Datenaustausch zwischen KIM oder anderen 65xx-Systemen und dem Mikrocomputersystem TRS-80

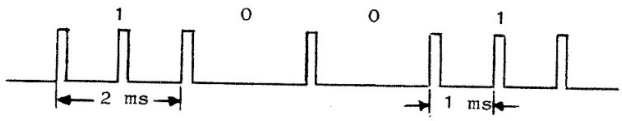
Von Dr. Claus Wünsche, Eschenweg 2, 5778 Meschede

E: A simple method is shown to exchange data between microcomputer systems. The audio tape format of the TRS-80 computer with 500 Baud has been realized with KIM-1. No other interface is needed than a simple IN/OUT port. Programs for KIM-1 are given to read a TRS-80 tape or to generate a TRS-80 readable tape.

In diesem Beitrag wird eine einfache Möglichkeit zum Datenaustausch zwischen Mikrocomputersystemen aufgezeigt. Das Kassettenaufzeichnungsformat des TRS-80 mit einer Übertragungsgeschwindigkeit von 500 Baud kann mit einem Mikroprozessor ohne Hardwareergänzung an einem IN/OUT-Port zum Lesen oder Schreiben realisiert werden. Es werden Programme für den KIM zum Einlesen eines auf dem TRS-80 erzeugten String-File und zur Gewinnung einer TRS-80 kompatiblen Kassettenaufnahme mit dem KIM mitgeteilt. Diese Möglichkeit des Datenaustausches wird für folgende Anwendungen verwendet:

1.) Steuerung einer automatischen Meßvorrichtung mit dem KIM. Die gewonnenen Daten werden im KIM gespeichert und blockweise auf Band abgelegt. Die Weiterverarbeitung der Daten erfolgt im TRS-80.
2.) Darstellung eines Speicherauszuges des KIM auf dem Bildschirm des TRS-80
3.) Verwendung des in Nr. 6 dieser Zeitschrift veröffentlichten in BASIC geschriebenen Assemblers für 65xx Prozessoren. Mit geringen Änderungen wurde er für den TRS-80 angepaßt. Der erzeugte Maschinencode wird per Band in den KIM übertragen.

Das Kassettenformat des TRS-80 arbeitet mit einem Impulsraster der Periode 500 Hz zur Synchronisation. Zwischen diese Impulse wird für eine '1' ein zusätzlicher Impuls eingeschoben (Impulsabstand jetzt 1 ms), für eine '0' bleibt der Impulsabstand 2 ms.



Die Impulslänge beträgt ca. 100 µs. Bedingt durch das schlechte Impulsverhalten der Kassettengeräte wird die Impulsform verzerrt. In dem vorliegenden Programm wird durch den KIM ein 88 µs langer Impuls erzeugt. Damit werden einwandfrei lesbare Bänder erhalten. Die Amplitude muß allerdings sorgfältig eingestellt werden, und gutes Bandmaterial muß verwendet werden.
Eine Bandaufnahme beginnt mit einem Vorlauf von ca. 800 Impulsen mit 2 ms Abstand. Die Daten werden im ASCII-Code ohne Paritätsbit mit MSB zuerst ausgegeben. Als Startzeichen dient % + Paritätsbit, d.h. A5 (Hex). Die Trennung zweier BASIC-Variablen erfolgt durch " ' ", d.h. als Trennzeichen dient 2C (Hex). Das Ende einer Aufnahme wird durch CR (carriage return), d.h. OD (Hex) angezeigt.
Damit die Lage des Bit-Impulses zwischen den Sychronisationsimpulsen nicht kritisch ist, wird durch KIM beginnend 895 µs nach dem Synchronisationsimpuls insgesamt 16 mal auf Bit =1?, d.h. über einen Zeitraum von 208µs abgefragt. Wenn innerhalb dieses Zeitraums keine 1 gefunden wurde, wird Bit ="0" gesetzt. Auf diese Weise wird Impulsverzerrungen durch den Kassettenrecorder Rechnung getragen.

Das Programm zur Erzeugung einer TRS-80 kompatiblen Kassettenaufnahme wird durch die Unterprogramme WORT und BYTE auf die gewünschte Variable in BASIC angepaßt. Hier wird nur die Ausgabe einer String-Variable aus Hexadezimalzeichen beschrieben, wofür WORT und BYTE identisch sind. Mit etwas größerem Aufwand können auch Gleitkommavariable mit einfacher oder doppelter Genauigkeit oder Integer-Variable ausgegeben werden. Einzelheiten zur Übertragung von Gleitkommazahlen können vom Autor auf Wunsch mitgeteilt werden. Das Programm zum Einlesen einer Kassettenaufnahme des TRS-80 durch den KIM wurde nur für Hexadezimalzeichen niedergeschrieben, da bisher nur hierfür Bedarf besteht.

Die Programme starten in OCOO bzw. ODOO. Zur Verschiebung in andere Speicherbereiche müssen die Adressen der Unterprogramme in den Aufrufen geändert werden. Absolute Sprünge erfolgen sonst nur am Programmende zurück in das Monitorprogramm. In der Programmniederschrift wurde der Einfachheit halber auf das $-Zeichen bei der Angabe von HEX-Adressen verzichtet. Die Startadresse des ein- oder auszugebenden Speicherbereichs im KIM wird fortlaufend erhöht, so daß durch wiederholte Starts aufeinanderfolgende Speicherbereiche eingelesen oder ausgegeben werden können.

Programm zum Einlesen eines String-File des TRS-80 in KIM
(1 String-Variable)
Das Programm ist für eine gerade Anzahl von HEX-Zeichen niedergeschrieben, die im KIM byteweise gepackt werden. Sollen beliebige im ASCII-Code übertragene Zeichen unverändert abgespeichert werden, so wird ab Adresse OD28 ein Sprung (4C 4O OD) eingesetzt. Die Wandlung in HEX-Zeichen und das Packen werden dadurch übersprungen. Für Wandlung und Packen wird die KIM-Routine PACKT (ab 1AOO) verwendet. Die Anfangsadresse in 17F5 und 17F6 wird hochgezählt. Dadurch kann zum Einlesen aufeinanderfolgender Zeichenketten sofort wieder bei Adresse ODOO gestartet werden. Das Programm endet bei Erreichen der Endadresse oder eines Endzeichens CR und der Monitor meldet sich mit der letzten Adresse +1. Kann ein Zeichen nicht als HEX-Zeichen interpretiert werden ,so meldet sich der Monitor mit FFFF.

```
<17F5>,<17F6> ADBE  Startadresse für Zeichenkette
<17F7>,<17F8> ADEN  Endadresse+1
<17E9>        SAVX
<00F7>        BYTE  momentane Daten
<00FA>,<00FB> ADMO  momentane Adresse
 X und Y werden verwendet
 PB2    Eingang  Signal
 PB1    Ausgang  Band Start/Stop
```

```
ODOO    A9 02          LDA #02       Einrichten Ports
        8D 03 17       STA  1703
        A9 00          LDA #00
        8D 02 17       STA  1702     PB1 low, Band Start
        AD F5 17       LDA  ADBE     Startadresse umspeichern
        85 FA          STA  ADMO
        AD F6 17       LDA  ADBE+1
        85 FB          STA  ADMO+1
OD14    20 70 OD STZ   JSR  HOLBIT   Warten auf Startzeichen
        26 F7          ROL  BYTE     Bit einschieben
        A9 A5          LDA #A5       Lade Startzeichen
        C5 F7          CMP  BYTE     Vergleiche
        DO F5          BNE  STZ
OD1F    20 91 OD ZEICH JSR  HOLBY    HEX-Zeichen 1
        A5 F7          LDA  BYTE
```

```
          C9 0D          CMP #0D        Vergleiche mit Endzeichen
          F0 38          BEQ  ZUEND
 0D28     20 00 1A       JSR  PACKT     PACKT in KIM
          98             TYA            Y ≠ 0 kein HEX-Zeichen
          D0 3A          BNE  FEHL
 0D2E     20 91 0D       JSR  HOLBY     HEX-Zeichen 2
          A5 F7          LDA  BYTE
          C9 0D          CMP #0D        Vergleiche mit Endzeichen
          F0 29          BEQ  ZUEND
 0D37     20 00 1A       JSR  PACKT     PACKT in KIM
          98             TYA            Y ≠ 0 kein HEX-Zeichen
          D0 2B          BNE  FEHL
          AD E9 17       LDA  SAVX
 0D40     91 FA          STA (ADMO),Y   HEX-Byte abspeichern
          E6 FA          INC  ADMO      Adresse erhöhen
          A5 FA          LDA  ADMO
          8D F5 17       STA  ADBE
          D0 07          BNE  NUEB
 0D4B     E6 FB          INC  ADMO+1

 0D4D     A5 FB          LDA  ADMO+1
          8D F6 17       STA  ADBE+1
          A5 FB    NUEB  LDA  ADMO+1    Mit Endadresse ver-
          CD F8 17       CMP  ADEN+1    gleichen
          D0 C6          BNE  ZEICH
          A5 FA          LDA  ADMO
          CD F7 17       CMP  ADEN
          D0 BF          BNE  ZEICH
 0D60     A9 02    ZUEND LDA #02
          8D 02 17       STA  1702      PB1 high, Band Stop
          4C 4F 1C       JMP  1C4F      zurück Monitor
          A9 02    FEHL  LDA #02
          8D 02 17       STA  1702      PB1 high, Band Stop
          4C 29 19       JMP  1929      zurück Monitor mit FFFF

 0D70     AD 02 17 HOLBIT LDA  1702      Laden PB
          29 04          AND #04        abtrennen Bit 2
          F0 F9          BEQ  HOLBIT    Synchronisationsimpuls ?
 0D77     A2 B3          LDX #B3        895 µs warten
          CA       S2    DEX
          D0 FD          BNE  S2
          A2 10          LDX #10
          AD 02 17 S3    LDA  1702      16 mal auf Bit prüfen
          29 04          AND #04          (208 µs)
          D0 05          BNE  S4        Sprung bei Bit = 1
          CA             DEX
          D0 F6          BNE  S3
          18             CLC            Bit = 0
          60             RTS            Bit in Carry
          A2 30    S4    LDX #30        Verzögerung bei    = 1
          CA       S6    DEX              (240 µs)
          D0 FD          BNE  S6
          38             SEC            Bit = 1
 0D90     60             RTS            Bit in Carry
```

```
0D91    A0 08     HOLBY LDY #08
        20 70 0D  S5    JSR  HOLBIT    Bit in Carry
        26 F7           ROL  BYTE
        88              DEY
        D0 F8           BNE  S5
0D9B    60              RTS
```

Programm zur Erzeugung einer TRS-80 kompatiblen Kassettenaufnahme mit KIM
Das Hauptprogramm liefert die Impulse, setzt das Startzeichen und verzweigt bei Startzeichen, Trennzeichen oder Endzeichen auf entsprechende Programmabschnitte.
Die Erzeugung des gewünschten Datenformats, das Zählen der Zeichen und das Einfügen des Endzeichens (CR) erfolgt in den Unterprogrammen BYTE und WORT.

```
PB0     Ausgang  Signal
PB1     Ausgang  Band Start/Stop
X und Y werden verwendet

0C00    A9 03            LDA #03
        8D 03 17         STA  1703      Ports einrichten
        A9 01            LDA #01
        8D 02 17         STA  1702      PB1 low, Band Start
0C0A    20 96 0C         JSR  VERZ      Verzögerung 262 ms
        A0 04            LDY #04        Vorlauf 800 Impulse
        A2 C8     L1     LDX #C8
        A9 FA     L2     LDA #FA
        8D 45 17         STA  1745      Timer 2 auf 2 ms
0C16    20 7B 0C         JSR  IMP       Impuls
        AD 47 17  W3     LDA  1747      Warten Timer 2
        10 FB            BPL  W3
        CA               DEX
        D0 F0            BNE  L2
        88               DEY
        D0 EB            BNE  L1        Ende Vorlauf, Y = 0
        84 E9            STY  BYZ       Bytezähler Null setzen
        A9 FA            LDA #FA
        8D 45 17         STA  1745      Timer 2 auf 2 ms
0C2B    A9 6B            LDA #6B
        8D 05 17         STA  1705      Timer 1 auf 856 µs
        A9 A5            LDA #A5        Startzeichen laden
0C32    A2 08     SBYTE  LDX #08        Laden Bitzähler
        2A        BIT    ROL  A         Bit in Carry
        48               PHA            Akku retten
        2C 07 17  W1     BIT  1707      Warten Timer 1
        10 FB            BPL  W1
        90 03            BCC  W2        Sprung bei Bit = 0
0C3D    20 7B 0C         JSR  IMP       Impuls
        2C 47 17  W2     BIT  1747      Warten Timer 2
        10 FB            BPL  W2
0C45    20 7B 0C         JSR  IMP       Impuls
        A9 FA            LDA #FA
        8D 45 17         STA  1745      Laden Timer 2 2 ms
0C4D    A9 6B            LDA #6B
        8D 05 17         STA  1705      Laden Timer 1 856 µs
        68               PLA            Akku holen
        CA               DEX
        D0 DE            BNE  BIT       Ende Schleife Bit
        2A               ROL  A         Byte in Akku wiederher-
```

```
                C9 0D           CMP #0D         Endzeichen ? stellen
                F0 12           BEQ  L4
                C9 2C           CMP #2C         Trennzeichen ?
                F0 09           BEQ  L3
                C9 A5           CMP #A5         Startzeichen ?
                F0 05           BEQ  L3
0C63            20 A1 0C        JSR  BYTE       neues Byte
                D0 CA           BNE  SBYTE

0C68            20 A1 0C L3     JSR  WORT       neues Wort beginnen
                D0 C5           BNE  SBYTE
0C6D            20 96 0C L4     JSR  VERZ       Verzögerung 262 ms
                A9 02           LDA #02
                0D 02 17        ORA  1702
                8D 02 17        STA  1702       PB1 high, Band Stop
0C78            4C 4F 1C        JMP  1C4F       zurück Monitor

0C7B            A9 01     IMP   LDA #01
                0D 02 17        ORA  1702
                8D 02 17        STA  1702       Signal high
0C83            A9 0B           LDA #0B         Laden Impulslänge
                8D 05 17        STA  1705       Timer 1 x 8 µs
                AD 07 17 W4     LDA  1707       Warten Timer 1
                10 FB           BPL  W4
                A9 FE           LDA #FE
                2D 02 17        AND  1702
                8D 02 17        STA  1702       Signal low
                60              RTS

0C96            A9 FF     VERZ  LDA #FF
                8D 07 17        STA  1707       Laden Timer 1
                AD 07 17 W5     LDA  1707       Warten Timer 1
                10 FB           BPL  W5
0CA0            60              RTS
```

Unterprogramm (BYTE / WORT) für eine String-Variable (HEX-Kette)

Folgende Adressen sind vor dem Start zu laden, bzw. werden belegt:

```
<00DC> BYE  Anzahl der auszugebenden Byte (max. 7C für TRS-80 )
<00DD> = 0  HZ  Hilfszähler für Halbbyte, zu Beginn Null
<00DE>,<00DF> SADL,SADH  Startadresse für Zeichenkette
<00E9>      BYZ  Bytezähler

0CA1    A5 E9     BYTE  LDA  BYZ
        C5 DC           CMP  BYE        Vergleiche auf Ende
        D0 04           BNE  B1
        A9 0D           LDA #0D         Lade Endzeichen CR
        D0 26           BNE  ENDE
        A5 DD     B1    LDA  HZ         Welches Halbbyte ?
        D0 0A           BNE  B2
        B1 DE           LDA (SADL),Y    Laden Byte
        4A              LSR
        4A              LSR
        4A              LSR
        4A              LSR             Bereitstellen Bit 4-7
```

```
                E6 DD          INC  HZ          Hilfszähler auf 1
                D0 0E          BNE  WD
                B1 DE    B2    LDA  (SADL),Y   Laden Byte
                29 0F          AND  #0F         Bereitstellen Bit 0-3
                E6 DE          INC  SADL        Adressen erhöhen
                D0 02          BNE  B3
                E6 DF          INC  SADH
                E6 E9    B3    INC  BYZ         Erhöhe Bytezähler
                C6 DD          DEC  HZ          Hilfszähler auf 0
                D8       WD    CLD              Wandle HEX-Zeichen
                C9 0A          CMP  #0A
                18             CLC
                30 02          BMI  WD1
                69 07          ADC  #07
                69 30    WD1   ADC  #30         Zeichen im ASCII-Code
06D1            60     ENDE    RTS
```