

Now click **Program PIC**. It will take a little time, then the programmer should display **SUCCESS**.

Mit einem Klick auf Program PIC wird das Software-Update gestartet. Es dauert eine Zeit und dann sollte SUCCESS erscheinen.

If the controller was realized, but you got a FAIL, don't be sad and try again. This can happen with some USB interfaces and has no effect on the later stability of the programmer. If it fails again have a look at the log window. In case, only Flash Memory was FAIL, click **Delta Update**, once, twice,.... Otherwise remove the **Erase Chip** checkmark and uncheck all successful steps. Try again **Program PIC**. *If Erase Chip is unchecked, it is possible to program each part of the firmware seperately.*

The problem with this firmware update is, that the FT232BM has no flow control in *BitBang* mode. Sending data always works fine, but verifying might be a problem. If it won't work at all, check **No Verify**. Now the verification will be skipped. When the hardware is working correctly, this is ok.

*Wenn der Controller erkannt wurde, aber trotzdem ein FAIL erscheint, hat das nichts zu bedeuten. Das kann bei einigen USB-Schnittstellen der Fall sein, hat jedoch keinen Einfluß auf die Zuverlässigkeit im späteren Betrieb. Einfach nochmal **Program PIC** anklicken. Sollte es auch wiederholt scheitern, bitte einmal einen Blick auf das Log-Fenster werfen. Ist nur Flash-Programming fehlgeschlagen (fail), dann so oft auf **Delta-Update** klicken, bis PASS erscheint.*

*Anderenfalls alles deselektieren, was **successful** war. In diesem Fall unbedingt auch das **Erase Chip** Häkchen entfernen, und erneut **Program PIC**. Solange wiederholen, bis alles Pass ist.*

*Das Problem mit dem Firmware-Update resultiert aus der fehlenden Flußkontrolle beim FT232BM im BitBang Modus. Das Programmieren funktioniert eigentlich immer einwandfrei, nur das Vergleichen scheitert. Sollte also garnichts helfen, kann ein Häkchen bei **No Verify** gesetzt werden. Jetzt wird kein Vergleich nach dem Schreiben mehr durchgeführt.*

The last step is to disconnect the programmer from the USB port and close Amadeus.

Nun die Programmer Hardware vom USB-Port entfernen.

Drumroll..... Trommelwirbel.....

Connect the hardware again and start Amadeus. This time the firmware must be read correctly.

Und wieder anschließen. Dieses mal muß die Firmware korrekt ausgelesen werden.

That's it! Now have fun and enjoy fast software updates. No more waiting...

So, das war's! Viel Spaß und genießt die schnellen Software-Updates. Nie wieder langes Warten.

3 Amadeus Software Description/Beschreibung

3.1 Basics/Grundlagen

First of all, select the desired microcontroller family. There's a dedicated interface for each type of microcontroller. Despite of differences, there are some constant elements, identical for all MCU types supported by Amadeus. To shorten this documentation, they are summarised in the following paragraphs. Slight differences might exist between several implementations.

Als erstes wähle bitte die Mikrocontroller-Familie aus. Jede Familie hat ihre eigenen Benutzeroberfläche. Von wenigen Elementen abgesehen gibt es viele Gemeinsamkeiten, die in den folgenden Abschnitten beschreiben werden.

3.1.1 File Selection/Dateiauswahl

Select the hex-file to be programmed. Therefore click the folder symbol right beside the hex-filename edit field and select a file. Files must be in *Intel-Hex* format.

Wähle die zu programmierende Hex-Datei. Klicke auf den Ordner rechts neben dem Dateieingabefeld. Alle Dateien müssen im Intel-Hex Format vorliegen



Illustration 5: File Selection

For *Microchip* devices, all information (program, EEPROM, ID and configuration) can be contained in one hex-file.

Für PIC-Controller können alle Information (Programm, EEPROM, ID und Konfiguration-daten) in einer Hex-Datei zusammengefasst sein.

For *Atmel* devices, normally the hex-file only contains the program. Therefore there are two folder buttons on the right side. Click the left one to select the program file and the right one, if EEPROM values have to be programmed. You can deselect the EEPROM file by canceling the EEPROM-file selection.

Bei Atmel enthält die Hex-Datei nur das Programm. Deswegen gibt es eine weitere Ordner-Schaltfläche rechts neben der Dateiauswahl-Schaltfläche für den Inhalt des EEPROMs. Abwählen kann man die EEPROM Datei, wenn man das Auswahlfenster öffnet und die Auswahl dann abbricht.

3.1.2 Identify-Button

Click to query MCU information. When the MCU is supported by the selected interface and the connection for in-circuit serial programming is correct, the MCU's name will be displayed.

*Ein Klick auf die **Identify** Schaltflächen zeigt die angeschlossene MCU an, sofern sie von der gewählten Benutzeroberfläche unterstützt wird. Bei einigen MCUs werden erst nach erfolgreicher Erkennung die Konfigurationdaten angezeigt.*

3.1.3 Get Configuration From-Button

Reads the current configuration settings from the connected MCU and displays them.

Lieft die aktuellen Konfigurationdaten aus der MCU und zeigt sie an.

3.1.4 Program, Read, Verify

Programs, reads or verifies the MCU. Depending on the MCU, there are some settings available, to give more detailed control.

Die drei Schaltflächen Program (programmieren), Read (auslesen) und Verify (vergleichen) lösen die entsprechenden Aktionen aus.

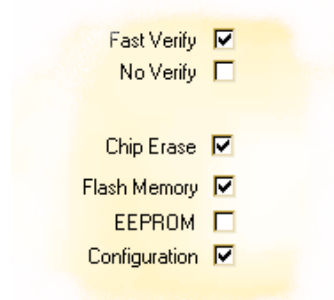


Illustration 6: Action Selection

Fast Verify speeds up verification by skipping all unprogrammed areas. To test, if unprogrammed areas are really empty (0xFF), disable **Fast Verify**. This is also valid for **Read!** In case of read, only memory areas are read, that are defined by the selected hex-file.

*Wenn **Fast Verify** angewählt ist, werden beim Vergleichen alle Bereiche übersprungen, die in der Hex-Datei nicht definiert sind. Es findet also keine Prüfung auf 0xFF statt. Beim Lesen sollte diese Option inaktiv sein.*

No Verify suppresses all read operations. In this case the verification result is always *successful*.

*Mit **No Verify** wird jeglicher Vergleich unterdrückt. Es wird immer **SUCCESS (Erfolg)** angezeigt.*

Erase Chip, Software, Flash Memory, EEPROM, ID-Location and **Configuration** enable or disable the corresponding functionality for programming, verification or reading.

*Die Optionen **Erase Chip, Software, Flash Memory, EEPROM, ID-Location** und **Configuration** aktivieren oder unterdrücken die entsprechende Aktion.*

Ignore Hex-File Configuration (PIC only) uses the displayed configuration settings instead of the settings stored in the hex-file.

*Wenn **Ignore Hex-File Configuration** (nur PIC) abgehakt ist, werden eventuelle Konfigurationsdaten aus der Hex-Datei ignoriert und stattdessen die angezeigten Daten programmiert/verglichen.*

Some additional functionality might be available for some MCUs.

Je nach MCU kann es noch weitere Funktionen geben.

3.1.5 RAM/EEPROM Read Reset

This is a little debugging feature. As far as possible it reads RAM (PIC18 only) or EEPROM content via the programming interface and displays them as hex dump. In all cases a reset accompanies this request.

Diese ist ein kleines Debugging-Feature. Es liest RAM- (nur PIC18) oder EEPROM-Inhalte aus und zeigt diese als Hex-Dump an. Das ganze wird von einem Reset begleitet.

3.1.6 Connection and Power Switching/Schalten der Versorgungsspannung

Because the programming interface goes into high-impedance mode (see Electrical Characteristics/Elektrische Eigenschaften on page 24) after programming, in most cases there's no need to disconnect the programmer from your target system after programming. That rises the wish to reset the whole target system after programming, because some external components might got upset during programming.

Da die Programmierschnittstelle nach dem Programmieren hochohmig wird (siehe auch Seite 24), ist es im allgemeinen unnötig den Programmer zu entfernen, damit die Zielschaltung normal arbeiten kann. Hier ist es dann natürlich interessant, die gesamte Schaltung Zurückzusetzen, da ja einige Teile durch die Programmierung durcheinandergeraten sein können.

Therefore the programmer can switch an external power supply. See Electrical Characteristics/Elektrische Eigenschaften on page 24 for details. In case that you need to switch voltages above about 20V you should design your own power switch and use the internal Power Switch (N4 and N5) or EXTVCC signal to control it. EXTVCC is somewhere between 4.6V and 5.2V during programming and can be used as power supply during programming for a stand alone programming interface (MCU, oscillator only).

Dafür bietet der Programmer die Möglichkeit die Spannungsversorgung des Zielsystem aus- und wieder einzuschalten. Man sollte allerdings darauf verzichten Spannungen von mehr als 20V mit dem Relais zu schalten. Zum Schalten von Netzspannungen muß Du dir eine eigene Schaltung entwerfen, die dann vom Relais oder EXTVCC angesteuert werden kann.

EXTVCC liefert während der Programmierung ca. 4,6-5,2V und ist als Spannungsversorgung für Standalone Programmiergeräte gedacht.

Caution: Don't use the relay for high voltage switching.

Achtung: Relais auf keinen Fall zum Schalten hoher Spannungen verwenden.

Power switching can be activated on the MCU selection view. You have the choice between inactive, manual and two versions for some predefined times.

Power switching is evoked, if active, every time the programmer accesses the target MCU. You can choose *power on* or *off* after every MCU access.

*Das Schalten der Spannungsversorgung mittels Relais (**Power Switching**) kann auf der MCU Auswahlseite eingestellt werden.*

*Neben **kein Schalten** (inactive), **Handbetrieb** (manual) gibt es eine Reihe vordefinierter Zeiten. Hier kann man dann auch entscheiden, ob nach dem Programmieren die Spannung ausgeschaltet (off) oder aus und wieder eingeschaltet werden soll (on).*

3.2 Debugging Features/Fehlersuch Möglichkeiten

Amadeus is *not* an in-circuit-debugger. However it offers at least one simple debugging feature.

Amadeus ist kein In-Circuit-Debugger. Trotzdem bietet es eine einfache Möglichkeit der Fehlersuche im System.

The EEPROM-Read-Reset give you the opportunity to display data as hex-dump. So it is possible to read back data from you MCU, even if no other communication way is implemented. Just write the desired data to the EEPROM. This can be done by a macro, that you can develop and test inside the simulation environment of MP-Lab or AVR-Studio.

Der EEPROM-Read-Reset gibt Dir die Möglichkeit ein Hex-Dump des EEPROMs abzurufen. Selbst wenn noch kein anderer Weg zur PC – Mikrokontroller kommunikation eingerichtet ist, kann man so Informationen aus dem Mikrokontroller zurücklesen. Schreibe die interessanten Daten einfach ins EEPROM. Ein Macro hierfür kannst Du in MP-Lab oder AVR-Studio entwickeln und testen.

Feel lucky, if you are using a PIC18. Amadeus can read PIC18 RAM directly, so no EEPROM write macros are necessary for debugging.

Sei froh, wenn Du einen PIC18 verwendest, denn Amadeus kann das RAM des PIC18 direkt lesen. Das Aufbereiten der Daten für das EEPROM fehlt.

In all cases, a reset accompanies the reading.

Alle Leseaktionen werden von einem Reset begleitet.

3.2.1 How Do I Use this feature/Was fang ich damit an

Imagine, you are trying to program the UART interface and nothing will work. So maybe you will ask yourself sometime, did the byte arrive? Okay, you can switch an LED on or off, or do something else, when the RX-IRQ occurs. But you never know, if the byte was received correctly. Think of the mistakes, you could make. Wrong baudrat, wrong parity...

Stell Dir vor, Du versuchst einen UART zu programmieren und nichts funktioniert. Vielleicht fragst Du dich irgendwann, "Kommen meine Daten überhaupt an?". Natürlich könnte man in der RX-ISR eine LED anschalten, wenn ein Byte empfangen wird, aber ob das Byte richtig ankam, erfährst Du so nicht. Denke an falsche Baudrate, falsche Parität...

In such cases, just copy the received byte to EEPROM/RAM and *loop forever*. Now use the Read-Reset to see the byte. In all cases the complete EEPROM/RAM will be hex-dumped.

In so einem Fall kopiere das empfangene Byte ins EEPROM/RAM gefolgt von einer Endlosschleife. Jetzt verwende die EEPROM/RAM-Read-Funktion.

It might happen, that the MCU starts executing some instructions before the memory can be read out. Insert some NOPs at the beginning.

Es kann vorkommen, daß einige MCUs bereits einige Befehle ausführen, bevor der Speicher gelesen werden kann. In solchen Fällen am Anfang es Programmes ein paar NOPs einfügen.

The other way to debug your software in-circuit step by step is to:

- Use endless loops instead of breakpoints.
- Copy all interested information to EEPROM/RAM before the endless loop.
- Perform the Read-Reset.
- Set the copy macros and the endless loop to the next *breakpoint* and reassemble, reflash and restart your code.
- And so on...

Special Function Registers can also be copied.

Der andere Weg Deine Software in der Schaltung zu Debuggen ist:

- *Verwende Endlosschleifen anstelle von Breakpoints.*
- *Kopiere alle interessanten Daten vor der Endlosschleife ins EEPROM/RAM.*
- *Führe den Read-Reset aus.*
- *Setze die Kopiermakros und die Endlosschleife zum nächsten interessanten Punkt, reassembliere und programmiere Deinen Schaltung neu.*
- *Und so weiter...*

3.2.2 Suggestion/Empfehlung

TIP: If the MCU choice is your's and saving some hours or days is worth investing additional 4€, take a PIC18. It's RAM-Read-Reset makes in-circuit debugging much easier than using the EEPROM variation.

- On every read, all variables are displayed.
- When you fill the complete RAM with 0xFF after reset, you can see, if there's a function wildly spreading data all over the memory.
- You don't need to copied interesting data to the EEPROM, only SFRs.

Tip: Wenn die Entscheidung, welche MCU verwendet werden soll, bei Dir liegt und 4€ nicht weh tun, nimm einen PIC18, da der RAM-Read-Reset einige Vorteile bietet.

- *Bei jedem lesen werden alle Variablen angezeigt (Hex-Dump)*
- *Wenn beim Programmstart der komplette Speicher mit 0xFF gefüllt wird, kannst Du leicht sehen, wenn eine Funktion wild im Speicher rumschreibt und alles durcheinanderbringt.*
- *Es brauchen keine Daten ins EEPROM kopiert werden, abgesehen von SFRs.*

3.3 ATtiny 11/12

When programming one of these devices, the programmer works in high-voltage mode, only. The low-voltage serial programming for ATtiny12 is not supported.

Both MCUs offer 1kByte program memory (512 instructions). Additionally, the ATtiny12 offers 64Byte EEPROM.

Der Programmer arbeitet bei diesem Typ immer im Hochvolt-Modus.

3.3.1 How To Connect/Programmierschnittstelle

For electrical details see chapter “Electrical Characteristics/Elektrische Eigenschaften“ on page 24.

Für elektrische Details bitte im Kapitel “Electrical Characteristics/Elektrische Eigenschaften“ auf Seite 24 nachschlagen.

6 lines are required for programming. *6 Leitungen werden benötigt.*

<i>Amadeus side</i>	<i>MCU side</i>	<i>comment</i>
n.c.	VCC	For in-circuit programming , the MCU gets its power from the target circuit. <i>Beim Programmieren in der Schaltung nicht anschließen.</i>
EXTVCC !!!	VCC	For stand alone programming only!!! Do never connect EXTVCC for in-circuit programming. <i>Spannungsversorgung für Standalone Programmiergeräte. Nie mit einer externen Spannung verbinden.</i>
GND	GND	Connecting one GND line might be enough. Connect both, if you are using longer programming cables. <i>Masseverbindung.</i>
MCLR/RST	PB5	Reset (about 12V during programming) Reset kann während der Programmierung auf 12V gehen.
PGM/PDI	PB0	Serial Data Input/ <i>Serieller Dateneingang</i>
PGD/PDO	PB1	Serial Instruction Input/ <i>Serieller Befehlseingang</i>
PGC/SCK	PB2	Serial Data Output/ <i>Serieller Datenausgang</i>
SCL	PB3	Serial Clock Input/ <i>Takteingang</i>

Table 2: ATtiny programming adapter definition

3.3.2 Programming/Programmieren

Have a look at the chapter “Basics” for details on the programming procedure.

Die Grundlagen findest Du im Kapitel “Basics/Grundlagen“ auf Seite 8.

3.3.3 Additionally Features/Zusätzliches

Atmel stores an oscillator calibration byte inside the MCU, that cannot be reached by software. Therefore Amadeus gives you the opportunity to copy this calibration byte either to the end of flash memory (high byte) or to the beginning or end of EEPROM (ATtiny12 only).

Es ist möglich das Kalibrierungsbyte entweder ans Ende des Flash oder am Anfang/Ende es EEPROMS zu positionieren (nur ATtiny12).

3.4 ATmega 8/16/32/64/128

See chapter “Supported Devices/Unterstützte MCUs“ on page 4 for a list of supported devices. On the contrary to other MCUs, ATmegas need a valid clock to be programmed. The internal 1MHz/8MHz oscillator is enabled by default. See documentaion of your MCU.

Eine Liste aller unterstützter MCUs findest Du im Kapitel “Supported Devices/Unterstützte MCUs“ auf Seite 4.

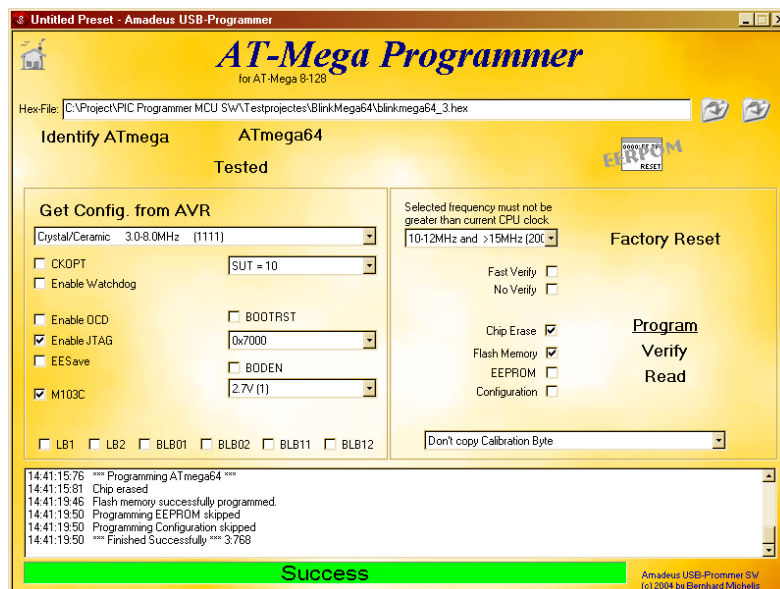


Illustration 7: ATmega Programming Interface

3.4.1 How To Connect/Programmierschnittstelle

For electrical details see chapter “Electrical Characteristics/Elektrische Eigenschaften“ on page 24.

Für elektrische Details bitte im Kapitel “Electrical Characteristics/Elektrische Eigenschaften“ auf Seite 24 nachschlagen.

5 lines are required for programming. *5 Leitungen werden benötigt.*

<i>Amadeus side</i>	<i>MCU side</i>	<i>comment</i>
n.c.	VCC	For in-circuit programming , the MCU gets its power from the target circuit. <i>Beim Programmieren in der Schaltung nicht anschließen.</i>
EXTVCC !!!	VCC	For stand alone programming only!!! Do never connect EXTVCC for in-circuit programming. <i>Spannungsversorgung für Standalone Programmiergeräte. Nie mit einer externen Spannung verbinden.</i>
GND	GND	Connecting one GND line might be enough. Connect both, if you are using longer programming cables. <i>Masseverbindung.</i>
MCLR/RST	RESET	Only low voltage programming. <i>Nur Niedervolt Programmierung.</i>
PGM/PDI	PDI/PE0 (ATmega64) / MOSI	MOSI (different ATmegas seem to use different pins for programming. Have a look at their datasheets, chapter “Memory Programming/SPI Serial Programming Pin Mapping”) <i>MOSI: Dateneingang</i>
PGD/PDO	PDO/PE1 (ATmega64) / MISO	MISO (different ATmegas seem to use different pins for programming. Have a look at their datasheets, chapter “Memory Programming/SPI Serial Programming Pin Mapping”) <i>MISO: Datenausgang</i>
PGC/SCK	SCK/PB1	SCK (different ATmegas seem to use different pins for programming. Have a look at their datasheets, chapter “Memory Programming/SPI Serial Programming Pin Mapping”) <i>SCK: Takt</i>

Table 3: ATmega programming adapter definition

<p>Be aware, that different MCU types of the ATmega series use different pins for programming.</p> <p><i>Beachte, daß verschiedene MCUs unterschiedliche Pins zur Programmierung verwenden.</i></p>
--

3.4.2 Programming/Programmierung

Only additional information on programming are given in the paragraph. Have a look at the chapter “Basics/Grundlagen” on page 8 for the basic programming procedure.

Die Grundlagen findest Du im Kapittel “Basics/Grundlagen“ auf Seite 8.

Before you start programming a device you should set up the programming speed correctly. Currently you can choose from 6 speed options.

Vor dem Programmieren muß die richtige Programmiergeschwindigkeit gewählt werden.

<i>Option</i>	<i>Description</i>
10000kHz Normal (2ms)	Select this option if you clock your MCU below 1MHz and internal oscillator does not offer internal 8MHz clock. <i>Wähle diese Einstellung, wenn die MCU mit weniger als 1MHz getaktet ist, und der interne Oszillator keine 8MHz bietet.</i> PT64: 20.6s

<i>Option</i>	<i>Description</i>
1000kHz Fast (2ms)	Select this option if you clock your MCU below 1MHz and the internal oscillator offers 8MHz. <i>Wähle diese Einstellung, wenn die MCU mit weniger als 1MHz getaktet ist, und der interne Oszillator 8MHz bietet.</i> PT64: 5.9s
1MHz Normal (2µs)	Select this, when MCU is clocked at 1MHz or faster. Use only when the MCU's oscillator does not offer internal 8MHz clock. <i>Wähle diese Einstellung, wenn die MCU mit wenigstens 1MHz getaktet ist, und der interne Oszillator keine 8MHz bietet.</i> PT64: 19.8s
1MHz Fast (2µs)	Select this, when MCU is clocked at 1MHz or faster. (Default) <i>Wähle diese Einstellung, wenn die MCU mit wenigstens 1MHz getaktet ist, und der interne Oszillator 8MHz bietet.</i> PT64: 4.3s
6,7MHz (300ns)	Select, when MCU is running above 6,7MHz. I.e. the internal oscillator@8MHz or a crystal, or ... <i>Wähle diese Einstellung, wenn die MCU mit wenigstens 6,7MHz getaktet ist.</i> PT64: 4.1s
10-12MHz and >15MHz (200ns)	High speed programming mode. Requires 10-12MHz or at least 15MHz. The range from 12 to 15 is not specified. <i>Wähle diese Einstellung, wenn die MCU mit 10-12MHz oder mehr als 15MHz getaktet ist.</i> PT64: 3.8s

Table 4: Programming Speeds

PT64 indicates the time that is needed to program the complete Flash memory of an ATmega64 with 0x00 and verify it. An ATmega128 will need about twice as long. Smaller Flash memories will be programmed faster.

If any Lock Fuses are set, don't use fast Mode. If the MCU clock is below 1MHz click Factory Reset first.

PT64 gibt die Programmierzeit für das 64k Flash des Mega64 an, wenn es komplett mit 0x00 beschrieben und verglichen wird.

Wichtig: Wenn Lock Fuses gesetzt sind, sollte man auf den **Fast-Mode** verzichten. Und bei einem MCU Takt von weniger als 1MHz vorher einen **Factory Reset** durchführen.

- During development it is recommended to select **Fast Verify**.
- If you don't need to update the EEPROM select the *EESave* fuse and disable *EEPROM* for programming. Then the EEPROM will not be erased by *Erase Chip* and reprogrammed. And programming is faster.
- *Während der Entwicklung sollte immer Fast Verify aktiv sein.*
- *Wenn das EEPROM nicht gelöscht werden muß, sollte die EESave Fuse gesetzt werden.*

If your program is smaller than the MCU's Flash memory, empty areas will be skipped. This speeds up programming further.

Wenn das Programm den Flashspeicher nicht ausfüllt, werden leere Bereiche übersprungen.

Warning

There are some ATmega MCUs, with a Reset Disable (RSTDISBL) option. Be aware, that it is theoretically possible to write this fuse (for instance by accident, or if you select the wrong programming speed!!!). After that, serial programming is no longer possible. The software does not explicitly offer the possibility to program this fuse, but if something goes wrong during programming, only a parallel programmer is able to reanimate the MCU.

Warnung

Einige MCUs haben eine RSTDISBL Fuse. Amadeus bietet zwar keine Möglichkeit diese explizit zu setzen, aber wenn ein falsches Timing (zu schnell) gewählt wird, kann es doch passieren. Also bei solchen MCUs unbedingt auf die richtige Programmiergeschwindigkeit achten, da ansonsten die MCU nicht mehr über SPI angesprochen werden kann. Hier hilft dann nur noch ein paralleler Hochvolt-Programmer.

3.4.3 Factory Reset / Reanimate Function

A fault during programming can be, that the wrong clock source was selected. In such a case, the programmer can no longer communication with the MCU.

This can be handle with the help of the Factory Reset function. To reanimate the MCU, a special clock frequency might be necessary. Therefore connect the programmers SCL pin to the MCU's XTAL1 pin and click **Factory Reset**. When Factory Reset was successful, disconnect the external clock (SCL).

*Ein Fehler, der immer wieder gemacht wird, ist, daß die falschen CKSRC Fuse Bits gesetzt werden. Hier schafft der Factory-Reset abhilfe. Im Fall es Falles den **SCL Pin** des Programmers mit **XTAL1** der MCU verbinden und Factory-Reset anklicken. Danach wieder trennen.*

3.4.4 Oscillator Calibration Byte/Kalibriungsbytes

When using an internal oscillator, you might need a calibration value for the oscillator. This calibration value cannot be accessed from a program. Therefore it is necessary to copy the value into the Flash memory or EEPROM.

Amadeus kann die Kalibriungsbytes bei Bedarf ans Ende des Flashspeichers oder an den Anfang/das Ende des EEPROMs kopieren.

The programmer itself does *not* offer a calibration routine, but if you otherwise managed to calibrate your MCU, and the calibration value is stored in the last Flash memory cell (high byte), it is possible to preserve this value during flash programming.

*Eine neu Kalibrierung kann der Programmer nicht durchführen, aber wer seinen Chip genau kalibriert hat, sollte diesen Wert ans Ende des Flashspeicher legen (Highbyte) und beim Programmieren **Preserve Calibration Byte at end of Flash** wählen.*

Hint: In case of **Preserve Calibration Byte** never erase the chip without reprogramming the Flash memory, or you will lose your manual calibration. This is also true for the Factory Reset.

***Hinweis:** Im Falle von **Preserve Calibration Byte** nie das Flash löschen ohne es neu zu programmieren. Beim **Factory Reset** geht das Byte ebenfalls verloren.*

3.4.5 EEPROM Programming/Programmierung

For the ATmega family, there's a *specail* way Amadeus handles EEPROM programming. If the **EESave** fuse is set, or you don't select **Erase Chip**, only EEPROM values given in the hex-file are updated. If there's a rage of EEPROM addresses missing in the hex-file, those EEPROM cells will not be updated (no 0xFF will be written there). Keep this in mind, if you store your manual calibration value inside the EEPROM.

Für die ATmega Familie handhabt Amadeus die EEPROM Programmierung auf eine besondere Art. Wenn EESave aktiv ist und Erase Chip nicht markiert ist, werden nur die in der Hex-Datei definierten Bytes programmiert. Die anderen Zellen bleiben unberührt.

Das kann für die Ablage des Kalibrierungsbytes im EEPROM von bedeutung sein.

3.5 PIC12F 629/675 / PIC16F 630/676

The four MCUs have identical programming features. They contain 1kWord (14bit) of flash memory (0x000 to 0x3ff) and 128Bytes of EEPROM. Further more an ID-location of 4x14bit words and a set of configuration bits.

Internal clock is 4MHz (1MIPS). Externally, up to 20MHz (5MIPS) are possible.

At flash address 0x3ff (hex-file 0x7fe), Microchip stores the Oscillator Calibration Byte in form of a RETLW xx instruction. So the user's software must end at address 0x3fe. The programming software restores the original Calibration Byte after every programming/flash erasing.

Another important calibration value is the Band Gap Calibration. It occupies two bits in the configuration word. They are also restored after every programming/flash erasing.

Amadeus programs the application software area (flash memory), EEPROM, ID-location and configuration, when they are contained in the hex file.

Es gibt zwei Abgleichwerte, die immer wieder hergestellt werden müssen. Nämlich die Oszillatorkalibrierung (letzte Wort im Flash) und die BG-Kalibrierung (im ID-Wort).

Hex-File-Address	Contents
0x0000 to 0x07fd	Software for MCU (14bit/word)
0x07fe to 0x07ff (will be ignored)	Oscillator Calibration Byte
0x4000 to 0x4007	4-14bit words programmed to ID-location
0x400e to 0x400f	Configuration (bits 0-8, BG bits 12/13 will be ignored)
0x4200 to 0x42ff	EEPROM 128Bytes (every second bytes will be programmed, every other byte in the hex-file contains 0x00). For details, see MP-Lab help.

Table 5: PIC12/16 and compatible hex-file definition

Besides the contents of the hex-file, it is possible to edit the configuration-bits. When **Ignore Hex-File Configuration** is marked, the hex-file's configuration is replaced.

*Wenn **Ignore Hex-File Configuration** markiert ist, werden die angezeigten Konfigurationsdaten für die Programmierung/Vergleich verwendet.*

3.5.1 How To Connect/Programmierschnittstelle

For electrical details see chapter “Electrical Characteristics/Elektrische Eigenschaften“ on page 24.

Für elektrische Details bitte im Kapitel “Electrical Characteristics/Elektrische Eigenschaften“ auf Seite 24 nachschlagen.

4 lines (+ power supply switching) are required for ICSP.

4 Leitungen (+Spannungsfreischaltung) werden für ICSP benötigt.

<i>Amadeus side</i>	<i>MCU side</i>	<i>comment</i>
n.c.	VCC	For in-circuit programming , the MCU gets its power from the target circuit. <i>Beim Programmieren in der Schaltung nicht anschließen.</i>
EXTVCC !!!	VCC	For stand alone programming only!!! Do never connect EXTVCC for in-circuit programming. <i>Spannungsversorgung für Standalone Programmiergeräte. Nie mit einer externen Spannung verbinden.</i>
GND	GND	Connecting one GND line might be enough. Connect both, if you are using longer programming cables. <i>Masseverbindung.</i>
MCLR/RST	MCLR	Reset (about 12V during programming) <i>Reset kann während der Programmierung auf 12V gehen.</i>
PGD/PDO	ICSPDAT	Serial Data
PGC/SCK	ICSPCLK	Serial Clock

Table 6: PIC12F/16F ICSP adapter definition

The programming specification says, that it is necessary to switch power (MCU supply voltage) on, after VPP has been applied. Practical tests from my side showed (at least for my PIC12F675), that power switching was not necessary.

However, the programmer can handle this situation, if required. For details on Power Switching see page 10.

Laut Datenblatt muß die Betriebsspannung geschaltet werden, wenn der Reset-Pin deaktiviert wurde.

Versuche haben aber gezeigt, daß es offensichtlich auch ohne geht. Bitte selber ausprobieren.

3.5.2 Programming/Programmierung

Only additional information on programming are given in the paragraph. Have a look at the chapter “Basics/Grundlagen” on page 8 for the basic programming procedure.

Die Grundlagen findest Du im Kapitel “Basics/Grundlagen“ auf Seite 8.

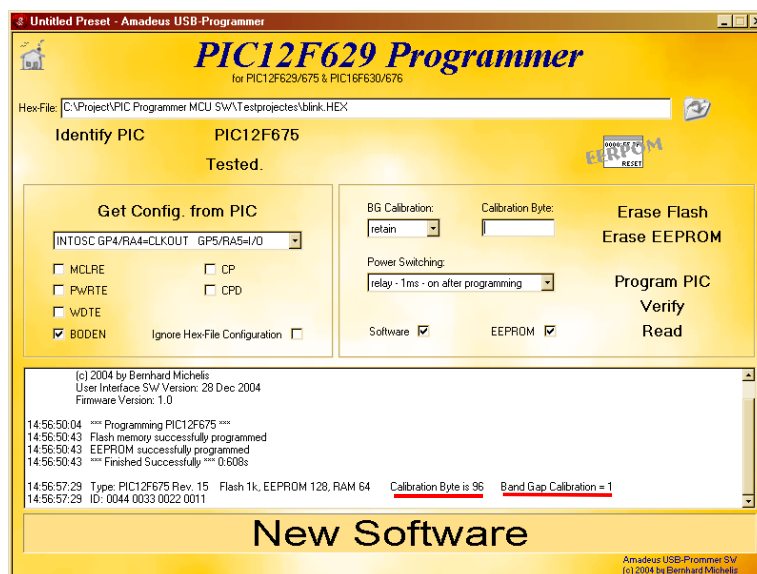


Illustration 8: PIC12F629 Programming Interface

Click on **Identify PIC**. The software tries to identify the target MCU. When the connection is working and the target device is powered up, there should be a message, displaying the MCU's name. Furthermore the *Oscillator Calibration Byte* (0-255) and the *BG Calibration* (0-3) will be displayed. Although they are restored automatically, it is suggested to write them down somewhere, to restore them manually, if something went wrong.

*Beim ersten erkennen des PICs mit Identify PIC sollte man sich das **Oscillator Calibration Byte** (0-255) und den **BG Calibration** Wert (0-3) aufschreiben, falls mal etwas schief geht.*

When the checkbox **Software** is checked, Flash memory, ID-location and configuration will be programmed. When **EEPROM** is checked, the EEPROM area will be programmed.

*Wenn **Software** abgehakt ist, werden Flashspeicher, ID-Location und Konfigurationsdaten programmiert. Wenn **EEPROM** abgehakt ist, wird das EEPROM programmiert.*

Erase Flash erases the program memory, ID-location and configuration and **Erase EEPROM** erases the EEPROM.

***Erase Flash** löscht den Flash Speicher, die ID-Location und die Konfigurationsdaten. **Erase EEPROM** das EEPROM.*

3.5.3 Calibration Recovery/Kalibrationswiederherstellung

If, for any reason, one of the calibration values vanished, it is possible to restore them manually.

Therefore you need the two values, you got, when you first clicked **Identify PIC**. Enter the value in the field **Calibration-Byte** and select the correct **BG-Calibration** value. Now click on **Erase Flash**. That's it. When **BG-Calibration** shows **retain**, **Erase Flash** doesn't touch any of the saved values.

*Wenn aus irgend einem Grund die Kalibrierung wieder hergestellt werden muß, muß der Oszillator Kalibrierungswert in **Calibration-Byte** geschrieben werden und der entsprechende*

*Eintrag unter **BG-Calibration** ausgewählt werden. Danach **Erase Flash** anklicken. Wenn **BG-Calibration retain** angezeigt, werden keine Kalibrierungsdaten geschrieben.*

3.6 PIC18F 242/248/252/258/442/448/452/458

At the moment, only these eight PIC18F MCUs are supported. They contain 16kBytes or 32kBytes Flash memory, 256Byte EEPROM, an 8Byte ID-Location and 14 configuration bytes.

There's no internal clock. Externally, up to 40MHz (10MIPS) are possible or in HS-PLL mode up to 10MHz (10MIPS!!!, too).

There are no calibration values, that have to be preserved.

Im Moment werden nur diese Typen unterstützt.

Es gibt keine Kalibrierungswerte die gesichert werden müßten.

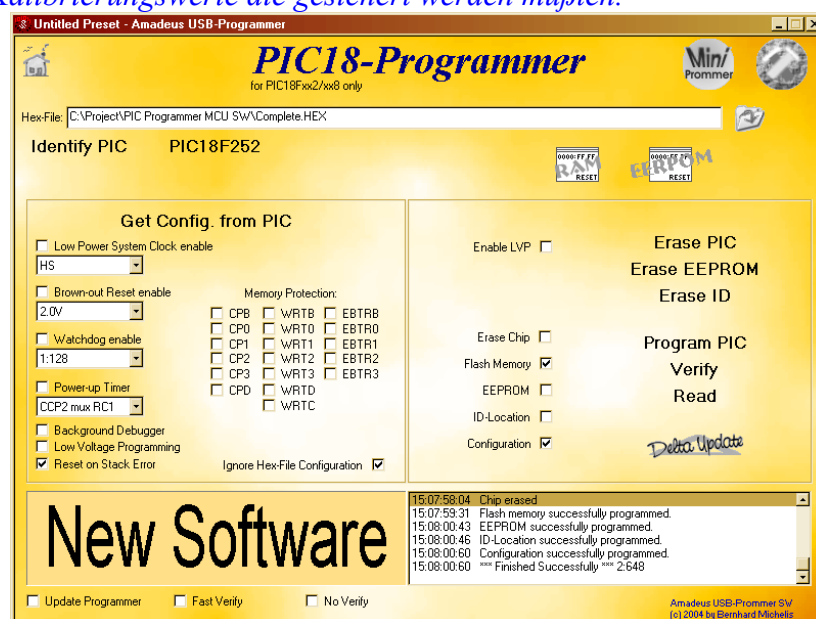


Illustration 9: PIC18F Programming Interface

3.6.1 How To Connect/Programmierschnittstelle

For electrical details see chapter “Electrical Characteristics/Elektrische Eigenschaften“ on page 24.

Für elektrische Details bitte im Kapitel “Electrical Characteristics/Elektrische Eigenschaften“ auf Seite 24 nachschlagen.

4/5 lines are required for programming. *4/5 Leitungen werden benötigt.*

<i>Amadeus side</i>	<i>MCU side</i>	<i>comment</i>
n.c.	VCC	For in-circuit programming , the MCU gets its power from the target circuit. <i>Beim Programmieren in der Schaltung nicht anschließen.</i>
EXTVCC !!!	VCC	For stand alone programming only!!! Do never connect EXTVCC for in-circuit programming. <i>Spannungsversorgung für Standalone Programmiergeräte. Nie mit einer externen Spannung verbinden.</i>
GND	GND	Connecting one GND line might be enough. Connect both, if you are using longer programming cables. <i>Masseverbindung.</i>
MCLR/RST	MCLR	Reset (about 12V during programming). LVP must not be set. <i>Reset kann während der Programmierung auf 12V gehen. LVP darf nicht gesetzt sein.</i>
PGD/PDO	PGD	Serial Data
PGC/SCK	PDC	Serial Clock
PGM/PDI	PGM	Only required for Low Voltage Programming. Enable LVP must be checked. <i>Nur erforderlich für Low Voltage Programmierung. LVP muß gesetzt sein.</i>

3.6.2 Programming/Programmierung

The PIC18F family offers a wide range of programming possibilities, that offer very fast programming. During development/debugging programming time can be reduced, when you keep the following hints in mind.

Die PIC18F Familie bietet viele Updatemöglichkeiten. Um die Programmierung zu beschleunigen immer Fast Verify setzen. Unterschiedliche Programmfragmente sollten bei 0x0000, 0x2000, 0x4000, 0x6000 beginnen. Darüber hinaus sollte man Erase Chip deselektieren und alle Programmierschritte entfernen, die nicht benötigt werden.

- Always select **Fast Verify**.
- The currently supported devices offer multi-panel flashing. So it is faster when different parts of the SW start at *org 0x0000* and *org 0x2000* and for 32kBytes devices in addition *org 0x4000* and *org 0x6000* instead of one big program chunk.
- PIC18F devices don't need to erase the complete chip, before they can reprogram the application software. So remove **Erase Chip** during development and disable all areas (EEPROM, ID-location and configuration) that don't need an update. Empty areas will be skipped during programming.

<i>Action</i>	<i>Duration</i>
Programming and verifying 16kBytes Flash (full)	2.5s
Programming and verifying 32kBytes Flash (full)	2.7s
EEPROM programming and verifying 256Bytes (full)	1s

Table 7: PIC18Fxx2/xx8 programming durations

For super fast software updates during development (down to 200ms programming time) you can use the **Delta Update** function. It is only available for the program flash memory. It can update single flash segments, while others aren't touched. To prevent large parts of the program to be shifted in memory (takes time), it is strongly recommended to leave some space after a function block. Use the **org** directive to give ISR, program main loop, functions for this, functions for that fixed addresses. Otherwise it is slower than normal programming.

Important: Before using **Delta Update**, you must have programmed the MCU normally (at least for one time), because Amadeus does not compare new data in hex-file with the data in the MCU, but with the data from last programming. If you restart Amadeus, the data history is all 0xFF.

*Für superschnelle Updates gibt es die **Delta Update** Funktion. Hier kann man kleine Änderungen in 200ms updaten. Das funktioniert aber nur, wenn sich nicht alle nachfolgenden Instruktionen verschieben. Deshalb sollte man den verschiedenen Programmteilen unterschiedliche, fixe Adressen zuweisen und zwischen ihnen etwas Platz für Erweiterungen lassen.*

Bevor Du mit Delta Update arbeitest, muß die MCU einmal normal programmiert werden. Diese gilt auch, wenn Amadeus neu gestartet wurde.

In case the programmer cannot find the PIC, check that **Enable LVP** is set correctly.

*Wenn der Programmer den PIC nicht finden kann, sollte man prüfen, ob **Enable LVP** richtig gesetzt ist. Für High-Voltage Programmierung LVP löschen.*

3.6.3 Mini Prommer Window

Have a look at the *Mini Prommer* Window. It stays always on top of your IDE, so you can quickly start the Flash update with one click.

Das Mini Prommer Fenster ist ein Fenster, das immer im Vordergrund bleibt. So kann man mit nur einem Klick sein Update durchführen.

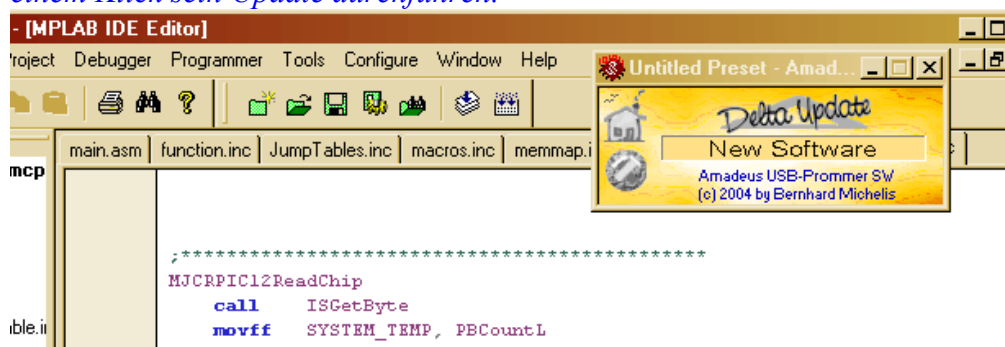


Illustration 10: MP-Lab with Mini-Prommer window in front of it

4 Amadeus Programmer Hardware

4.1 Electrical Characteristics/Elektrische Eigenschaften

4.1.1 Serial Programming Interface

At the moment, the complete hardware is designed for 5V programming. Every output can drive about 20mA.

Derzeit arbeitet die Hardware nur mit 5V. Die Ausgänge können ca. 20mA Treiben. Die 12V dürfen nicht belastet werden max. 2-3mA.

If you plan to program MCUs working with less than 5V you might follow the following idea (**on your own risk**). Nearly every pin of nowadays MCUs have internal ESD protection diodes, that pull down the voltage to ca 0.6V above VCC. VCC might rise a little, too, depending on the load of other components on VCC. With a resistor in the programming cable this might work for you (**On your own risk!!!**). Also keep in mind, that some timings might shift depending on VCC, so that programming might be impossible.

I, personally, tried an ATmega64 @3V with the upper mentioned method. Worked so far.

Have a look on the following tables. They show, where to find every programmer pin.

Die folgenden Tabellen zeigen die Pinbelegung des Programmers.

<i>Pin on Programmer Hardware</i>	<i>Description</i>
Pin 1: SCL	High-impedance during stand-by. Can drive up to 20mA during programming. <i>Hochohmig im Standby. Kann während der Programmierung bis zu 20mA treiben.</i>
Pin 2: GND	Ground <i>Masse</i>
Pin 3: SDA	High-impedance during stand-by. Can drive up to 20mA during programming. <i>Hochohmig im Standby. Kann während der Programmierung bis zu 20mA treiben.</i>

Table 8: N2

<i>Pin on Programmer Hardware</i>	<i>Description</i>
Pin 1: PGM/PDI	High-impedance during stand-by. Can drive up to 20mA during programming. <i>Hochohmig im Standby. Kann während der Programmierung bis zu 20mA treiben.</i>
Pin 2: GND	Ground <i>Masse</i>
Pin 3: PGD/PDO	High-impedance during stand-by. Can drive up to 20mA during programming. <i>Hochohmig im Standby. Kann während der Programmierung bis zu 20mA treiben.</i>
Pin 4: GND	Ground <i>Masse</i>
Pin 5: PGC/SCK	High-impedance during stand-by. Can drive up to 20mA during programming. <i>Hochohmig im Standby. Kann während der Programmierung bis zu 20mA treiben.</i>
Pin 6: MCLR/RST	High-impedance (0.1mA@5V/ca 55kΩ) during stand-by. See 74LS05 datasheet for details on open-collector output. Keep in mind, that this pin rises to ca 12V during high-voltage programming when designing your target device. If the MCU pin has reset functionality, use a 100-200kΩ pull-up resistor. If the <i>reset</i> is deactivated, the connected components must not draw more than 2-3mA from this line. Otherwise the 12V breaks down. <i>Für jochohmig, siehe Datenblatt zu 74LS05.</i> <i>Wenn ein Pull-Up Widerstand für Reset benötigt wird sollte er im bereich 100-200kΩ liegen.</i>
Pin 7: EXT VCC	Switched power-supply for standalone programming. Do never connect to any other power supply. Do not draw more than 100mA. <i>Spannungsversorgung für Standalone Programmer. Nie mit mehr als 100mA belasten.</i>

Table 9: N3

<i>Pin on Programmer Hardware</i>	<i>Description</i>
N4(Pin1) to N5(Pin1)	Relay (50mΩ; ca 500ns). Use only for voltages below 20V <i>Nicht mit mehr als 20V verwenden.</i>
N4(Pin2) to N5(Pin2)	Relay (50mΩ; ca 500ns). Use only for voltages below 20V <i>Nicht mit mehr als 20V verwenden.</i>

Table 10: N4 and N5

4.2 Schematics

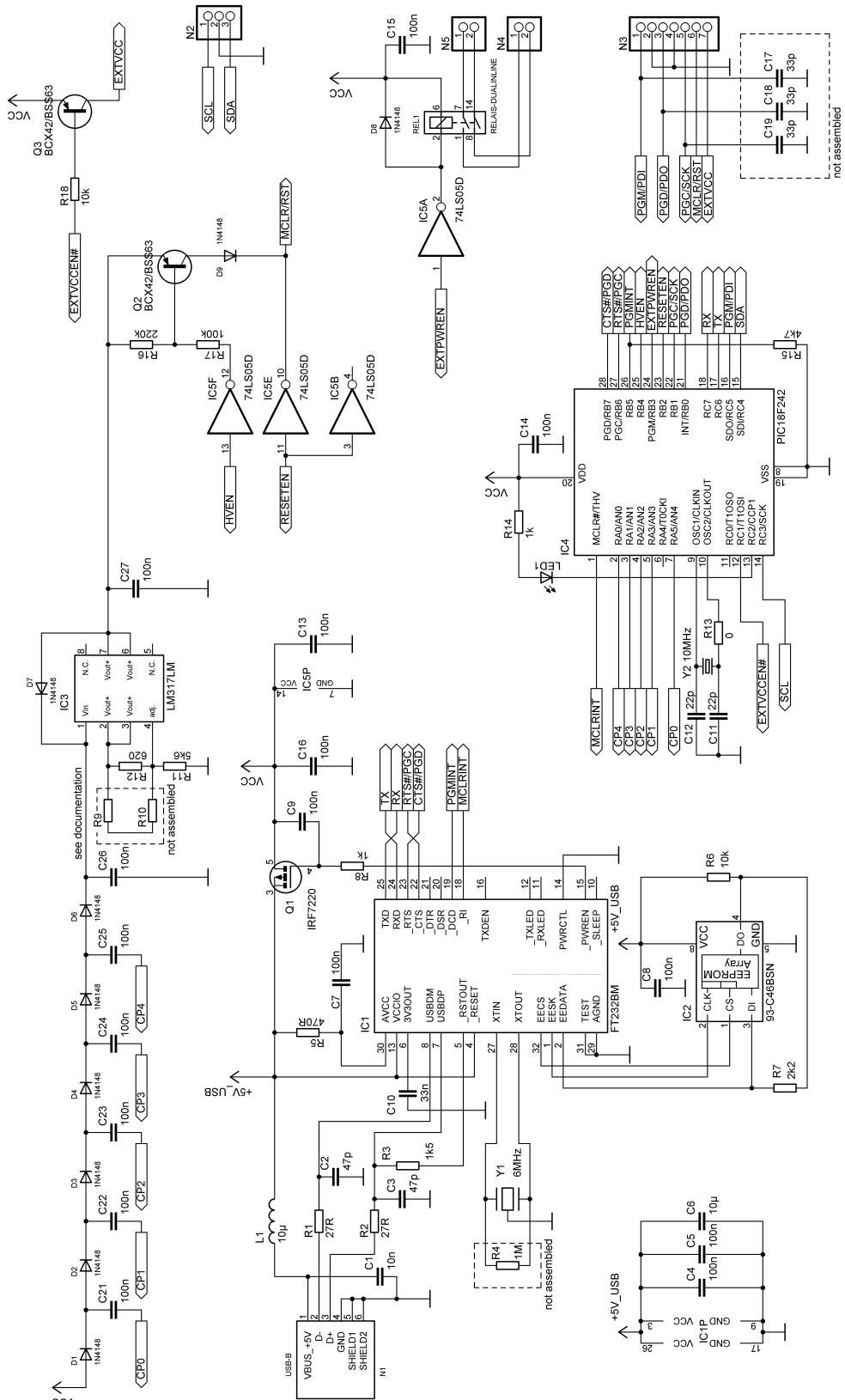


Illustration 11: Schematics

4.3 Partlist

Bezugsquelle für Deutschland: Derzeit (Dezember 2004) sind alle Teile bei www.reichelt.de zu bekommen.

<i>Position</i>	<i>Value</i>	<i>Package</i>	<i>Comment/Reichelt ordercode</i>
C1	10n	C0805	
C10	33n	C0805	
C11	22p	C0805	
C12	22p	C0805	
C13	100n	C1206 or C0805	
C14	100n	C0805	
C15	100n	C0805	
C16	100n	C0805	
C17	Not Assembled	C0805	
C18	Not Assembled	C0805	
C19	Not Assembled	C0805	
C2	47p	C0805	
C21	100n	C0805	
C22	100n	C0805	
C23	100n	C0805	
C24	100n	C0805	
C25	100n	C0805	
C26	100n	C0805	
C27	100n	C0805	
C3	47p	C0805	
C4	100n	C0805	
C5	100n	C0805	
C6	10 μ	C5,8X3,2MM	
C7	100n	C0805	
C8	100n	C0805	
C9	100n	C0805	
D1	1N4148 SMD	MINI-MELF	1N 4148 SMD
D2	1N4148 SMD	MINI-MELF	
D3	1N4148 SMD	MINI-MELF	
D4	1N4148 SMD	MINI-MELF	
D5	1N4148 SMD	MINI-MELF	
D6	1N4148 SMD	MINI-MELF	
D7	1N4148 SMD	MINI-MELF	
D8	1N4148 SMD	MINI-MELF	
D9	1N4148 SMD	MINI-MELF	
IC1	FT232BM	LQFP32	FTDI FT232BM
IC2	93-C46BSN	SO-08	
IC3	LM317LM	SO-08	LM317LM

Position	Value	Package	Comment/Reichelt ordercode
IC4	PIC18F242 or PIC18F252	SO-28W	PIC18F242 (or PIC18F252 might be interesting for Updates) <i>Important: If you are recycling an old PIC, Low Voltage Programming must be enabled.</i> <i>PIC18F252 wäre für eventuelle, spätere Erweiterungen sinnvoll. Derzeit ist der Speicher zu 60% gefüllt.</i>
IC5	74LS05D	SO-14	
L1	10 μ	L0805	JCI 2012 10 μ
LED1	LED5MM	LED5MM	
N1	USB_SOCKET	USB-BW Assmann	USB BW for print assembly
N2	1X03	pinhead	
N3	1X07	pinhead	
N4	1X02	pinhead	2,54mm or 0.1"
N5	1X02	pinhead	
Q1	IRF7220	SO-08	
Q2	BCX42/BSS63	SOT-23	
Q3	BCX42/BSS63	SOT-23	
R1	27R	R1206	
R10	Not Assembled	R1206	You can assembled other resistors (R9+R10 R12) to adjust the accuracy of VPP (12V) <i>Mit den Widerständen werden die 12V eingestellt.</i>
R11	5k6	R1206	
R12	620R	R1206	You can assembled other resistors (R9+R10 R12) to adjust the accuracy of VPP (12V) <i>Mit den Widerständen werden die 12V eingestellt.</i>
R13	0R	R1206	
R14	1k	R1206	
R15	4k7	R1206	
R16	220k	R1206	
R17	100k	R1206	
R18	10k	R1206	
R2	27R	R1206	
R3	1k5	R1206	
R4	Not Assembled	R1206	
R5	470R	R1206	
R6	10k	R1206	
R7	2k2	R1206	
R8	1k	R1206	
R9	Not Assembled	R1206	You can assembled other resistors (R9+R10 R12) to adjust the accuracy of VPP (12V) <i>Mit den Widerständen werden die 12V eingestellt.</i>
REL1	RELAIS-DUALINLINE	DIL-14-RELAIS	MEDER electronic "DIP 7221-L 5V" www.meder.com
Y1	6MHz	RESONATOR	CSTCC 6,00
Y2	10MHz	Crystal HC49U-/S or -HC18	10-HC18

Table 11: Partlist

4.4 Assembly

Here is the assembly instruction for the programmer hardware. As you can see, it's only a single side PCB.

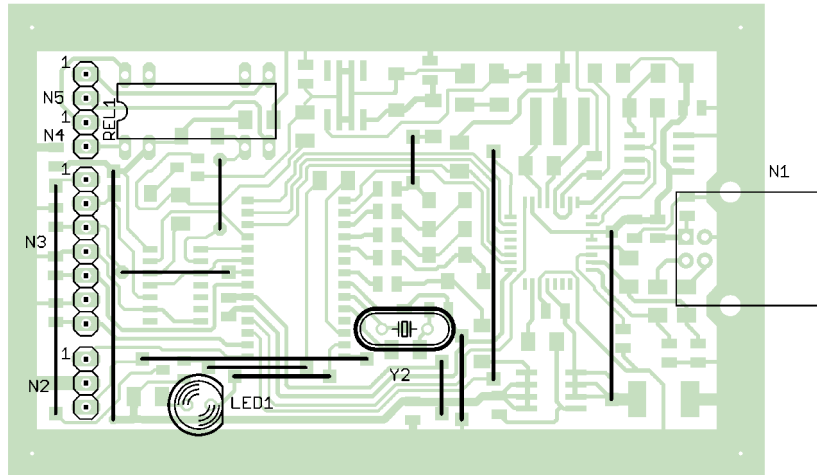


Illustration 12: PCB top view

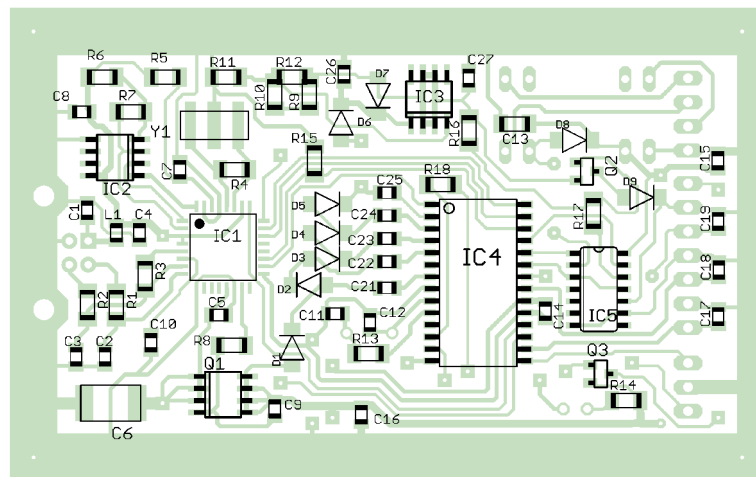
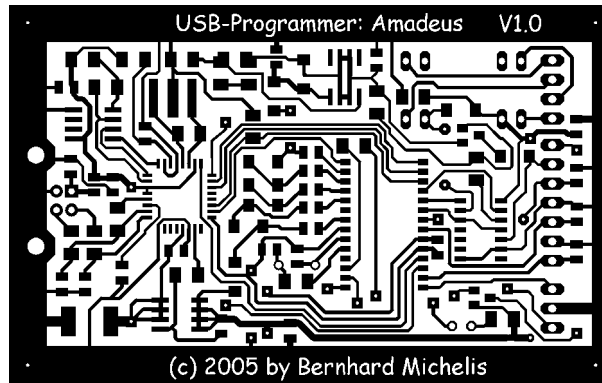


Illustration 13: PCB bottom view

4.5 Layout

The layout is available as a separate file named “Amadeus_PCB.gif” with a higher resolution. After printing the layout must be 80mmx50mm of size.

Nach dem Drucken muß das Layout 80mmx50mm groß sein.



History

<i>Date</i>	<i>Comment</i>
29.12.04	First version
02.01.05	German documentation added