

## Alles unter Kontrolle: COM, I<sup>2</sup>C und EEPROM

Mikrocontroller werden häufig als Herzstück autonomer Messstationen eingesetzt. Folgende Voraussetzungen sollten sie dazu erfüllen:

1. Sie müssen einen Speicher besitzen, der die Messdaten sicher verwahren kann - möglichst auch dann noch, wenn die elektrische Versorgung des Mikrocontrollers ausfällt.
2. Sie müssen gängige Kommunikationsschnittstellen zu Sensoren besitzen.
3. Sie müssen gängige Kommunikationsschnittstellen zu Terminals besitzen, damit die Daten problemlos zur weiteren Auswertung auf Computer übertragen werden können.

All dies kann unser Attiny2313 mit MikroForth leisten: Er besitzt ein EEPROM, welches Daten auch ohne elektrische Quelle dauerhaft speichern kann. Über den I<sup>2</sup>C-Bus kann er mit Sensoren und anderen Geräten kommunizieren und über die COM-Schnittstelle kann er die gespeicherten Daten an ein Terminal senden. Für genauere Erläuterungen zu EEPROM, I<sup>2</sup>C und COM-Schnittstelle sei auf die entsprechenden Abschnitte verwiesen. Hier soll anhand eines einfachen Beispiels betrachtet werden, wie MikroForth zum Anfertigen eines Temperaturmessprotokolls eingesetzt werden kann.

Der Aufbau ist einfach: An die I<sup>2</sup>C-Buchse der Attiny-Platine wird ein Temperatursensor LM75 mit der I<sup>2</sup>C-Adresse 157 angeschlossen (Abb. 1). Nicht vergessen sollten Sie, die Jumper zu setzen, die zum Pull-Up der beiden Leitungen SDA und SCL erforderlich sind. Der Datenaustausch mit dem PC erfolgt über dasselbe Kabel wie die Programmierung.

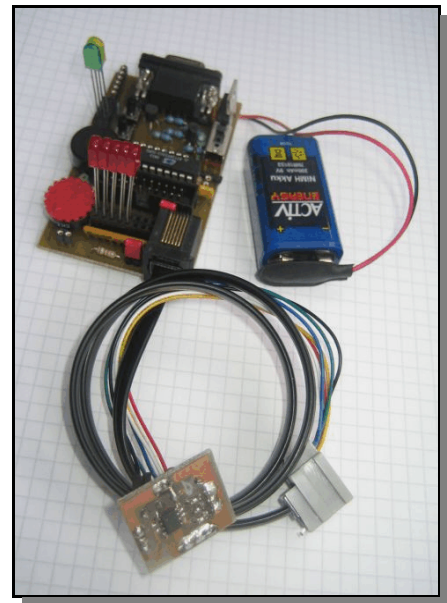


Abbildung 1

Folgende FORTH-Wörter stellt MikroForth für EEPROM, I<sup>2</sup>C und COM zur Verfügung:

Wort	Typ	Kommentar	Stack
>com	A	sendet TOS an die COM-Schnittstelle. Vorher muss die COM-Schnittstelle mit INITCOM initialisiert worden sein.	(n -)
>eprom	A	schreibt den Wert w in die Adresse a des EEPROMs. Vgl. eprom>	( w a - )

Wort	Typ	Kommentar	Stack
com>	A	legt über COM-Schnittstelle empfangenes Byte auf den Stack. Vgl. >com.	( - n)
eprom>	A	liest den Wert w aus der EEPROM-Adresse a und legt ihn auf den Stack Vgl. >eprom	( a - w )
i2cread	A	Ein Wert wird vom Slave gelesen; wenn ACK = 0 ist, wird ein Acknowledge-Signal gegeben.	(ACK - Wert)
i2cstart	A	Startsignal für I <sup>2</sup> C-Bus wird gesendet (SDA von 1 auf 0; dann SCL von 1 auf 0)	( - )
i2cstop	A	initialisiert den I <sup>2</sup> C-Bus (SCL und SDA auf 1); Datenrichtungsbits für SDA (PortB.5) und SCL (PortB.7) werden gesetzt.	( - )
i2cwrite	A	Ein einzelnes Byte (Wert oder Adresse) wird an den Slave gesendet; das Acknowledge-Bit wird auf den Stack gelegt.	(Wert/Adr - ACK)
initCom	A	initialisiert die COM-Schnittstelle: D0 = RxD D1 = TxD Baudrate = 9600 8 Bit kein Paritätsbit	( - )

Zunächst schauen wir uns den Messprozess an: Der Attiny soll im Sekundenabstand 20 Temperaturwerte im EEPROM aufzeichnen. Dies leisten die beiden Wörter `messung` und `main`:

```
: messung i2cstop i2cstart 1 wait 157 i2cwrite 1 i2cread ;
: main 20 1 do messung I >eprom loop ;
```

Schauen wir uns zunächst die Definition von `messung` an: Mit `i2cstop` wird die I<sup>2</sup>C-Schnittstelle initialisiert (SCL und SDA auf high), mit `i2cstart` wird das Startsignal gegeben (SDA wechselt von high auf low), mit `157 i2cwrite` wird unser Temperatursensor adressiert. Schließlich wird mit `1 i2cread` ein Temperaturwert vom LM75 abgefragt und auf den Stack gelegt; der Parameter 1 sorgt dafür, dass kein Acknowledge-Signal gegeben wird. Ein Acknowledge würde nämlich den LM75 dazu veranlassen, als nächstes die Nachkommastelle zu senden. Man beachte ferner, dass der LM75 bis zu 300 ms für eine einzige Temperaturmessung benötigt. Eine längere Pause (hier 1 Sekunde) zwischen den einzelnen Messungen ist

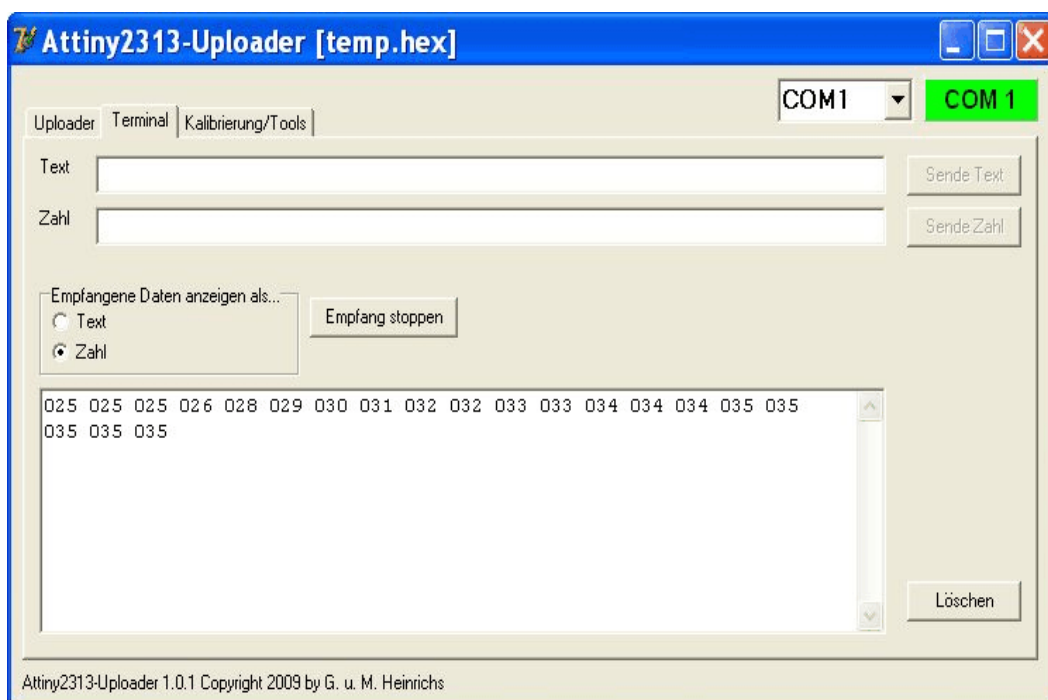
also unabdingbar!

Das Wort `main` besteht im Wesentlichen aus einer Zählschleife, bei der die einzelnen Messwerte im EEPROM abgelegt werden. Der Schleifenindex `I` gibt jeweils die Adressnummer des EEPROMs an.

Nun müssen wir die gemessenen Werte noch an den PC übertragen. Dazu setzen wir das Wort `eprom2com` ein:

```
: eprom2com eprom> >com 1 waitms ;  
: main initCom 20 1 do I eprom2com loop ;
```

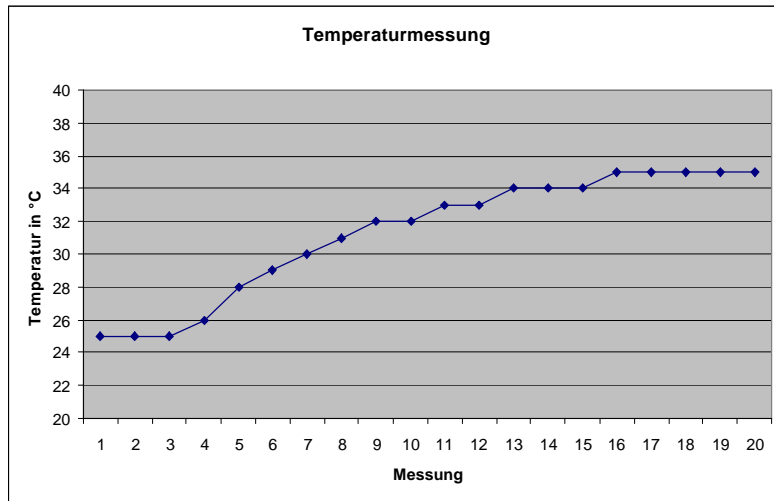
Bevor die Daten Byte für Byte über die COM-Schnittstelle übertragen werden können, muss diese initialisiert werden. Dies geschieht mit dem Wort `initCom`. Beim Senden der Bytes muss das Timing beachtet werden: Das Übertragen eines Bytes über die serielle Schnittstelle dauert eine gewisse Zeit. Während des Sendevorgangs arbeitet der Mikrocontroller aber schon sein Programm weiter ab. Würde man auf den Befehl `1 waitms` verzichten, würde der Mikrocontroller die nächste Übertragung starten wollen, bevor das letzte Byte vollständig übertragen wäre.



**Abbildung 2**

Wenn wir nun das Terminalprogramm von UPLOADER laufen lassen, werden sämtliche Messwerte in weniger als 1 Sekunde an den Computer übertragen. Abb. 2 zeigt das Ergebnis einer solchen Übertragung. Deutlich erkennt man, dass die Temperaturwerte zunächst konstant bleiben und dann rasch ansteigen. Wie ist das zustande gekommen? Während des Messvorgangs wurde

eine Lampe direkt über den LM75 gehalten; dadurch wurde dieser stark erwärmt. In Abb. 3 sind die Ergebnisse mit einem Tabellenkalkulationsprogramm graphisch dargestellt worden.



**Abbildung 3**