

Aufgaben

A Bearbeiten Sie eine der folgenden drei Aufgaben:







1. Addiersklave

Der Attiny2313 soll zwei Zahlen (zwischen 0 und 255) vom Terminalprogramm empfangen und deren Summe wieder an das Terminalprogramm senden.

Wenn Sie sich nicht ganz sicher fühlen, können Sie folgendermaßen vorgehen:

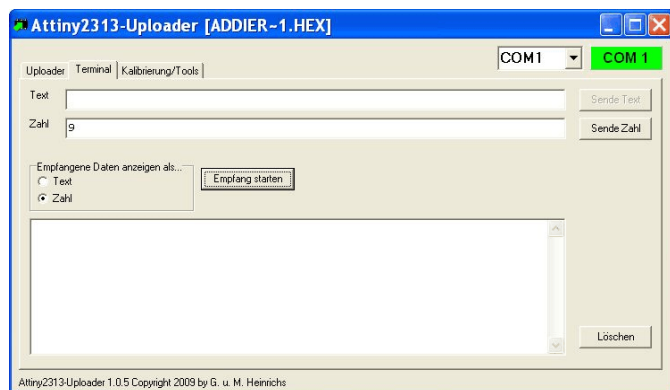
1.1 Öffnen Sie BASCOM, laden Sie die Vorlage-Datei und speichern Sie sie in einem **neuen Verzeichnis** unter einem **neuen Namen** ab. Denken Sie auch daran, in einem Kommentar am Anfang des Programms eine kurze Beschreibung des Programms zu geben.

1.2 Schreiben Sie nun das Programm; es muss folgendes leisten:

-  COM-Schnittstelle initialisieren und auf 9600 baud einstellen
-  Drei Variablen deklarieren
-  Ersten Summanden von der seriellen Schnittstelle empfangen und einer Variablen zuordnen
-  Zweiten Summanden...
-  Summanden addieren und der dritten Variablen zuordnen
-  Inhalt der dritten Variablen über die serielle Schnittstelle ausgeben

1.3 Programm auf PC sichern, kompilieren und auf den Attiny übertragen (ggf. Kurzanleitung zu Rate ziehen!)

1.4 Zuerst Uploader-Programm öffnen (falls nicht noch von der letzten Übertragung geöffnet) und die Lasche “Terminal” anklicken. Dann “Zahl” bei “Empfangene Daten anzeigen als...” aktivieren und Empfangsschaltfläche betätigen. Ggf. erscheint schon eine “empfangene” Zahl im Eingangsfeld; dann war der Empfangspuffer des Terminals nicht leer. Sie können sie bei Bedarf löschen.



1.5 Geben Sie nun Ihren ersten Sum- **Abbildung 1**

manden, zum Beispiel die Zahl 9 ein. Benutzen Sie dabei unbedingt die zweite Eingabezeile. Senden Sie diese Zahl an den Attiny. Verfahren Sie ebenso mit dem zweiten Summanden. Unmittelbar nach dem Abschicken des zweiten Summanden müssten Sie das Ergebnis erhalten.

Sollten Sie bei Ihrem Programm keine Schleife benutzt haben, müssen Sie für jede Rechenaufgabe den Attiny neu einschalten.

- 1.6 **Für Freaks:** Ergänzen Sie Ihren Rechensklaven so, dass er addieren **und** subtrahieren kann. Das Rechenzeichen soll nach den beiden Operanden als Zeichen gesendet werden.

2. Zahlen vom Terminal am Port B ausgeben

Zahlen, welche vom Terminal gesendet werden (zweite Eingabezeile benutzen!), sollen durch LEDs am Port B binär angezeigt werden.

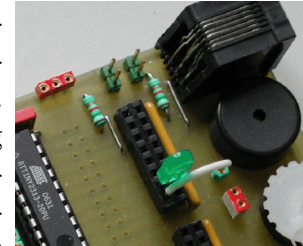
Diese Aufgabe ist etwas leichter als die Aufgabe 1; dafür gibt es aber keine weiteren Hinweise. Ggf. können die Erläuterungen bei Aufg. 1 aber hilfreich sein!

3. Cäsarkodierung

- 3.1 Programmieren eine (nicht-zyklische) Cäsarkodierung mit Shiftwert 3 und testen Sie sie aus. Beachten Sie ggf. die Hinweise bei Aufgabe 1. Bei dieser Aufgabe muss allerdings die erste Eingabezeile des Terminals benutzt werden und die Anzeige als Text aktiviert werden.
- 3.2 Ergänzen Sie Ihr Programm derart, dass ein Leerzeichen nicht(!) übertragen wird. Warum ist das sinnvoll?
- 3.3 Ändern Sie Ihr Programm so ab, dass die Cäsarkodierung zyklisch erfolgt; Sie können dies z. B. mit einer Verzweigungsstruktur erreichen.

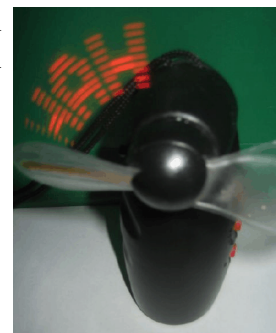
B Bearbeiten Sie ein oder zwei der fünf folgenden Aufgaben

1. Ergänzen Sie den Morsesender (Datei Morsesender.bas im Verzeichnis Morsesender) so, dass die Signale nicht als Lichtzeichen, sondern als Töne über den Beeper ausgegeben werden. Schließen Sie dazu den Beeper mit einem Kabel an den Ausgang PortB.0 an (vgl. Abb. 2; die abgebildete LED ist nicht erforderlich). Benutzen Sie für beide Töne jeweils ein eigenes Unterprogramm. Beachten Sie, dass ein Unterprogramm auch von einem Unterprogramm aus aufgerufen werden kann.

**Abbildung 2**

Information: Durch das Ein- und Ausschalten wird eine Membran im Beeper bewegt. Wird die Membran sehr rasch hin und her bewegt, so hört man einen Ton. Der Ton ist um so höher, je schneller die Membran sich bewegt. (Vgl. Aufg. 3 der letzten Veranstaltung und die Datei Ton.bas)

2. Wie Aufgabe 1; es soll jedoch für die Tonerzeugung ein einziges Unterprogramm mit Parameter `tonlaenge` benutzt werden: Wenn `tonlaenge` gleich 0 ist, dann soll ein kurzer Ton erzeugt werden, ansonsten ein langer.
3. Je nachdem, ob T0 oder T1 betätigt wird, soll ein hoher oder tiefer Ton zu hören sein. Benutzen Sie für die Töne ein einziges Unterprogramm `Ton` mit dem Parameter `hoehe`: Wenn `hoehe` gleich 0 ist, soll sie einen tiefen Ton produzieren, ansonsten einen hohen.
4. Nur für Freaks: Schreiben Sie ein Programm, welches Morsesignale empfängt: Ein kurzes Tastsignal bei T0 (kürzer als 0,5 Sekunden) soll zur Ausgabe des Textes “kurz”, ein längeres Tastsignal zur Ausgabe von “lang” über die serielle Schnittstelle führen. Schreiben Sie zunächst ein Unterprogramm, welches die Länge in Hundertstel Sekunden an den Unterprogramm-Parameter übergibt. (Der Parameter muss dazu nicht explizit in der Deklaration als “call by reference” gekennzeichnet werden.)
5. Nur für Freaks: Programmieren Sie Ihren Attiny so, dass er den Text “HAH” anzeigt, wenn die Platine im Dunkeln bewegt wird (vgl. den Ventilator in Abbildung 3). Legen Sie für jeden Buchstaben ein eigenes Unterprogramm an.

**Abbildung 3**