Print Decimal, Hex and Binary routines#

```
00010          .LI OFF
00020 *
00030 *
00040 *
00050 *
00060 *
00070 *
00080 ------------------------------
00090 * DEZ0:
00100 * Ausgabe einer 16-Bit Zahl mit
00110 * fuehrenden Nullen.
00120 *
00130 * DEZSP:
00140 * Ausgabe einer 16-Bit Zahl
00150 * Ohne fuehrende Nullen.
00160 *
00170 * <A> Register: LO-Byte der Zahl
00180 * <X> Register: HI-Byte der Zahl
00190 *
00200 ------------------------------
00210 *
00220 *
00230 DEZ0     STA WERT      Zahl in
00240          STX WERT+1    Hilfregister
00250          LDA #0        "Null"-Flag
00260          STA FLAG      setzen
00270          JMP NUM       Ausgabe
00280 *
00290 DEZSP    STA WERT      Zahl in
00300          STX WERT+1    Hilfsregister
00310          LDA #$80      keine Nullen
00320          STA FLAG
00330 *
00340 NUM      LDX #8        5 Stellen
00350 .1       LDA #0        Ziffer=0
00360          STA DIGIT
00370 .2       LDA WERT      Zahl-
00380          SEC           entsprechender
00390          SBC NTAB,X    Zehner-
00400          TAY           potenz
00410          LDA WERT+1
00420          SBC NTAB+1,X  ist Subtraktion erfolgreich
00430          BCC .3        Nein\^_\^_\^_
00440          STA WERT+1    Differenz
00450          STY WERT      speichern
00460          INC DIGIT     Ziffer hochzaehlen
00470          JMP .2        Noch ein Versuch
00480 *
00490 .3       LDA DIGIT     Ist Ziffer
00500          BNE .4        ungleich Null, Ja\^_\^_\^_
00510          CPX #0        Einerstelle erreicht
00520          BEQ .4        Ja, dann Ausgabe
00530          BIT FLAG      "Null"-Flag
00540          BMI .5        testen. Negativ, dann Null nicht ausgeben.
00550 .4       ASL FLAG      "Null"-Flag aus Ausgabe schieben
00560          ORA #$30      +ASCII "0"
```

```
00570          STX HOLD      <X> retten
00580          JSR CHAROUT   Ziffer ausgeben
00590          LDX HOLD      <X> holen
00600 .5       DEX           naechste
00610          DEX           Ziffer
00620          BPL .1
00630          RTS
00640 *
00650 NTAB     .DA 1,10,100,1000,10000
00660 *
00670 ------------------------------
00680 *
00690 * HEX16:
00700 * Ausgabe einer 16-Bit Zahl in
00710 * hexadezimaler Form.
00720 *
00730 * HEX8:
00740 * Ausgabe einer 8-Bit Zahl in
00750 * hexadezimaler Form.
00760 *
00770 * <A> Register: LO-Byte der Zahl
00780 * <X> Register: HI-Byte der Zahl
00790 *
00800 ------------------------------
00810 *
00820 HEX16    PHA           LO-Byte retten
00830          TXA           HI-Byte zuerst ausgeben
00840          JSR HEX8
00850          PLA           dann LO-Byte
00860 *
00870 HEX8     PHA           Wert retten
00880          LSR           erst HI-Nibble
00890          LSR
00900          LSR
00910          LSR
00920          JSR HEX       ausgeben
00930          PLA           dann
00940          AND #$F       LO-Nibble
00950 *
00960 HEX      CMP #10       Zahl>=10
00970          BCC .1        Nein\^_\^_\^_
00980          CLC           +7 fuer "A"-"F"
00990          ADC #7
01000 .1       ADC #$30      +ASCII "0"
01010          JMP CHAROUT   Ausgebe
01020 *
01030 ------------------------------
01040 *
01050 * BIN:
01060 * Gibt die Zahl im <A> Regis-
01070 * ter in dualer (binaerer)
01080 * Form aus.
01090 ------------------------------
01100 *
01110 BIN      STA WERT      Wert retten
01120          LDA #8        8 Bits
01130          STA HOLD
01140 *
01150 .1       ASL WERT      Bit ins Carry schieben
```

```
01160          LDA #$18      <A>=ASCII "0" /2
01170          ROL           Carry ins <A> Register schieben
01180          JSR CHAROUT   und ausgeben
01190          DEC HOLD      naechstes
01200          LDA HOLD      Bit
01210          BNE .1        wiederholen
01220          RTS
01230 *
01240 ------------------------------
01250 *
01260 * ZEICHENAUSGABEROUTINE
01270 *
01280 ------------------------------
01290 *
01300 CHAROUT  TAX
01310          LDA $E407
01320          PHA
01330          LDA $E406
01340          PHA
01350          TXA
01360          RTS
01370 *
01380 ------------------------------
01390 *
01400 * Arbeitsregister
01410 *
01420 ------------------------------
01430 *
01440 WERT     .HX 0000
01450 HOLD     .HX 00
01460 FLAG     .HX 00
01470 DIGIT    .HX 00
```