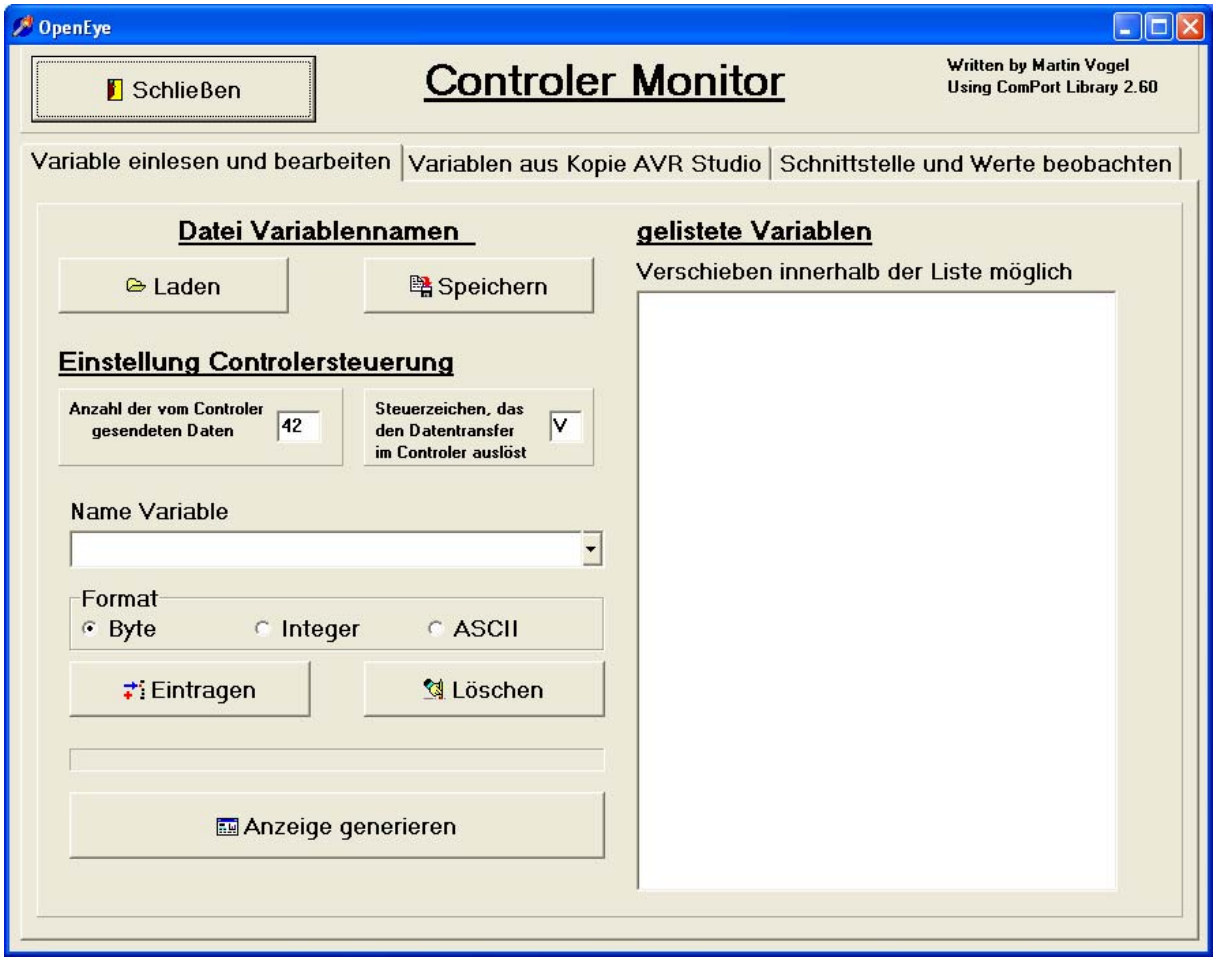
OpenEye.

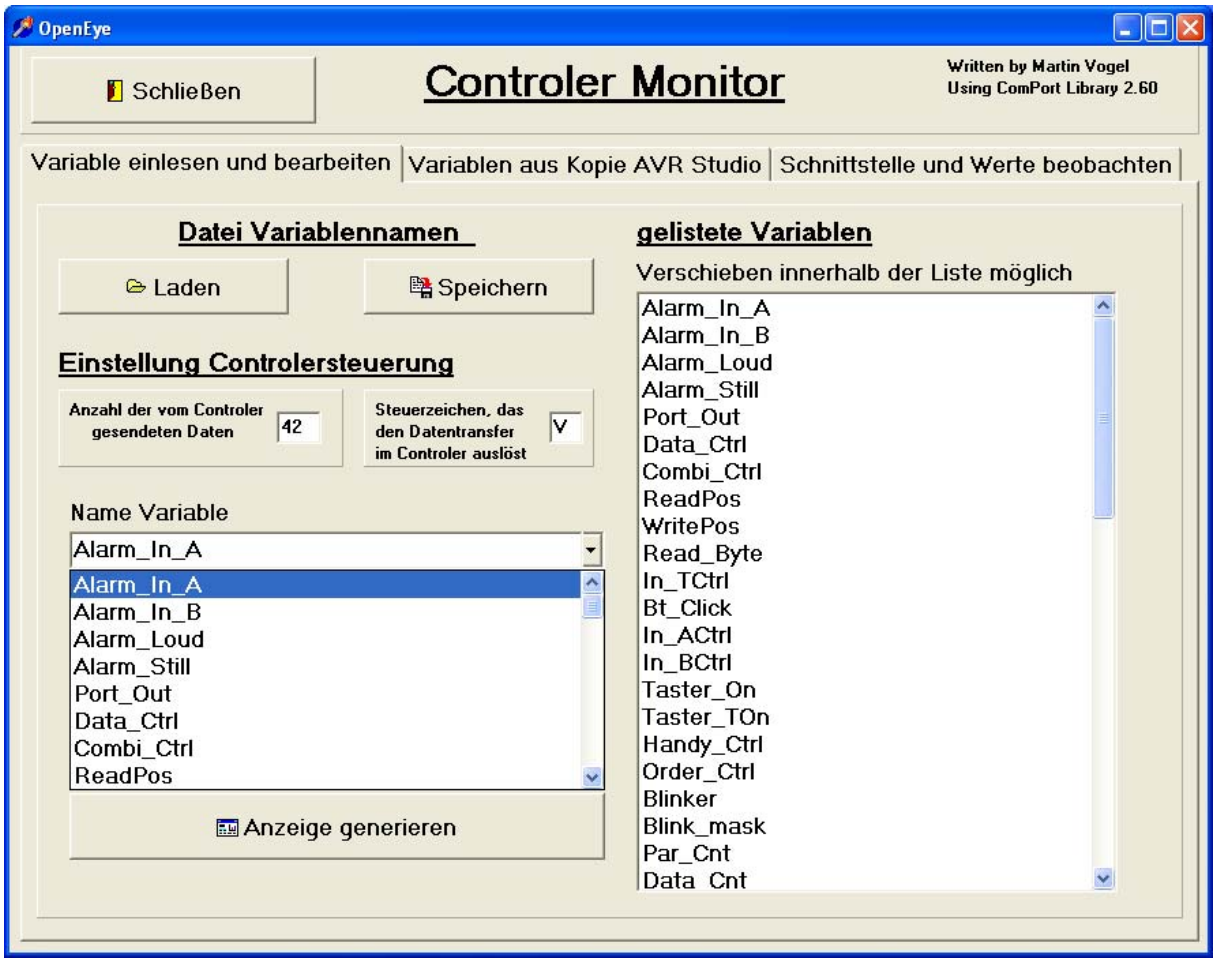
Geboren aus der ständigen Problematik, das Controlerprogramme erst mal nicht so wirklich kooperativ sind, habe ich diesen Monitor zur Ansicht der Variablenwerte entworfen. Er ist nicht superkomfortabel, aber durchaus ein Fenster in den Controler zur Laufzeit.
Im Prinzip besteht er aus einer Eingabeoberfläche für die Variablen. Dadurch ist eine einfache Darstellung später unter Verwendung der Originalnamen möglich. Eine Liste kann abgespeichert werden und ist somit immer wieder abrufbereit.
So ist das erste Tabulatorblatt ziemlich selbsterklärend.



Es wird der Name der im Controler verwendeten Variable eingegeben und das Format festgelegt, in welchem es später dargestellt werden soll. Mittels des Eintragen – Buttons wird die Variable übernommen. Doppelte Einträge sind nicht möglich, allerdings ist eine Anpassung bzw. Änderung des Formates jederzeit erlaubt. Nach einer Sicherheitsabfrage kann eine Änderung durchgeführt werden. Auch ist das Löschen eines Eintrages möglich, allerdings sollte man beachten, das dieses Programm mit den fortlaufenden Adressen des DSEG beschrieben wird. Bisher ist nur Byteweise die Zuordnung berücksichtigt. Daher sind nur Deklarationen „Byte 1“ zulässig, alles andere muß im Anschluß deklariert sein und kann nicht beobachtet werden.
Sind alle gewünschten Namen eingetragen, muß der Button Anzeige generieren betätigt werden. Dies löst die Generierung der Felddarstellung auf der 3. Seite aus.
Das Feld mit der Anzahl der gesendeten Daten muß mit der Anzahl der vom Controler gesendeten Bytes übereinstimmen. Das Steuerzeichen ist der vom PC gesendete Befehl und muß im Controler als Anforderung zur Datenübermittlung programmiert werden. Dies werde ich am Schluß noch behandeln.

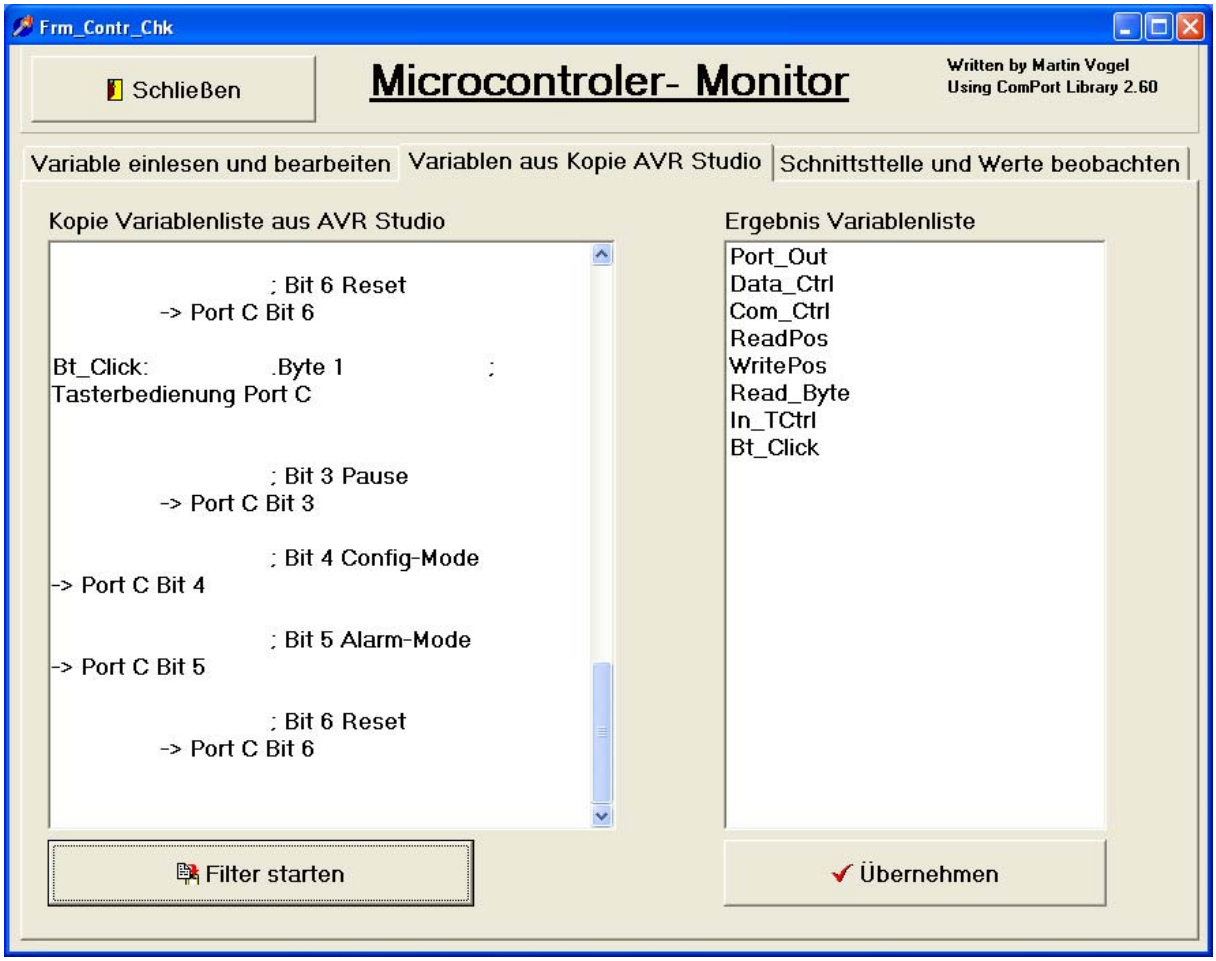
Nun noch die Erkärung zu der Liste der Variablennamen im rechten Teil der Seite. Sie dient dazu, die Namen zu verschieben. Nehmen wir einmal an, wir haben eine Liste erstellt und stellen fest, das wir eine Variable übersprungen haben. Natürlich können wir diese dann im Controlerprogramm verschieben und hier dann nachtragen. Ich bevorzuge den Weg, diese hier nachzutragen und dann zur richtigen Position zu schieben. Einfach anklicken, Maustaste festhalten und an der gewünschten Stelle loslassen. Nun brauch nur noch einmal eine Generierung durchgeführt werden und die Liste stimmt wieder mit der Reihenfolge der Variablen im Controler überein.

Die beiden Dateibutton öffnen die Dateidialoge, um diese Namenliste der Variablen sowie die Vorgabe der Anzahl der zu empfangenen Daten und das Steuerzeichen in einer Datei abzulegen, bzw. zu laden.



So sieht eine mit den Variablen des AVR-Programmes gefüllte Seite aus.

Kommen wir zur 2. Seite „Variablen aus Kopie“
Unter bestimmten Voraussetzungen ist es möglich, die Variablen aus dem DSEG des Controlerprogrammes direkt mittels Copy / Einfügen einzutragen. Dieser simple Parser zerlegt die ins linke Feld kopierten Daten und filtert die benutzten Variablen heraus, die er dann zur Kontrolle in die rechte Liste einfügt.

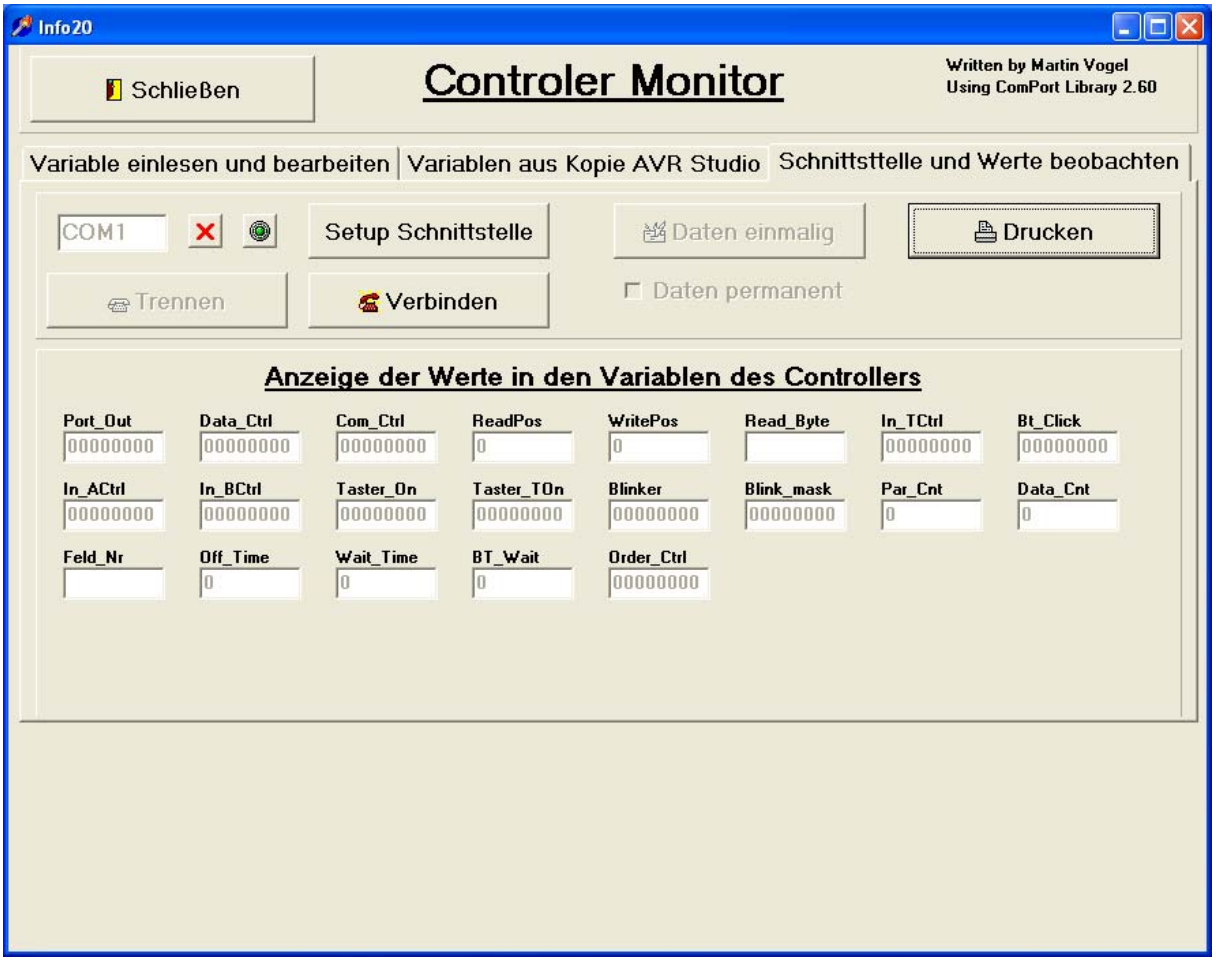


Nach dem Betätigen des Buttons „Filter starten“ wird die Variablenliste erzeugt, die nach Kontrolle übernommen werden kann.

Die Anpassung der Formate ist auf der Seite „Variablen einlesen und bearbeiten“ nachzuarbeiten.

Eine Übernahme leerer Listen ist nicht möglich. Ein Hinweistext informiert darüber.

Die Seite „Schnittstelle und Werte beobachten“
Hinweis: . Eine Bedienung ist nur möglich, wenn die Variablenfelder generiert sind.
Auf der 3.Seite ist der eigentliche Monitor Hier wird die Schnittstelle parametriert, kontaktiert oder getrennt. Ist eine RS 232 Verbindung hergestellt, kann ein einmaliges Lesen mittels Button oder ein permanentes Lesen durch die Checkbox initiiert werden.
Bei permanentem Lesen wird alle 100ms das Steuerzeichen an den Controler gesendet. Dieser muß nun einen Programmteil besitzen, dieses Steuerzeichen zu erkennen und das Senden der Variablenwerte auszulösen.



Wer möchte, der kann sich diese Ansicht des Formulares ausdrucken.

Zur Erfassung von Registerwerten sind diese auf Variablen zu kopieren. Auch kann ein Bytefeld angesehen werden, allerdings ist die Schreibarbeit auf der Controlerseite etwas größer. Da eine Deklaration „Byte 10“ z.B. nicht erkannt wird, muß das Feld mit 10 einzelnen Bytes deklariert werden. Z. B.
Wertfeld1 : Byte1
Wertfeld2 : Byte1
Wertfeld3 : Byte1 etc.
Sicherlich nicht umfangreich, doch hilfreich, bei der Erfassung der Wertezuweisung im Controler. Schnelle Signale müssen evtl.zwischengespeichert werden, aber wenn das Programm getestet ist und seinen Zweck erfüllt, spricht nichts gegen eine Entfernung von nicht benötigtem Code. Allerdings bevorzuge ich es, die paar Serviceroutinen zu belassen. Damit habe ich immer den Zugriff auf den inneren Ablauf im Controler.

Zum Schluß noch die Programmteile des Controlers in Assembler, die in das eigene Programm eingebunden werden müssen:

In der Hauptschleife habe ich einen Aufruf

```
Loop:                   ….
            RCall       PC_Order    ; empfangene Steuerzeichen bearbeiten
            ….
            RJMP        Loop
```

---------- Nun die wesentlichen Teile aus PC_Order ----------

```
PC_Order:   CLR         Temp
            LDS         Prg_Ctrl, Data_Ctrl
            ANDI        Prg_Ctrl, 0b00000001    ; Daten komplett
            BREQ        Get_Adr
            RCALL       ......                  ; Datenauswertung ….
            RJMP        End_Order
Get_Adr:    LDS         R_Merker, ReadPos       ; Pufferzeiger lesen
            LDS         S_Merker, WritePos      ; Pufferzeiger schreiben
            CP          R_Merker, S_Merker      ; beide gleich, dann Ende
            BRNE        Do_Order
            RJMP        End_Order
Do_Order:   LDI         XL,LOW(Com_Buff)        ; x-Pointer auf Empfangspuffer
            LDI         XH,HIGH(Com_Buff)
            ADD         XL, R_Merker
            ADC         XH, Temp
            LD          Buf_Reg, X              ; Zeichen aus Empfangspuffer
            STS         Read_Byte, Buf_Reg      ; in Variable schreiben
            INC         R_Merker                ; Lesezeiger erhöhen
            CPI         R_Merker, 90            ; Puffergrenze erreicht
            BRLO        Store_Pos
            CLR         R_Merker                ; Puffer auf Anfang (Ringspeicher)
Store_Pos:  STS         ReadPos, R_Merker       ; Lesezeiger zurückschreiben
            LDS         Prg_Ctrl, Data_Ctrl
            ANDI        Prg_Ctrl, 0b10000000    ; Daten erkannt ?
            BRNE        Go_D_In                 ; dann Daten eintragen
            CPI         Buf_Reg, 'R'            ; ab hier Steuerzeichen
            BRNE        Chk_Par
            RCALL       Reset                   ; z. B. Resetroutine
            RJMP        End_Order
Chk_Par:    CPI         Buf_Reg,'x'             ; irgendwas anderes
            BREQ        irgendwas
            CPI         Buf_Reg, 'V'            ; Werte angefordert ?
            BRNE        End_Order
            RCALL       Send_Value              ; sende Variablenwerte
            RJMP        End_Order
irgendwas:  RCALL       ……..
            RJMP        End_Order
Go_D_In:    RCALL       Writebuf
            RJMP        End_Order
irgendwas:  ….
End_Order:  RET
```

```
---------------------- Senden der Variableninhalte -------------------------------------
Send_Value:     LDI     Send_Byte, 'V'          ; Send_Byte =Register
                RCALL   SerOut
                LDI     Send_Byte, 'A'
                RCALL   SerOut
                LDI     Send_Byte, 'L'
                RCALL   SerOut
                LDI     Send_Byte, 'U'
                RCALL   SerOut
                LDI     Send_Byte, 'E'          ; nur zur Kennung
                RCALL   SerOut
                CLR     Temp
                CLR     Index_Reg               ; Bytezähler
Repeat22:       LDI     XL,LOW(Variable)        ; Adresse der 1. Variable
                LDI     XH,HIGH(Variable)
                ADD     XL, Index_Reg           ; Bytezähler addieren
                ADC     XH,     Temp
                LD      Send_Byte, X            ; Wert eintragen (Register)
                RCALL   SerOut                  ; Übergaberegister senden
                INC     Index_Reg               ; Bytezähler erhöhen
------------------ die Anzahl der gesendeten Daten  muß im Monitor übereinstimmen ------------

                CPI     Index_Reg, 42           ; Anzahl der Bytes erreicht

-----------------------------------------------------------------------------------------------------------
                BRLO    Repeat22                ; wenn nicht, nächste Var.
                RCALL   Send_Ok                 ; OK Kennung senden
                RCALL   Send_Ok                 ; 2*
                RET

;-------------------Empfangene Daten an PC quittieren ----------------------------
Send_Ok:        LDI     Send_Byte, 'O'
                RCALL   SerOut
                NOP
                LDI     Send_Byte, 'K'
                RCALL   SerOut
                RET

;--------------------------------- Byte senden ---------------------------------------------

SerOut:         SBIS    UCSRA,UDRE              ; Warten Freigabe UDR
                RJMP    SerOut
                OUT     UDR, Send_Byte          ; Übergaberegister
                RET
```

Ich hoffe, das wesentliche ist klar.

Ach ja, der Empfang vom PC

```
;-------------------- Empfangsroutine im Controler -----------------

int_rxc:    PUSH    temp_Reg                ; temp auf dem Stack sichern
            IN      temp_Reg, sreg          ; SREG sichern
            PUSH    temp_Reg
            PUSH    S_Merker                ; Beschreiben des Ringpuffers
            CLR     Zero
            LDS     S_Merker, WritePos      ; Position im Ringpuffer
            IN      Work_Reg, UDR
            LDI     XL,LOW(Com_Buff)        ; x-Pointer auf Empfangspuffer
            LDI     XH,HIGH(Com_Buff)
            ADD     XL, S_Merker
            ADC     XH, Zero
            ST      X, Work_Reg
            INC     S_Merker
            CPI     S_Merker, 90            ; Puffergrenze erreicht?
            BRLO    End_rxc
            CLR     S_Merker
End_rxc:    STS     WritePos, S_Merker      ;aktuelle Position merken
```

Vielleicht sollte ich trotzdem ganz kurz beschreiben, wie’s funktioniert.
Über ein vom PC gesendetes Steuerzeichen wird die Routine Send_Value aufgerufen. Nach dem Versenden einer Kennung wird eine Schleife durchlaufen. (ich hab hier das Wort „VALUE“ benutzt ,ist leider nicht frei wählbar)
Die Anfangsvariable, bzw. die Adresse der Anfangsvariable wird in das X-Registerpaar (>R25 , ist in der Include-Datei definiert) geschrieben. Es ist aber auch ein anderes Adressregisterpaar denkbar.
Durch eine direkte Konstante werden nun entsprechend viele Durchläufe der Schleife gesetzt. Dabei wird jedes Mal die Adresse im X-Registerpaar auf die nächste Variable gesetzt, der Wert in ein Register (hier Send_Byte) geschrieben und der Senderoutine übergeben.
Abschließend zur sauberen Trennung wird ein „OKOK“ gesendet.
Der Wert der Konstante muß im Monitor eingetragen werden. Vielleicht werde ich diesen Wert später mal vom PC senden, um etwas flexibler zu sein.
Viel Spaß mit OpenEye, dem „offenen Auge“ in den Controler.

Martin Vogel