

Handbuch

Ätzmaschine

(für Firmware 2.0 mit Bootloader 1.1)



Selbstbauprojekt

Inhaltsverzeichnis

1	Einführung	3
1.1	Was ist eine Ätzmaschine	3
1.2	Funktionsprinzip der Ätzmaschine.....	3
2	Bedienung.....	4
2.1	Bedienelemente und Anschlüsse	4
2.1.1	Frontseite	4
2.1.2	Anschluss-Seite	5
2.2	Inbetriebnahme	6
3	Service-Mode	7
3.1	Service-Mode aktivieren	7
3.2	Servo-Anschläge einstellen	7
3.3	Service-Mode verlassen.....	7
4	Das serielle Interface RS232.....	8
4.1	RS232-Parameter	8
4.2	Das RS232-Protokoll	8
4.3	Kommandos	9
4.3.1	Reset (Kommando 1)	9
4.3.2	EEProm (Kommando 2)	10
4.3.3	Debug-Modus (Kommando 3)	10
4.3.4	Debug-Interval (Kommando 4)	11
4.3.5	Speed (Kommando 10)	11
4.3.6	Hub (Kommando 11)	11
4.3.7	Soll-Temperatur (Kommando 12)	12
4.3.8	Ist-Temperatur (Kommando 13)	12
4.3.9	Servo-Begrenzung Links (Kommando 20)	13
4.3.10	Servo-Begrenzung Rechts (Kommando 21)	13
4.3.11	Servo-Mittelstellung (Kommando 22)	13
4.3.12	Servo-Position (Kommando 23)	14
4.3.13	Heizleistung (Kommando 30)	14
4.3.14	Sonder-Kommandos (Hilfe und Versions-Info)	16
4.3.14.1	Versions-Info (Kommando ,?') 16	
4.3.14.2	Hilfe (Kommando ,h') 17	
5	Der Debug-Modus	18
6	Hyper-Terminal.....	19
6.1	Terminalprogramm einrichten	19
7	Firmware und Bootloader	24
7.1	Info zum Bootloader.....	24
7.1.1	Programmieradapter	24
7.1.2	Programmer wählen	24
7.1.3	Fusebits am Controller setzen	25
7.1.4	Bootloader einspielen	26
7.2	Firmware aktualisieren.....	30
7.2.1	Programmierschritte	30

8	Anhang	31
8.1	Steckerbelegung Platine.....	31
8.2	Montagematerial	32
8.3	Versionen - History.....	33
8.4	Abbildungs-Verzeichnis	34
8.5	Tabellen-Verzeichnis	34

1 Einführung

Dieses Handbuch beschreibt den Aufbau und die Funktionsweise einer Selbstbau-Ätzmaschine, wie sie im Hobby-Bereich Verwendung finden kann. Diese Ätzmaschine wurde speziell für diesen Zweck entwickelt und dient in erster Line dazu, Platinen zu ätzen. Der Aufbau ist einfach gehalten, um den Nachbau zu fördern.

1.1 Was ist eine Ätzmaschine

Eine Ätzmaschine dient der Leiterplatten-Herstellung - im speziellen zum Ätzen bereits belichteter und entwickelter Platinen. Sie sorgt zum einen für eine kontinuierliche Badbewegung während des Ätzevorgangs, um Unterätzungen und ungleichmäßiges Ätzen zu vermeiden. Zum anderen sorgt die Ätzmaschine für eine temperierte Bad-Temperatur (40°C bis 50°C), um den Ätzevorgang zu beschleunigen.

1.2 Funktionsprinzip der Ätzmaschine

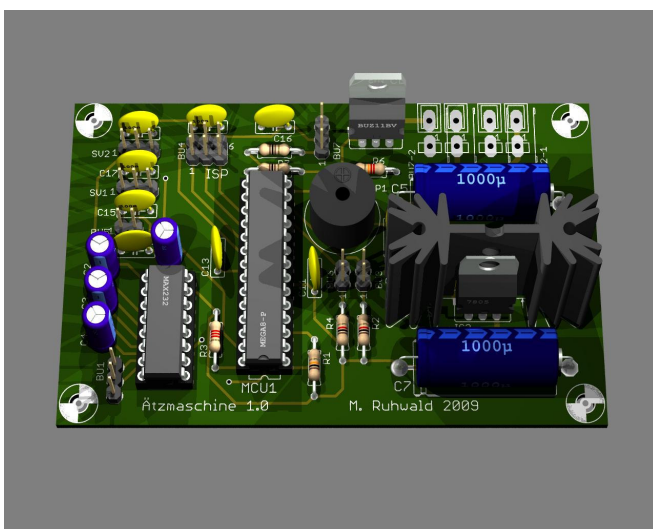


Abbildung 1 - Ansicht der Platine in 3D

Die Ätzmaschine besteht im Wesentlichen aus 2 Komponenten. Zum einen aus einem Gehäuse, welches überwiegend aus Holz gefertigt ist. Das Besondere am Gehäuse ist die bewegliche Wipp-Platte, auf die eine Ätzechale gestellt wird. Durch die Wipp-Bewegung kann so das Ätzebad in Bewegung gehalten werden.

Die zweite Komponente ist die Steuer-Elektronik. Sie sorgt für die korrekte Ansteuerung eines handelsüblichen Stell-Servo (6V) aus dem Modellbau, der die Wipp-Platte definiert bewegt.

Als Kernstück der Steuerung kommt ein ATmega8 Controller der Firma ATMEL zum Einsatz. Über 2 Potentiometer können Geschwindigkeit und Hub der Wipp-Platte eingestellt werden.

Als weiteres Feature ist auf der Steuerplatine eine Regelung vorgesehen, um die Ätzebad-Temperatur auf eine geeignete Ätze-Temperatur zwischen 40°C und 50°C einzustellen. Ein Temperatur-Sensor (KTY81-200) dient hierbei als Temperaturfühler.

Die wichtigsten Bedien-Funktionen der Ätzmaschine können mit den vorhandenen Potentiometern auf der Frontseite eingestellt werden. Weitere Einstellungen können über die serielle Schnittstelle durchgeführt werden.

2 Bedienung

2.1 Bedienelemente und Anschlüsse

Auf der Frontseite der Ätzmaschine befinden sich die Bedienungs- und Anzeige-Elemente. Die linke Seite ist mit allen notwendigen Anschlüssen versehen, die für den Betrieb und die Parametrierung notwendig sind.

2.1.1 Frontseite

Schematische Ansicht der Frontplatte

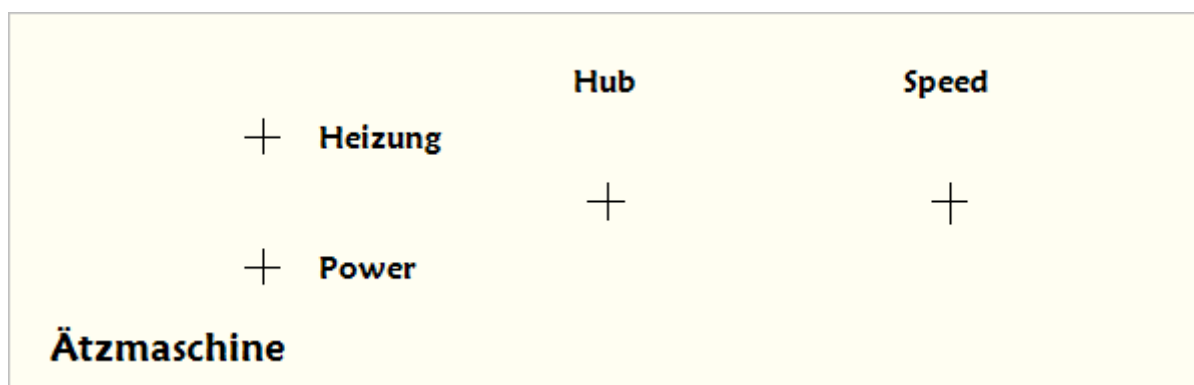


Abbildung 2.1 - Frontseite

Auf der Vorderseite der Ätzmaschine befinden sich zwei Potentiometer mit den Bezeichnungen „Hub“ und „Speed“.

Mit dem Einstellregler „Hub“ kann die Stärke der Wipp-Bewegung (auch Hub genannt) eingestellt werden.

Mit dem Einstellregler „Speed“ kann die Wipp-Geschwindigkeit der beweglichen Wipp-Platte eingestellt werden.

Links neben den Potentiometern auf der Frontplatte befinden sich zwei Leuchtdioden mit den Bezeichnungen „Heizung“ und „Power“.

Die rote LED „Heizung“ zeigt die aktive Heizphase der Heizung an. Sie leuchtet immer dann, wenn das Heizelement heizt.

Hinweis:

Die LED ist nur dann aktiv, wenn ein Temperatursensor angeschlossen ist!
Bei 100% Heizleistung leuchtet die LED permanent. Wenn die zu regelnde Temperatur annähernd erreicht wird, blinkt die LED im 0,5Hz Takt. Die Dauer der An-Zeit zur Aus-Zeit der LED ist ein Maß für die ausgegebene Heizleistung. (1 Sekunde an und 1 Sekunde aus = 50% Leistung)

Die grüne LED „Power“ zeigt den Betriebs-Modus der Ätzmaschine an. Sie leuchtet im Betrieb permanent.

2.1.2 Anschluss-Seite

Die Anschluss-Seite weist vier Anschluss-Buchsen auf.

- ✓ Versorgung
- ✓ Heizung
- ✓ KTY81
- ✓ RS232

Schematische Ansicht der Anschluss-Seite

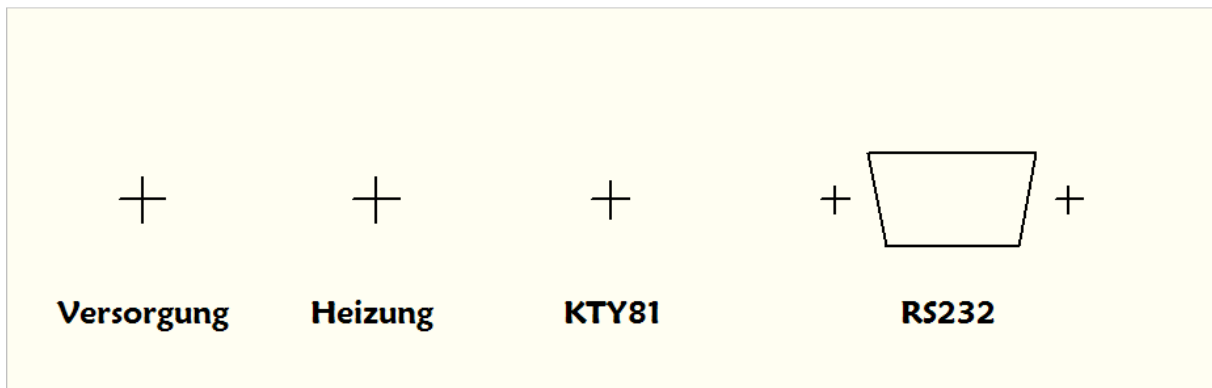


Abbildung 2.2 - Anschluss-Seite

Versorgung

An diesem Anschluss wird die Versorgungsspannung der Ätzmaschine angeschlossen. Grundsätzlich kann hier eine stabilisierte Gleichspannungs-Versorgung von 8V bis 30V angeschlossen werden. Es ist dabei unbedingt auf richtige Polarität achten! Dieser weite Spannungsbereich kommt dem verwendeten Heizelement zu gute. Es ist angeraten, die Versorgungsspannung so niedrig wie möglich zu halten.

Als Spannungsversorgung empfehle ich ein Stecker-Netzteil vom Typ SNG24-48W-A. Es handelt sich hierbei um ein Schaltnetzteil und liefert eine einstellbare Ausgangsspannung bis 24 Volt und 2 Ampere. Erhältlich bei www.Conrad.de, Best.-Nr. 510525.

Heizung

An diesen Anschluss wird ein geeignetes Heizelement mit einer Leistung von ca. 30W bis 45W angeschlossen. Diese Heizleistung ist notwendig, um die Ätzbad-Temperatur auf ca. 40°C bis 50°C zu erwärmen. Ein verwendbares Heizelement ist im Anhang 8.2 unter Montagematerial zu finden.

KTY81

An den Anschluss „KTY81“ kann optional ein Temperaturfühler vom Typ KTY81-200 angeschlossen werden. Der Controller erkennt einen angeschlossenen Fühler und startet automatisch die Temperatur-Regelung des Ätzbades. Hierfür ist jedoch ein Heizelement erforderlich. Die aktivierte Heizung wird auf der Frontplatte durch die rote LED „Heizung“ signalisiert.

RS232

Nicht alle Einstellungen der Ätzmaschine sind über die Potentiometer an der Frontseite einstellbar. Deshalb wurde die Ätzmaschine mit einer seriellen Schnittstelle ausgestattet, um die Parameter zu erreichen. Die Schnittstelle ist auch für Diagnose des Ätzgeräts und Firmware-Updates einsetzbar.

RS232-Einstellung: 19200Baud, 8 Datenbits, 1 Stoppbit, keine Parität, kein Handshake

2.2 Inbetriebnahme

- Schließen Sie (wenn vorhanden) zunächst das Heizelement an die Buchse „Heizung“ an.
- An die Buchse „KTY81“ schließen sie (wenn vorhanden) den Temperatur-Sensor (Typ KTY81-200) an.
- Drehen sie beide Einstellregler etwa in Mittelstellung.
- Schließen Sie die externe Versorgungsspannung an die Buchse „Versorgung“ an.

Auf der linken Seite der Ätzmaschine sind 4 Buchsen angeordnet. An der Buchse „Versorgung“ wird eine externe Versorgungsspannung zwischen 9V und 24V Gleichspannung angeschlossen.

Die Betriebsspannung ist überwiegend vom Heizelement abhängig. Wenn ein Heizelement mit 12V verwendet wird, sollte auch die Versorgungsspannung diesen Wert nicht überschreiten, da bei 100% Heizleistung die Versorgungsspannung permanent am Heizelement anliegt. Es ist bedingt jedoch auch möglich, höhere Betriebsspannungen zu verwenden, wenn die Heizleistungsbegrenzung verwendet wird (siehe Absatz 4.3.13).

An die Buchse „KTY-81“ kann optional ein Temperatur-Sensor vom **Typ KTY81-200** angeschlossen werden. Der Sensor (PTC) ermittelt die Temperatur des Ätzbades um die Leistung für das Heizelement zu bestimmen. Der Anschluss dieses Sensors macht jedoch nur Sinn, wenn auch ein Heizelement verwendet wird. Der Sensor wird mit Schrumpfschlauch wasserdicht umschlossen und in die Ätzflüssigkeit eingetaucht.

Alle Parameter der Ätzmaschine sind voreingestellt, so dass die Funktion des Gerätes überprüft werden kann.

Wenn die Funktion soweit in Ordnung ist, d. h. dass der Hub der Wipp-Platte sich erhöht, wenn der Hub-Poti nach rechts gedreht wird und die Wipp-Geschwindigkeit nimmt zu, wenn der Speed-Poti nach rechts gedreht wird, kann der Service-Mode aufgerufen werden .

Der Service-Mode ist wichtig und sollte unbedingt durchgeführt werden. Damit werden die Anschläge des Servo und die waagerechte Stellung der Wipp-Platte justiert. Mehr zum Service-Mode erfahren Sie im Absatz 3 Service-Mode.

3 Service-Mode

Der Service-Mode dient in erster Linie zur Einstellung von Parametern, die nur selten eingestellt werden müssen. Hierzu gehören die folgenden Parameter:

- Mechanischer Anschlag des Servo Links
- Mechanischer Anschlag des Servo Rechts
- Mechanische Mittelstellung der Wipp-Platte

3.1 Service-Mode aktivieren

Die Aktivierung des Service-Mode sowie die darauf folgenden Einstellungen werden mit den beiden Einstellern „Hub“ und „Speed“ gesteuert.

Um in den Service-Mode zu gelangen, müssen die beiden Einsteller „Hub“ und „Speed“ auf rechten Anschlag gebracht werden und dann die Versorgungsspannung einstecken.

Nach ca. 1-2 Sekunden piepst die Ätzmaschine 4x und zeigt damit den aktivierten Service-Mode an.

3.2 Servo-Anschläge einstellen

Poti „Hub“ in Mittelstellung bringen. Die Ätzmaschine piepst 1x und der erste Parameter kann eingestellt werden.

Servoanschlag mit dem Poti „Speed“ einstellen. Die Wippe bewegt sich synchron zum Poti „Speed“. Den Anschlag so einstellen, dass ein maximaler Hub erreicht werden kann. Der Servo darf dann nicht tackern. Tut er es, soweit zurückdrehen, bis das Tackern verschwindet.

Zur nächsten Einstellung gelangt man, indem der Poti „Hub“ kurz auf Rechtsanschlag gedreht und wieder zur Mitte bewegt wird. Die Ätzmaschine piepst einmal bei Rechtsanschlag und noch einmal, wenn der Rechtsanschlag verlassen wurde.

Es werden so die Parameter in der folgenden Reihenfolge durchlaufen:

Servo Rechtsanschlag
Servo Linksanschlag und
Wippe Mitte

Die Wippe ist auf Mittelstellung einzustellen. Durch diese Einstellung können mechanische Ungenauigkeiten des Servo-Hebels ausgeglichen werden. Die Wipp-Platte ist dann waagrecht.

3.3 Service-Mode verlassen

Sind alle drei Einstellungen durchgeführt. Wird der Poti „Hub“ ein letztes Mal auf Rechtsanschlag gedreht. Die Ätzmaschine piepst einmal.

Die Ätzmaschine beginnt nun direkt mit ihrer Arbeit und wippt je nach Einstellung von „Hub“ und „Speed“ unterschiedlich stark und schnell.

4 Das serielle Interface RS232

Das serielle Interface ist dazu gedacht, Parameter einzustellen, die über den Service-Mode schlecht einzustellen sind.

Für eine Verbindung mit dem PC wird ein Standard-RS232 Kabel verwendet. Hyper-Terminal (hypertrm.exe) von Windows oder andere Terminal-Programme dienen zur Konfiguration der Ätzmaschine. Siehe auch Absatz 6 Hyper-Terminal. Bei Verwendung eines Bootloaders kann die Firmware auch aktualisiert werden, ohne das Gerät öffnen zu müssen.

4.1 RS232-Parameter

Es sind folgende Schnittstellenparameter einzustellen:

Baudrate:	19200
Datenbits:	8
Stopbits:	1
Parität:	keine
Handshake:	kein

4.2 Das RS232-Protokoll

Das Protokoll ist einfach und wird im ASCII-Format übertragen. Jede Anfrage / Rückgabewert wird mit Carriage Return [CR] abgeschlossen. Zahlenwerte werden in dezimaler Schreibweise angegeben. Die Rückgabe besteht bis auf die Ausnahme der Sonderkommandos immer aus einer (1) Zeile.

Schreiben:

Um ein Kommando auszuführen oder ein Parameter an die Ätzmaschine zu senden, wird ein großes ‚P‘ gefolgt von der Parameter-Nummer, Doppelpunkt als Trennzeichen und dem Parameter gesendet. Das Kommando darf keine Leerzeichen (Space) enthalten und schließt mit [CR] ab. Die Ätzmaschine sendet keine Antwort zurück. Ausnahme ist, wenn ein Fehler auftritt. Dies ist der Fall, wenn eine ungültige Parameter-Nummer oder ungültiger Parameter gesendet wird

Allgemeiner Syntax Schreiben: **P<Param-Nummer>[:<Parameter>]**

Der zu übergebende Parameter ist von der Parameter-Nummer abhängig.

Lesen:

Um ein Parameter auszulesen, wird ein kleines ‚p‘ gefolgt von der Parameter-Nummer gesendet. Das gesendete Kommando darf keine Leerzeichen (Space) enthalten und schließt mit [CR] ab.

Allgemeiner Syntax Lesen: **p<Param-Nummer>**

Parameter werden beim Lesen nicht übergeben. Der Antwortstring ist von variabler Länge, enthält keine Leerzeichen (Space) und schließt mit Carriage Return [CR] ab. Zurückgegeben wird direkt der angefragte Parameter. Im Folgetext werden die Parameter ausführlich beschrieben.

4.3 Kommandos

Die folgenden Parameter werden über die Schnittstelle unterstützt:

Parameter-Nr.	Zugriff	Bereich	Beschreibung
1	R	1	Reset ausführen
2	W	1, 2	Daten im EEPROM sichern
3	R / W	0, 1	Schreiben: Schaltet den Debug-Modus ein (1) oder aus (0) Lesen: Gibt die internen Flags als Hex-Wert aus (Byte)
4	R / W	25 - 65536	Setzt einen Wert für Debug-Intervall oder gibt diesen zurück.
10	R	0 - 1023	ADC-Wert des Poti „Speed“
11	R	0 - 511	½ ADC-Wert des Poti „Hub“
12	R / W	0 - 1023	Setzt den ADC-Wert für Soll-Temperatur oder gibt den Wert zurück.
13	R	0 - 1023	Gibt den ADC-Wert für Ist-Wert zurück.
20	R / W	0 - 1023	Servo-Begrenzung Links-Anschlag
21	R / W	0 - 1023	Servo-Begrenzung Rechts-Anschlag
22	R / W	0 - 1023	Servo-Mittelstellung
23	R	0 - 1023	Aktuelle Servo-Position
30	R / W	0 - 255	Begrenzung der Heizleistung

Tabelle 4.1 - Kommandoübersicht

Parameter, die gelesen werden können sind mit Zugriff ‚R‘ gekennzeichnet. Beschreibbare Parameter sind mit Zugriff ‚W‘ gekennzeichnet.

Im Folgenden sind alle Parameter ausführlich beschrieben.

4.3.1 Reset (Kommando 1)

Syntax: p1[CR]

Dieses Kommando führt einen Restart im Controller aus. Dies kann nützlich sein, um Softwaregesteuert einen Reset z.B. für Firmware-Updates durchzuführen. Auch im Debug-Modus kann diese Option nützlich sein.

Zugriff: nur Lesen
Parameter-Nummer: 1
Rückgabewert: keine

Anmerkung:

Um Firmware-Updates per serieller Schnittstelle durchzuführen, ist ein Bootloader erforderlich. Firmware-Updates können auch direkt über die ISP-Schnittstelle in den Flash-Speicher geschrieben werden. Hierfür ist dann ein geeigneter Programmierer erforderlich.

4.3.2 EEPROM (Kommando 2)

Syntax: P2:<Parameter>[CR]

Mit diesem Kommando können Einstellungen dauerhaft gesichert oder Default-Werte im EEPROM hergestellt werden.

Zugriff:	nur Schreiben
Parameter-Nummer:	2
Parameter:	1 2

Anmerkung:

Wird als Parameter ‚1‘ übergeben, wird das EEPROM auf definierte Standardwerte zurückgesetzt und anschließend ein Restart durchgeführt. Dieses Kommando könnte man mit *Werkseinstellung* definieren.

Wird als Parameter ‚2‘ übergeben, werden Einstellungen im EEPROM gesichert und anschließend ein Restart durchgeführt. Die gespeicherten Werte sind Servo-Begrenzungen und Servo-Mitte, Temperatur-Einstellung und Heizleistung.

4.3.3 Debug-Modus (Kommando 3)

Syntax: p3[CR] oder
P3:<Parameter>[CR]

Der Debug-Modus kann eingeschaltet und ausgeschaltet werden. Bei aktiviertem Debug-Modus werden die internen Daten regelmäßig über die serielle Schnittstelle ausgegeben. Die Intervallzeit zur Ausgabe der Debug-Daten beträgt 0,5 Sekunden.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	3
Parameter:	0 1
Rückgabe:	0 - 255 (Hex-Byte)

Anmerkung:

Lesen ohne Parameter gibt ein Rückgabewert im Bereich von 0 bis 255 und repräsentiert die intern verwendeten Flags. Der Rückgabewert ist im Format Hex-Byte. Gelesen werden sollte das Flag-Byte nur, wenn der Debug-Modus deaktiviert ist.

Wird als Parameter (schreiben) eine ‚1‘ übergeben, wird der Debug-Modus aktiviert.

Wird als Parameter (schreiben) eine ‚0‘ übergeben, wird der Debug-Modus wieder deaktiviert.

Die Einstellung des Debug-Modus wird nicht im EEPROM gesichert. Ein Restart bewirkt somit wieder einen „normalen“ Betrieb der Ätzmaschine.

Weitere Hinweise zum Debug-Modus sind im Abschnitt „5 Der Debug-Modus“ beschrieben.

4.3.4 Debug-Interval (Kommando 4)

Syntax: P4:<Parameter>[CR] oder
P4[CR]

Im Debug-Modus werden regelmäßig interne Daten der Ätzmaschine über die serielle Schnittstelle versendet. Mit diesem Kommando kann der eingestellte Intervall-Wert ermittelt und geändert werden. Weitere Infos können im Absatz **5 - Der Debug-Modus** nachgelesen werden.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	4
Parameter:	0 – 65536 (Word) [in 0,02s Ticks]
Rückgabe:	0 – 65536 (Word) [in 0,02s Ticks]

Anmerkung:

Als Parameter können ganze Zahlenwerte verwendet werden. Der Wert gibt die Anzahl Ticks an, die vergehen, bis ein Debug-Datensatz erneut gesendet wird. Ein Tick dauert exakt 0,02 Sekunden. Wenn beispielsweise alle 30 Sekunden ein Datensatz gesendet werden soll, ist als Parameter 30s / 0,02s -1 einzugeben. Also 1500-1 Ticks. Der Wert ,1' muss abgezogen werden, da die Ätzmaschine intern bis ,0' hinunter zählt.

4.3.5 Speed (Kommando 10)

Syntax: p10[CR] oder

Mit diesem Kommando kann die aktuelle Speed-Einstellung ausgelesen werden. Der zurück gelieferte Wert ist vom Typ Word (0 – 1023) und repräsentiert den ADC-Wert des Speed-Poti.

Zugriff:	nur Lesen
Parameter-Nummer:	10
Rückgabe:	0 – 1023 (Word)

Anmerkung:

Linksanschlag des Poti entspricht dem ADC-Wert 1023, Rechtsanschlag dem Wert 0.

4.3.6 Hub (Kommando 11)

Syntax: p11[CR]

Mit diesem Kommando kann die aktuelle Hub-Einstellung ausgelesen werden. Der Zurückgelieferte Wert ist vom Typ Word (0 – 511) und repräsentiert den ADC-Wert des Hub-Poti.

Zugriff:	nur Lesen
Parameter-Nummer:	11
Parameter:	0 – 511 (Word)

Anmerkung:

Linksanschlag des Poti entspricht dem ADC-Wert 0, Rechtsanschlag dem Wert 511. Dies entspricht dem ½ ADC-Wert. Bei Werten<15 wird die Heizungsbegrenzung auf 0% gesetzt.

4.3.7 Soll-Temperatur (Kommando 12)

Syntax: p12[CR] oder
P12:<Parameter>[CR]

Mit diesem Kommando kann die Einstellung der Soll-Temperatur ausgelesen oder verändert werden. Der Parameter und der Rückgabewert sind vom Typ Word (0 – 1023). Der Wert selbst repräsentiert **nicht** die Temperatur in °C, sondern den ADC-Wert des Temperatur-Vergleichs-Wertes. Beim Schreiben wird der Soll-Wert (ADC-Wert) gesetzt.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	12
Parameter:	0 – 1023 (Word)
Rückgabe:	0 – 1023 (Word)

Anmerkung:

Wird die Solltemperatur gesetzt, wird keine Antwort zurück gesendet. Der gesetzte Wert muss bei Bedarf explizit zurück gelesen werden.

4.3.8 Ist-Temperatur (Kommando 13)

Syntax: p13[CR]

Die Ist-Temperatur kann nur gelesen werden und repräsentiert die Temperatur am Fühler (KTY) als ADC-Wert. Der Rückgabewert ist vom Typ Word (0 – 1023).

Zugriff:	nur Lesen
Parameter-Nummer:	13
Parameter:	keine
Rückgabe:	0 – 1023 (Word)

Anmerkung:

Die Rückgabewerte können theoretisch die Werte im Bereich von 0 bis 1023 erreichen, da die Werte direkt vom AD-Wandler stammen. Praktisch ist am ADC-Eingang jedoch ein Spannungsteiler aus Temperatur-Sensor und Widerstand. Deshalb werden die Werte im Bereich von ca. 450 bis 589 liegen, was den Temperaturen von ca. +10°C bis +80°C entspricht. Durch Bauteiltoleranzen ist mit kleinen Abweichungen zu rechnen. Es empfiehlt sich, die Temperatur mit einem Thermometer einmalig zu justieren.

Die optimale Ätzbad-Temperatur sollte zwischen 40°C und 50°C liegen. Es sind also ca. 45°C als Mitte anzustreben.

4.3.9 Servo-Begrenzung Links (Kommando 20)

Syntax: p20[CR] oder
P20:<Parameter>

Dieses Kommando ermöglicht die Einstellung für den linken Servoanschlag auszulesen oder zu verändern. Diese Einstellung kann auch im Service-Mode durchgeführt werden.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	20
Parameter:	1000-1500 [μ s]
Rückgabe:	1000-1500 [μ s]

Anmerkung:

Die Servoposition wird in Mikro-Sekunden [μ s] angegeben. Die Werte müssen beim Setzen zwischen 1000 und 1500 liegen mit Tendenz gegen 1000.

4.3.10 Servo-Begrenzung Rechts (Kommando 21)

Syntax: p21[CR] oder
P21:<Parameter>[CR]

Dieses Kommando ermöglicht die Einstellung für den rechten Servoanschlag auszulesen oder zu verändern. Diese Einstellung kann auch im Service-Mode durchgeführt werden.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	21
Parameter:	1500-2000 [μ s]
Rückgabe:	1500-2000 [μ s]

Anmerkung:

Die Servoposition wird in Mikro-Sekunden [μ s] angegeben. Die Werte müssen beim Setzen zwischen 1500 und 2000 liegen mit Tendenz gegen 2000.

4.3.11 Servo-Mittelstellung (Kommando 22)

Syntax: p22[CR] oder
P22:<Parameter>[CR]

Dieses Kommando ermöglicht die Einstellung der Mittelstellung des Servo auszulesen oder zu verändern. Diese Einstellung kann auch im Service-Mode durchgeführt werden.

Zugriff:	Lesen / Schreiben
Parameter-Nummer:	22
Parameter:	1000-2000 [μ s]
Rückgabe:	1000-2000 [μ s]

Anmerkung:

Die Position der Servo-Mittelstellung (Wippe waagrecht) wird in Mikro-Sekunden [μ s] angegeben. Beim Setzen der Servo-Mitte sind Werte um die $1500^{\pm 100}$ μ s ideal. Bei stärkeren Abweichungen sollte die Mechanik entsprechend nachjustiert werden.

4.3.12 Servo-Position (Kommando 23)

Syntax: p23[CR]

Dieses Kommando ermöglicht es, die aktuelle Servostellung auszulesen.

Zugriff:	Lesen
Parameter-Nummer:	23
Parameter:	keine
Rückgabe:	1000-2000 [μ s]

Anmerkung:

Zum Zeitpunkt der Auslesung des Parameters, wird die gerade angefahrne Servo-Position zurückgegeben.

4.3.13 Heizleistung (Kommando 30)

Syntax: p30[CR] oder
P30:<Parameter>[CR]

Das angeschlossene Heizelement wird per PWM mit ca. 0,5Hz angesteuert und erhält direkt die eingespeiste Versorgungs-Spannung. Wenn ein 12V-Heizelement verwendet wird und die Versorgungs-Spannung ebenfalls 12V beträgt, gibt es keine Probleme. Wenn jedoch ein 12V-Heizelement mit 24V Versorgungs-Spannung betrieben wird, könnte dieses durch Überhitzung zerstört werden. Zu beachten ist auch, dass das Heizelement die aufgelegte Ätzschale zum Schmelzen bringen kann. Um dies zu verhindern kann die Heizleistung begrenzt werden.

Zugriff:	Lesen und Schreiben
Parameter-Nummer:	30
Parameter:	0 – 100 (Byte)
Rückgabe:	0 – 100 (Byte)

Anmerkung:

Eine eingestellte Heizleistung von 100% bedeutet, dass die resultierende Leistung aus der Betriebsspannung und dem verwendeten Heizelement zu 100% durchgelassen wird.

Beispiel:

Sie möchten ein Heizelement mit 30W Leistung verwenden. Das Heizelement hat eine Nennspannung von 15V. Die Betriebsspannung soll angenommen 20V betragen. Welcher Wert ist nun für die Heizungs-Leistungsbegrenzung einzustellen?

Lösung:

Zuerst ist die resultierende Leistung bei Nennspannung $U=15V$ zu ermitteln. Dazu wird die folgende Formel benötigt:

$$P = \frac{U^2}{R} \quad \text{Bzw. umgestellt} \quad R = \frac{U^2}{P}$$

Es wird zunächst der Widerstand des Heizelements bei Nennspannung ermittelt:

$$R = \frac{U^2}{P} = \frac{15V^2}{30W} = 7.5\Omega$$

Der Widerstand ändert sich nicht, aber die Spannung für das Heizelement. Daraus folgt die resultierende Leistung bei 20V:

$$P = \frac{U^2}{R} = \frac{20V^2}{7.5\Omega} = 53.3W$$

Nun ist nur noch der prozentuale Anteil zu bestimmen, um die 30W bei 20V nicht zu übersteigen. Hierzu die folgende Formel, die den einzustellenden Prozentwert ausgibt:

$$\text{Leistung}[\%] = \frac{P_{\text{Nenn}}}{P_{\text{Real}}} \cdot 100 = \frac{30W}{53.3W} \cdot 100 = 56,28\%$$

Es ist somit die Heizungsbegrenzung auf 56% einzustellen.

4.3.14 Sonder-Kommandos (Hilfe und Versions-Info)

Es gibt noch zwei weitere Kommandos, die nicht in das bisherige Protokoll-Schema der vorangegangenen Beschreibungen passt. Hierbei handelt es sich nicht um Abfragen von Parameter, sondern um eine abrufbare Zusatzinformation.

Es handelt sich um das Abfragen der Firmware-Version mit Kompilier-Datum der Firmware und um eine Hilfestellung bei der Parametrierung mit einem Terminal-Programm.

Diese beiden Sonder-Kommandos werden im Folgenden beschrieben.

4.3.14.1 Versions-Info (Kommando ,?')

Syntax: ?[CR]

Wird ein ,?'-Zeichen (Fragezeichen) an die Ätzmaschine gesendet, wird die folgende Information zurückgegeben:

- Copyright
- Version der Firmware und
- Kompilier-Datum und Zeit der Firmware
- Der Chip als Zifferncode

Eine Anfrage mit dem Terminal-Programm ergibt folgende Antwort:

```
Ätzmaschine

Copyright: M. Ruhwald 2008
Version:   2.0
Compiled:  09-07-2009 21:46:45
MCU:       17 (ATmega8)

Press 'h' and Enter for Help
```

Nach dem Copyright wird hier die Version 2.0 angezeigt. Anschließend folgt das Kompilier-Datum. Bei der Firmware Version 2.2 ist dies der 9.7.2009 um Uhr 21:46. MCU zeigt den verwendeten Controller an. Der Chipcode ,17' wird vom Hersteller des Controllers vergeben und zeigt einen ATmega8 Controller an.

4.3.14.2 Hilfe (Kommando ,h')

Syntax: h[CR]

Das Hilfe-Kommando ,h' sendet einen mehrzeiligen String zurück, der alle Kommandos auflistet, die von der Ätzmaschine unterstützt werden. Bei der Version 2.0 ergibt sich folgende Antwort:

```
Hilfe-Ätzmaschine
Syntax: P|p<ParaNum>[:<Value>]<CR>

P = Parameter Setzen ; p = Parameter lesen
<ParaNum> = Parameter-Nummer ; <Value> = zu setzender Wert

 1 - Reset
 2 - EEPROM: Value=1: Defaults setzen / Value=2: Save
 3 - Debugger an/aus
 4 - Debug-Interval
10 - ADC-Speed
11 - ADC-Hub
12 - ADC-Temperatur Soll
13 - ADC-Temperatur Ist
20 - Servo-Min
21 - Servo-Max
22 - Servo-Mitte
23 - Servo-Position
30 - Heizung-Power-Max
31 - Heizung-Ticks-Max
Beispiel: p20<CR>      - liest Servo-Min
Beispiel: P12:560<CR> - setzt neue Soll-Temperatur
```

Nach dem Text „Hilfe Ätzmaschine“ wird der Syntax angezeigt, wie die Kommandos und Parameter einzugeben sind. In der Auflistung ist links die Parameter-Nummer angegeben und daneben die Funktion, die sich hinter diesem Parameter verbirgt. Angefügt sind noch 2 Beispiele.

5 Der Debug-Modus

Der Debug-Modus wird mit dem Kommando 3 aktiviert und deaktiviert:

Kommando: P3:0|1

Bei aktiviertem Debug-Modus (Parameter=1) sollten keine Anfragen an die Ätzmaschine durchgeführt werden. Grund ist, der Debug-Modus sendet alle 0,5 Sekunden alle internen Daten im CSV-Format. Eine Anfrage zwischendurch würde das Format durcheinander bringen.

Über den Debug-Modus ist es möglich, die internen Daten zu loggen. Dies kann beispielsweise mit dem Terminal-Programm Hyperterm, welches zum Windows Betriebssystem gehört, durchgeführt werden. Da die geloggten Daten im CSV-Format vorliegen, können diese z.B. in Excel importiert und weiter verarbeitet werden. So lassen sich beispielsweise der Temperaturverlauf des Ätzbades und die dazu gehörende Heizleistung mit Soll- und Ist-Temperatur grafisch als Diagramm darstellen.

Die Daten werden als CSV-Format in folgender Reihenfolge ausgegeben.

Position	Parameter	Bereich	Bemerkung
0	Count	[Long]	Nr. des Datenpakets
1	ADC-Speed	0-1023 [Word]	Poti Speed
2	ADC-Hub	0-511 [Word]	Poti Hub (intern umgerechnet)
3	Temp. Sollwert	0-1023 [Word]	ADC-Wert Temperatur Sollwert
4	Temp. Istwert	0-1023 [Word]	ADC-Wert Temperatur von KTY-Sensor
5	Servo-Min	1000-1500 [µs]	Linke Anschlagbegrenzung Servo
6	Servo-Max	1500-2000 [µs]	Rechte Anschlagbegrenzung Servo
7	Servo-Mitte	1000-2000 [µs]	Mittelstellung Servo
8	Servo-Position	1000-2000 [µs]	Aktuelle Servo-Position
9	Heizung-Power-Max	0-100 [%]	Maximale Heizleistung (Begrenzung der Heizleistung)
10	Heizung-Power	0-100 [%]	Aktuelle Heizleistung (Reale Heizleistung)
11	P-Anteil	0-100	Proportional-Anteil des PI-Reglers
12	I-Anteil	0-100	Integral-Anteil des PI-Reglers

Tabelle 5.1 - Debug-Daten Reihenfolge

6 Hyper-Terminal

Das Windows Betriebssystem enthält von Haus aus ein Terminal-Programm. Es nennt sich **Hyper-Terminal** und kann zur Konfiguration der Ätzmaschine verwendet werden.

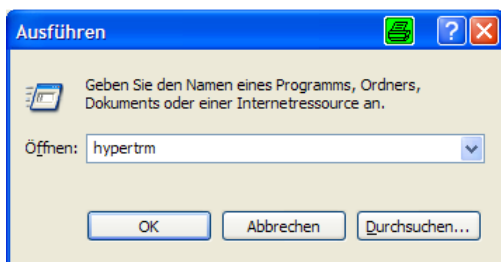


Abbildung 6.1 – Start Hyperterm

Zum Starten der Terminal-Software wählen Sie in der Windows Taskleiste **Start -> Ausführen**. Es Öffnet sich ein kleines Fenster wie in Abbildung 6.1 dargestellt. Tragen Sie im Feld „Öffnen“ den Programmname „hypertrm“ ein und klicken Sie auf OK.

Windows startet nun das Programm Hyperterm. Um es mit der Ätzmaschine verwenden zu können, muss es noch konfiguriert werden.

6.1 Terminalprogramm einrichten

Wenn HyperTerminal bereits einmal eingerichtet und die Einstellung gespeichert wurde, kann diese Einstellung geladen werden. Wählen Sie Menü Datei -> Öffnen den Eintrag „Ätzmaschine.ht“ aus.



Abbildung 6.2 – Neue Verbindung

Besteht die Datei noch nicht, muss sie zuerst angelegt werden. Rufen Sie im Menü den Eintrag **Datei -> Neue Verbindung** auf. Es öffnet sich ein Dialog wie in Abbildung 6.2 dargestellt. Geben Sie im Eingabefeld „Name“ der einzurichtenden Verbindung den Name „Ätzmaschine“ und wählen dann OK an.

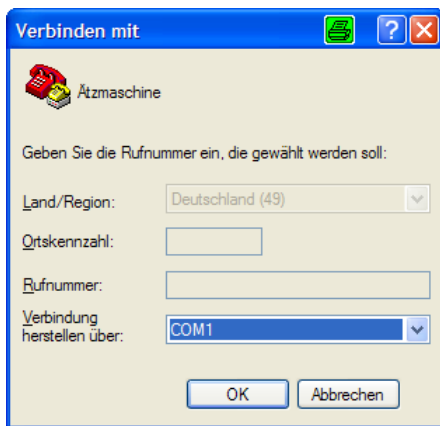


Abbildung 6.3 - Schnittstelle

Es erscheint ein neuer Dialog, in dem die zu verwendende Verbindung anzugeben ist. In unserem Fall wollen wir über COM1 eine Verbindung herstellen. Wenn USB-Serial-Adapter verwendet werden, können auch emulierte COM-Ports angewählt werden.

Nachdem die Einstellung durchgeführt wurde, wird der Dialog mit OK geschlossen. Es öffnet sich ein dritter Dialog.

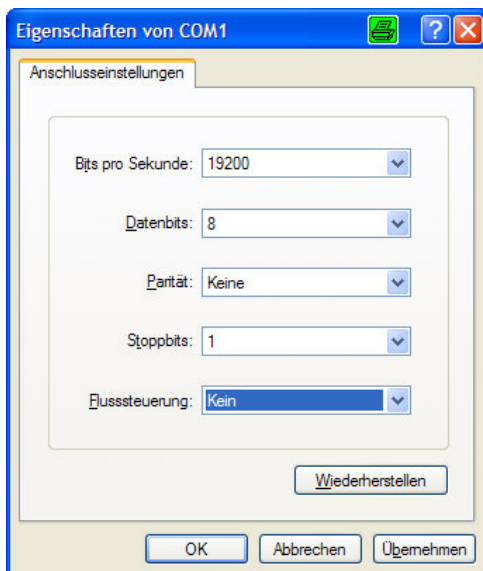


Abbildung 6.4 - Eigenschaften

Im dritten Dialog sind die Anschlusseinstellungen zu ändern.

Alle Einstellungen sind wie in Abbildung 6.4 dargestellt durchzuführen.

Schließen Sie den Dialog mit OK.

Das Terminal-Programm öffnet nun die Verbindung und ist bereit, um Daten zu senden und zu empfangen.

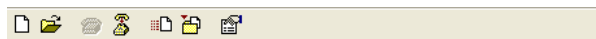


Abbildung 6.5 – Symbolleiste Hyper-Terminal

Die Konfiguration ist jedoch noch nicht ganz abgeschlossen. Wie in Abbildung 6.5 an dem Symbol mit dem angenommenen Telefonhörer zu erkennen ist, ist eine Verbindung geöffnet. Schließen Sie zunächst die Verbindung, indem Sie dieses Symbol anklicken. Bei getrennter Verbindung wird das Symbol mit dem aufgelegten Telefonhörer aktiviert.

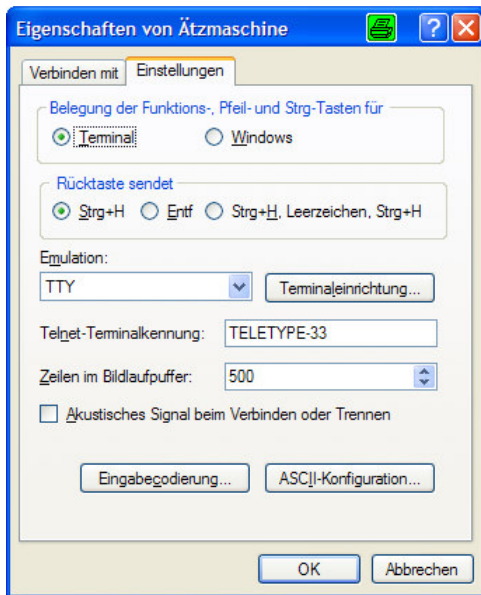


Abbildung 6.6 - Eigenschaften

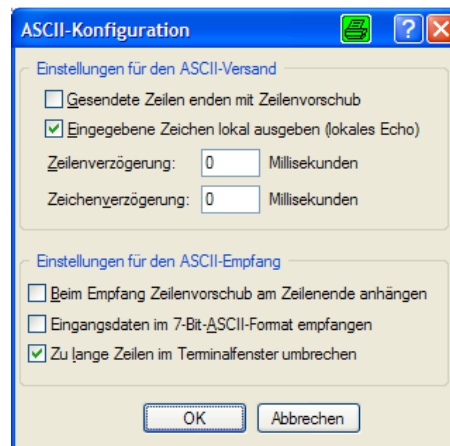
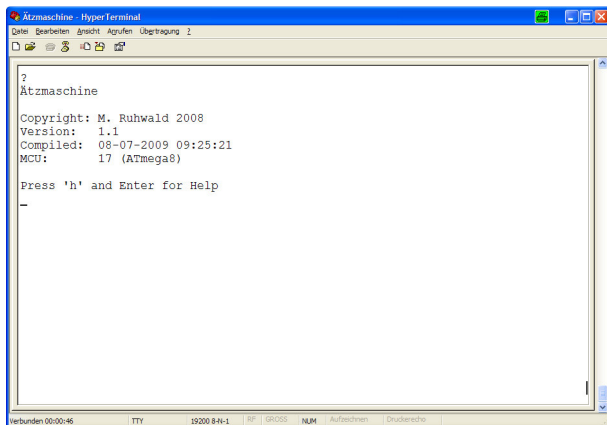


Abbildung 6.7 – ASCII Konfiguration

Wählen Sie im Menü den Eintrag Datei -> **Eigenschaften** aus. Es wird der Dialog (siehe Abbildung 6.6) aufgerufen. Wählen Sie den Karteireiter „Einstellungen“ aus.

Stellen Sie Emulation auf „TTY“ um. Bevor Sie den Dialog schließen, rufen Sie noch „ASCII-Konfiguration“ auf. Stellen Sie dort die Optionen wie in Abbildung 6.7 dargestellt ein. Schließen Sie die Dialoge „ASCII-Konfiguration“ und „Eigenschaften von Ätzmaschine“ mit OK. Die Einstellung dann nicht vergessen zu speichern.

Öffnen Sie nun die Verbindung um einen kleinen Test durchzuführen. Klicken Sie das Symbol „Anrufen“ in der Symbolleiste an. – Die Verbindung wird geöffnet.

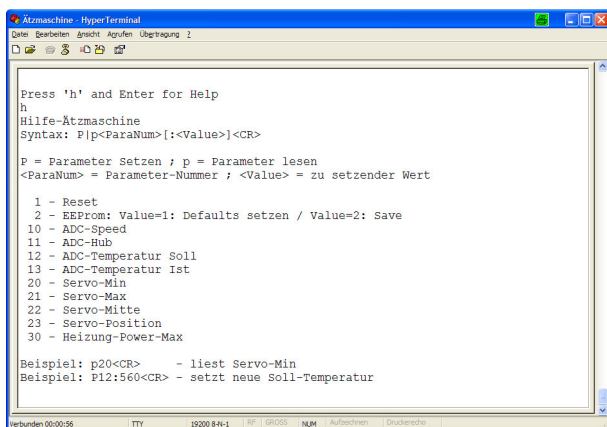


```
Ätzmaschine - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung 2
?
Ätzmaschine
Copyright: M. Ruhwald 2008
Version: 1.1
Compiled: 08-07-2009 09:25:21
MCU: 17 (ATmega8)
Press 'h' and Enter for Help
-
```

Abbildung 6.8 – Versions-Info

Geben Sie ein Fragezeichen (,?') ein und drücken Return.

Es wird die Versionsinfo zur Firmware ausgegeben, wie in Abbildung 6.8 dargestellt.



```
Ätzmaschine - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung 2
Press 'h' and Enter for Help
h
Hilfe-Ätzmaschine
Syntax: P|p<ParaNum>[:<Value>]<CR>
P = Parameter Setzen ; p = Parameter lesen
<ParaNum> = Parameter-Nummer ; <Value> = zu setzender Wert
1 - Reset
2 - EEPROM: Value=1: Defaults setzen / Value=2: Save
10 - ADC-Speed
11 - ADC-Hub
12 - ADC-Temperatur Soll
13 - ADC-Temperatur Ist
20 - Servo-Min
21 - Servo-Max
22 - Servo-Mitte
23 - Servo-Position
30 - Heizung-Power-Max
Beispiel: p20<CR> - liest Servo-Min
Beispiel: P12:560<CR> - setzt neue Soll-Temperatur
```

Abbildung 6.9 – Hilfe der Ätzmaschine

Welche Befehle bzw. Kommandos von der Ätzmaschine unterstützt werden kann mit der Hilfe-Funktion in Erfahrung gebracht werden.

Geben Sie den Kleinbuchstaben ‚h‘ ein und drücken Enter.

Es wird eine Syntax-Beschreibung mit einer Liste der Parameter-Nummern ausgegeben. Zwei Beispiele sind noch beigefügt.

Anmerkung:

Eine genaue Beschreibung der einzelnen Parameter und Kommandos, sowie deren Verwendung ist im Abschnitt **4.3 Kommandos** beschrieben.

7 Firmware und Bootloader

Die Firmware stellt die so genannte Intelligenz des Controllers dar. Sie ist die Software, die den Controller mit seiner Peripherie zur Ätzmaschine macht.

Um die Firmware in den Controller (ATmega8) einzuspielen, muss zuerst ein Bootloader vorhanden sein. Der Bootloader sorgt dafür, dass die Firmware über die serielle Schnittstelle in den Controller eingespielt werden kann. Er stellt den ersten Programmteil im Controller dar, der nach einem Reset als erstes ausgeführt wird.

7.1 Info zum Bootloader

Die Ätzmaschine ist mit einem Bootloader ausgestattet. Dadurch kann die Firmware über die serielle Schnittstelle aktualisiert werden, ohne das Gerät öffnen zu müssen. Der Bootloader muss nur bei der aller ersten Programmierung des Controllers eingespielt werden.

7.1.1 Programmieradapter

Wenn der Bootloader bereits eingespielt wurde, kann dieser Teil der Anleitung übersprungen werden. Die Anleitung zum Einspielen bzw. Aktualisierung der Firmware ist im Absatz 0 nachzulesen.

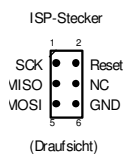


Abbildung 7.1 - ISP

Um den Bootloader einzuspielen oder zu aktualisieren ist es erforderlich, die ISP-Programmier-Schnittstelle über einen SP12-Programmieradapter mit dem PC zu verbinden. Der Bootloader kann nicht über die serielle Schnittstelle aktualisiert werden.

In Abbildung 7.1 ist die Pinbelegung (Draufsicht) der ISP-Schnittstelle abgebildet. Sie befindet sich auf der Steuerplatine.

7.1.2 Programmer wählen

Je nach verwendetem Programmier-Adapter muss ein entsprechender Programmer ausgewählt werden.

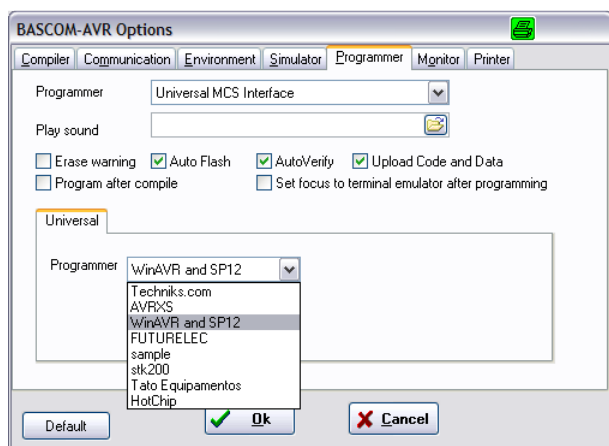


Abbildung 7.2 - Programmer auswählen

Bei Verwendung des SP12-Programmier-Adapters, wird dieser mit der parallelen Schnittstelle (LPT1) verbunden. In Bascom-AVR wird im Menü unter „Optionen –

Programmer“ die Programmier-Software „Universal MCS Interface“ wie in Abbildung 7.2 aufgezeigt ausgewählt und eingestellt. Das Protokoll der Programmierung wird über die Unterkategorie des Programmers eingestellt. Hier wird „WinAVR and SP12“ ausgewählt. Die Eingabe wird mit Klick auf OK abgeschlossen.

7.1.3 Fusebits am Controller setzen

Bevor nun der Bootloader eingespielt werden kann, müssen im Controller noch diverse Fusebits gesetzt werden. Das Programmierkabel an BU4 anschließen.

☐ Chip	
Name	MEGA8
Calibration 0	A5
Calibration 1	A3
Calibration 2	9F
Calibration 3	A0
☐ Lockbits	
Lockbit 65	11:No restrictions for SPM or LPM accessing the boot loader section
Lockbit 43	11:No restrictions for SPM or LPM accessing the application section
Lockbit 21	11:No memory lock features enabled for parallel and serial programming
☐ Fusebits	
Fusebits	24
Fusebit C	0:BODLEVEL 4.0V
Fusebit B	0:BODEN enabled
Fusebit KLA987	100100:Int. RC Osc. 8 MHz; Start-up time: 6 CK + 64 ms; [CKSEL=0100 SUT=10]
☐ Fusebits High	
Fusebits High	D0
Fusebit High M	1:PIN PC6 is RESET
Fusebit High J	1:WDT enabled by WDTCSR
Fusebit High I	0:SPI enabled
Fusebit High H	1:CKOPT 1
Fusebit High G	0:Preserve EEPROM when chip erase
Fusebit High FE	00:1024 words boot size, C00
Fusebit High D	0:Reset vector is boot loader reset

Abbildung 7.3 - Fusebits / Lockbits

Die obige Abbildung 7.3 zeigt die korrekten Einstellungen des Controllers. Ein besonderes Augenmerk gilt den *Fusebits* und den *Fusebits High*.

„Fusebit C“ setzt den Bootlevel auf 4V.

„Fusebit KLA987“ wird auf externen Quarz eingestellt, der den Controller taktet.

„Fusebit High G“ das EEPROM soll bei Neuprogrammierung nicht gelöscht werden.

„Fusebit High FE“ gibt die Bootloader-Größe an.

„Fusebit High D“ startet den Bootloader.

Sind alle Änderungen durchgeführt, müssen die Fusebits im Controller gesetzt werden. Hierzu klickt man auf die aktiven Buttons *Write FS* bzw. *Write FSH*.

Bitte nicht vergessen:

Nach Upload des Bootloader muss eine Einstellung noch mal geändert werden. Später dazu mehr.

7.1.4 Bootloader einspielen

Zum Einspielen des Bootloader muss das Programmierkabel am ISP-Stecker angeschlossen werden.

Sie können einen Programmierer ihrer Wahl verwenden. Wenn Sie jedoch Bascom AVR verwenden möchten, gehen sie wie folgt vor.

Starten Sie die AVR-IDE Bascom AVR. Legen Sie ein neues File an (Datei -> Neu). Diese Datei muss nicht beschrieben werden und auch nicht abgespeichert werden. Sie hat nur den Zweck, dass die Symbolleiste aktiviert wird.

Wählen Sie im Menü den Eintrag **Programmieren -> Zum Chip senden -> Manuell programmieren** aus. Es tauchen nun mehrere Fehlermeldungen auf, die getrost ignoriert werden dürfen. Schließlich öffnet sich ein Dialog wie in Abbildung 7.4

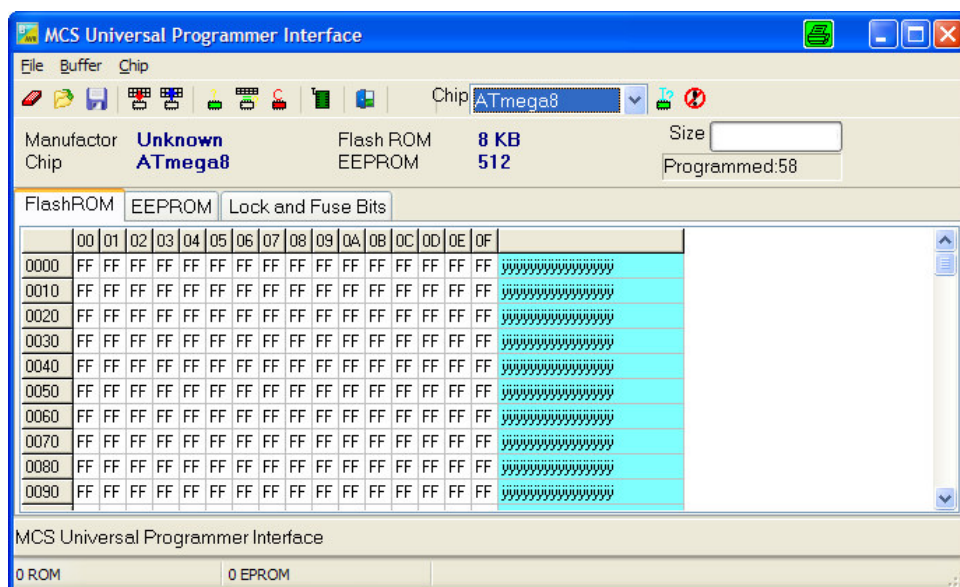


Abbildung 7.4 – Manuell Programmieren

Im Eingabefeld Chip ist nun der Eintrag ATmega8 auszuwählen und der Karteireiter auf „FlashROM“ einzustellen. Diese Einstellung ist wichtig!

Nun wird in der Symbolleiste das Symbol „Load File into Buffer“ ausgewählt. Siehe Abbildung 7.5.

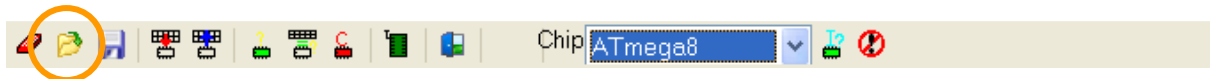


Abbildung 7.5 - Programmer Symbolleiste

Nun kann die Datei ausgewählt werden, die in den Controller geschrieben werden soll.

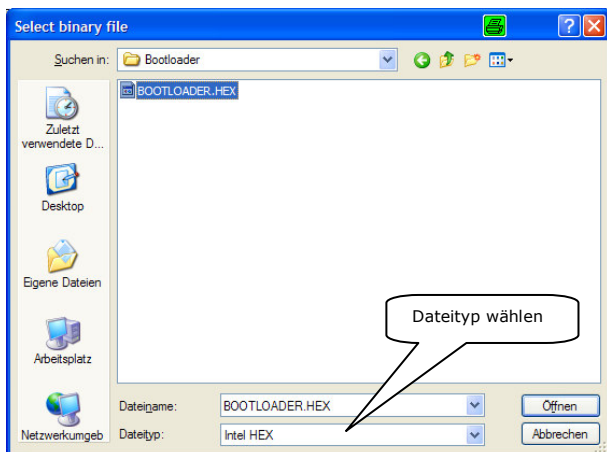


Abbildung 7.6 - Bootloader auswählen

Hier kann wahlweise das BIN-File oder das HEX-File geladen werden.

Wählen Sie die Bootloader-Datei an, um sie in den Buffer zu laden.

Mit Öffnen wird der Dialog geschlossen.

Jetzt befindet sich der Bootloader in Buffer (IDE) und muss nur noch in den Controller geschrieben werden.

Zum Schreiben nun in der Symbolleiste das Symbol „**Write Buffer to FlashROM**“ anwählen (Das Symbol mit rotem Pfeil). Der Flashvorgang beginnt und der Bootloader wird in den Controller geschrieben.

Das Programm (Bootloader) im Controller beginnt augenblicklich zu laufen und wartet auf die Firmware, die noch eingespielt werden muss. Dies wird im folgenden Abschnitt 7.2 beschrieben.

Zur Erinnerung:

Lockbit65 nach Upload des Bootloader umstellen wie hier angegeben:

Chip	
Name	MEGA8
Calibration 0	A5
Calibration 1	A3
Calibration 2	9F
Calibration 3	A0
Lockbits	
Lockbit 65	10:SPM is not allowed to write to the boot loader section
Lockbit 43	11:No restrictions for SPM or LPM accessing the application section
Lockbit 21	11:No memory lock features enabled for parallel and serial programming
Fusebits	
	24
Fusebit C	0:BODLEVEL 4.0V
Fusebit B	0:BODEN enabled
Fusebit KLA987	100100:Int. RC Osc. 8 MHz; Start-up time: 6 CK + 64 ms; [CKSEL=0100 SUT=10]
Fusebits High	
	D0
Fusebit High M	1:PIN PC6 is RESET
Fusebit High J	1:WDT enabled by WDTCR
Fusebit High I	0:SPI enabled
Fusebit High H	1:CKOPT 1
Fusebit High G	0:Preserve EEPROM when chip erase
Fusebit High FE	00:1024 words boot size, C00
Fusebit High D	0:Reset vector is boot loader reset

Abbildung 7.7 – Lockbit nach Bootloader Upload

Es muss noch das Lockbit65 auf den Wert 10 umgestellt werden. Diese Einstellung verhindert, dass bei Updates über die serielle Schnittstelle der Bootloader versehentlich überschrieben werden kann.

7.2 Firmware aktualisieren

Zur Aktualisierung der Firmware (nicht Bootloader) wird der MCS-Bootloader verwendet. Der Programmierer von Bascom ist wie in den folgenden Abbildungen einzustellen.

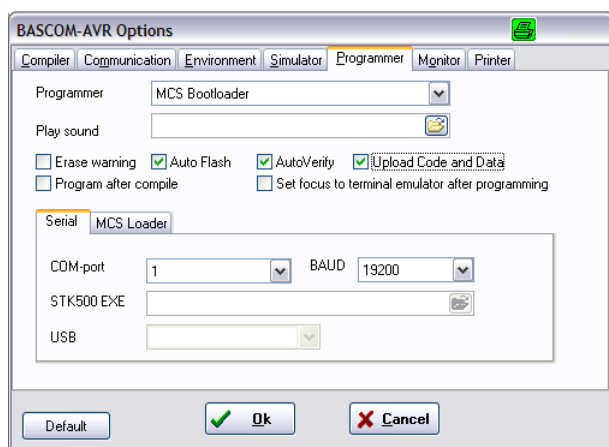


Abbildung 7.8 – MCS Bootloader (Serial)

Reiter Serial (Abbildung 7.8)

COM-Port einstellen, Angabe in der Abbildung kann abweichen. Baudrate 19200.

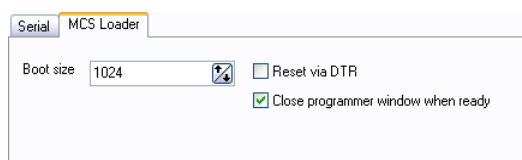


Abbildung 7.9 – MCS Bootloader (MCS Loader)

Reiter MCS Loader (Abbildung 7.9)

Bootsize auf 2048 einstellen. Reset via DTR deaktivieren.

Mit OK wird der Dialog verlassen.

Da per serieller Schnittstelle (Reset via DTR) nicht durchgeführt werden kann, muss der Reset „von Hand“ erfolgen.



Abbildung 7.10 – Symbolleiste (Programmiersymbol)

7.2.1 Programmierschritte

Folgende Steps sind der Reihe nach durchzuführen, um die Firmware per Bootloader einzuspielen:

1. Ziehen Sie die Stromversorgung der Ätzmaschine ab.
2. Klick auf das Programmieren-Symbol (siehe Abbildung 7.10)
3. Stromversorgung wieder einstecken.

Zwischen dem Klick auf Programmieren und dem Einstecken des Netzteils in die Ätzmaschine sollten nicht mehr als ca. 3 Sekunden liegen. Sollte die Programmierung misslingen, die Schritte 1 bis 3 wiederholen.

Bei erfolgreicher Programmierung meldet sich die Ätzmaschine durch einen kurzen Piep.

8 Anhang

8.1 Steckerbelegung Platine

Steckverbinder auf der Steuerungsplatine

LED-Stecker
Heizung (rot)



(Draufsicht)

Hier wird die rote Low-Current LED für die Signalisierung der aktiven Heizperiode angeschlossen. Der Vorwiderstand ist auf der Steuerplatine integriert. Polarität beachten!

LED-Stecker
Power (grün)



(Draufsicht)

Hier wird die grüne Low-Current LED für die Signalisierung des Betriebs angeschlossen. Der Vorwiderstand ist auf der Steuerplatine integriert. Polarität beachten!

Poti-Stecker
Hub



(Draufsicht)

Anschluss für das 100k Ohm Potentiometer zur Einstellung des Hub. Der Schleifer wird mit dem Pin „Hub“ verbunden.

Poti-Stecker
Speed



(Draufsicht)

Anschluss für das 100k Ohm Potentiometer zur Einstellung der Wipp-Geschwindigkeit. Der Schleifer wird mit dem Pin „Speed“ verbunden.

KTY81-Stecker
Temperatur-Sensor



(Draufsicht)

Anschluss für den Temperatur-Sensor KTY81-200. Der Pin „Sense“ ist der „Heiße“ Draht. Die Polarität ist zu beachten.

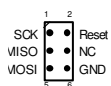
RS232



(Draufsicht)

Anschluss für die serielle Schnittstelle.
Parameter: 19200Baud, 1 Stoppbit, 8 Datenbits, keine Parität, kein Handshake

ISP-Stecker



(Draufsicht)

Anschluss der ISP-Schnittstelle. Hiermit kann der Controller programmiert werden.

8.2 MontagematerialAbbildung 8.1 –
StromanschlussEinbaukupplung (isoliert)
2,0mm-StiftBeziehbar:
www.conrad.de

Best.-Nr: 738687

Abbildung 8.2 - Anschluss
Versorgung

für 2,0mm-Stift

Beziehbar:
www.conrad.de

Best.-Nr: 738688

Abbildung 8.3 - Anschluss
HeizungEinbaukupplung (isoliert)
1,3mm-StiftBeziehbar:
www.Conrad.de

Best.-Nr: 738687

Abbildung 8.4 - Heizung-
Anschluss

für 1,3mm-Stift

Beziehbar:
www.conrad.de

Best.-Nr: 738609

Alu-Platte
Für Front und Seite
(ohne Abbildung)BxHxT:
128,4 x 40,3 x 2,5 mmBeziehbar:
www.conrad.de

Best.-Nr: 525499

Sonstiges Material



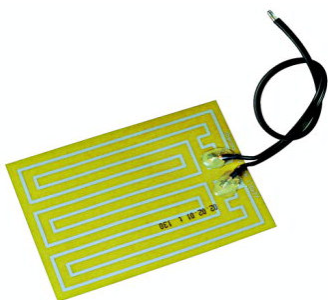
Steckernetzgerät
6-24V / 2A (max. 48W)

Ausgangsspannung mit Jacks
einstellbar, Polarität umkehrbar.

Beziehbar:
www.conrad.de

Best.-Nr: 510524

Abbildung 8.5 - Steckernetzgerät



Heizfolie
12V / 12W

Beziehbar:
www.conrad.de

Best.-Nr: 532878

Abbildung 8.6 - Hezelement

8.3 Versionen - History

Datum	Firmware	Bemerkung
Mai 2009	1.0	Erste öffentliche Version
08.07.2009	1.1	Verbesserte Darstellung für Versions-Info und Hilfe
09.07.2009	2.0	Bootloader und Firmware überarbeitet Bootloader Version 1.1: <ul style="list-style-type: none"> • Zeigt jetzt Status über LED's der Ätzmaschine an. Firmware: <ul style="list-style-type: none"> • Bei Hubeinstellung=0 wird Heizung ausgeschaltet • Logfile erweitert um 3 Einträge • Log-Intervall jetzt einstellbar • Bug beseitigt: Debug-Ausgabe • PI-Regler hinzugefügt (Reglerabweichung = 0) • Hilfe-Ausgabe angepasst Hinweis: Der Bootloader 1.0 funktioniert auch mit Firmware 2.0.

8.4 Abbildungs-Verzeichnis

ABBILDUNG 1 - ANSICHT DER PLATINE IN 3D.....	3
ABBILDUNG 2.1 - FRONTSEITE	4
ABBILDUNG 2.2 - ANSCHLUSS-SEITE	5
ABBILDUNG 6.1 – START HYPERTERM.....	19
ABBILDUNG 6.2 – NEUE VERBINDUNG	19
ABBILDUNG 6.3 - SCHNITTSTELLE	20
ABBILDUNG 6.4 - EIGENSCHAFTEN	20
ABBILDUNG 6.5 – SYMBOLLEISTE HYPER-TERMINAL.....	20
ABBILDUNG 6.6 - EIGENSCHAFTEN	21
ABBILDUNG 6.7 – ASCII KONFIGURATION.....	21
ABBILDUNG 6.8 – VERSIONS-INFO.....	22
ABBILDUNG 6.9 – HILFE DER ÄTZMASCHINE	22
ABBILDUNG 7.1 - ISP.....	24
ABBILDUNG 7.2 - PROGRAMMER AUSWÄHLEN.....	24
ABBILDUNG 7.3 - FUSEBITS / LOCKBITS.....	25
ABBILDUNG 7.4 – MANUELL PROGRAMMIEREN	26
ABBILDUNG 7.5 - PROGRAMMER SYMBOLLEISTE	27
ABBILDUNG 7.6 - BOOTLOADER AUSWÄHLEN.....	28
ABBILDUNG 7.7 – LOCKBIT NACH BOOTLOADER UPLOAD.....	29
ABBILDUNG 7.8 – MCS BOOTLOADER (SERIAL).....	30
ABBILDUNG 7.9 – MCS BOOTLOADER (MCS LOADER).....	30
ABBILDUNG 7.10 – SYMBOLLEISTE (PROGRAMMIERSYMBOL).....	30
ABBILDUNG 8.1 – STROMANSCHLUSS	32
ABBILDUNG 8.2 - ANSCHLUSS VERSORGUNG	32
ABBILDUNG 8.3 - ANSCHLUSS HEIZUNG.....	32
ABBILDUNG 8.4 - HEIZUNG-ANSCHLUSS	32
ABBILDUNG 8.5 - STECKERNETZGERÄT	33
ABBILDUNG 8.6 - HEIZELEMENT.....	33

8.5 Tabellen-Verzeichnis

TABELLE 4.1 - KOMMANDOÜBERSICHT.....	9
TABELLE 5.1 - DEBUG-DATEN REIHENFOLGE.....	18