Telex-monitor-program

M. B. Immerzeel

Those who own a "simple" microcomputer such as the KIM or the SYM, and who also own a telex transceiver, will long since have thought about using the telex as a terminal and may already have a program for that purpose. However, most of these programs are limited to noting down addresses with their contents one below the other, and are therefore not particularly economical with paper. The program described below offers the following possibilities:
a. Loading a program in hexadecimal code.
b. Loading ASCII characters.
c. Printing memory contents as hexadecimal numbers.
d. Printing ASCII characters.
e. Starting a program.

Although the program was written for the KIM, it uses only a few subroutines of the KIM monitor program. It can therefore easily be rewritten for any other machine with a 6502 processor. In doing so, account must be taken of the capabilities the computer in question has for input and output of data. This means that the telex can also be used as a printer with this program on an expanded 6502 computer.

Subroutines
Because the length of the elements of the Murray code characters used by the telex is 20 ms (see the earlier article "Morse-telexconvertor" in RB December 1980 and February 1981), and these therefore take relatively long, it was unfortunately not possible to use many subroutines of the KIM monitor program. The program therefore requires the full pages $02 and $03, with the table used for converting ASCII to Murray, plus one further subroutine, on page $01. When loading a cassette with this program, it is therefore advisable to use $0100 as the start address and $0400 as the end address.
After connecting the telex (see the last section) and loading the program, it can be started by starting at address $0200. If everything goes correctly, the telex will then type out the word "KIM", followed by a carriage return and a line feed. The telex keyboard can now be used to choose one of the operating modes already mentioned under points a through e. Some keys have been given a different or an additional function. The "carriage return" key, for instance, will not only return the carriage to the start of the line but will also give a line feed. This frees up the "line feed" key, which can be used to leave a chosen mode and restart the program.

An important feature for a monitor program is that a number of subroutines are available to the programmer. The following subroutines can be called:

CRLF at address $039E. This subroutine returns the telex carriage to the start of the line and gives a line feed.

HBIT at address $03A7. This subroutine makes the telex print a space.

PR at address $033C. The contents of the accumulator are printed by the telex as a two-digit hexadecimal number.

PRTASC at address $0350. If the accumulator contains the code for an ASCII character, the corresponding character is printed by the telex.

ADR at address $03F0. The contents of addresses $17EE (VEBC) and $17ED (VEBB) are displayed by the telex in that order (without a space between the bytes).

INKRKT at address $03AB. This subroutine contains a wait loop in which it waits until a key is pressed. The ASCII code for the relevant key is then loaded into the accumulator. After pressing "figures" or "letters", it waits for the next key. Memory locations $00F8 (INL) and $00FF (MODE) are used for this. After pressing the "line feed" key, the accumulator contains $00.

Lijst 1

```
17C0 18             CLC
17C1 F8             SED
17C2 A5 01          LDAZ  $01
17C4 65 03          ADCZ  $03
17C6 8D ED 17       STA   $17ED
17C9 A5 00          LDAZ  $00
17CB 65 02          ADCZ  $02
17CD 8D EE 17       STA   $17EE
17D0 20 9E 03       JSR   $039E
17D3 20 F0 03       JSR   $03F0
17D6 4C 00 02       JMP   $0200
```

Using the monitor

As an example of how to use the monitor, the program shown in listing 1 is used, which adds two four-digit decimal numbers. The first number (0132) must be loaded into memory locations $0000 and $0001, and the second number (5694) into memory locations $0002 and $0003 (in high-low order). DAZ $01 means that the contents of memory location $0001 are loaded into the accumulator (LDA, Z-page). After storing the result in memory locations $17ED and $17EE, two subroutines of the monitor program are used, namely CRLF (JSR 039E) for the carriage return and line feed, and then ADR (JSR 03F0) to print the result. The monitor program is then restarted (JMP 0200).

Lijst 2

```
KIM
LH
OK
17C0 18 F8 A5 01 65 03 8D ED 17A17 A5 00 65 02 8D EE 17 20 9E
17D2 03 20 F0 03 4C 00 02

KIM
PH
OK
17C0 17D9
17C0 18 F8 A5 01 65 03 8D ED 17 A5 00 65 02 8D EE 17
17D0 20 9E 03 20 F0 03 4C 00 02
KIM
LH
OK
0000 01 32 56 94

KIM
G
OK17C0
5826
KIM
```

To load the program, keys L and H on the telex must be pressed after the word "KIM" appears. The telex now reports "OK", and the first address to be loaded with data (17C0) can be keyed in

(listing 2). A space is then given, the data for this address is keyed in, a space is given, data for the second address, and so on. The data can be keyed in one after another; the monitor program itself ensures that the data ends up at the correct address. If a mistake is made while typing, it can be corrected by keying in the data again, without a space. If the line is full, the "carriage return" key should be used. The telex goes to the start of a new line, after which a new address can optionally be typed in. If this is not necessary, only the space key needs to be pressed. The telex then displays the address of the next empty memory location, followed by a space. The data for this memory location can then be entered, and so on. The first five lines of listing 2 show the loading of the program. Pressing the "line feed" key interrupts the monitor program and restarts it. This is necessary in order to be able to choose a different mode after loading.
On line four of the listing a mistake was made (a space was forgotten). This was corrected by keying in the data (17) again. A space must always be given after entering data, including at the end of a line!
Pressing keys P and H after "KIM" causes the entered program to be printed out. To do this, the start address, a space, and the end address must be keyed in after "OK". The end address can be found after loading by pressing the "carriage return" and "space" keys after entering the last piece of data. The telex then shows the end address. After keying in the end address, a "carriage return" must follow, after which the telex prints the program. Up to sixteen bytes per line are shown, with the address of the first byte of that line at the start of the line. After printing, the program returns to START.
The printout is also shown in listing 2. From this it follows that the mistake made during loading has indeed been corrected.

Now the two numbers for which the addition program is to calculate the sum still need to be entered. This can be done after "LH" and the address "0000" by simply keying them in, in groups of two digits (with spaces).

The program is run by pressing the G key after the telex has printed "KIM", and then, after "OK" (this time no line feed follows), typing in the start address (17C0).
Pressing the "carriage return" key starts the program. In this case the result is printed (5826), because the ADR subroutine is called within the program. The program then returns to the monitor program.

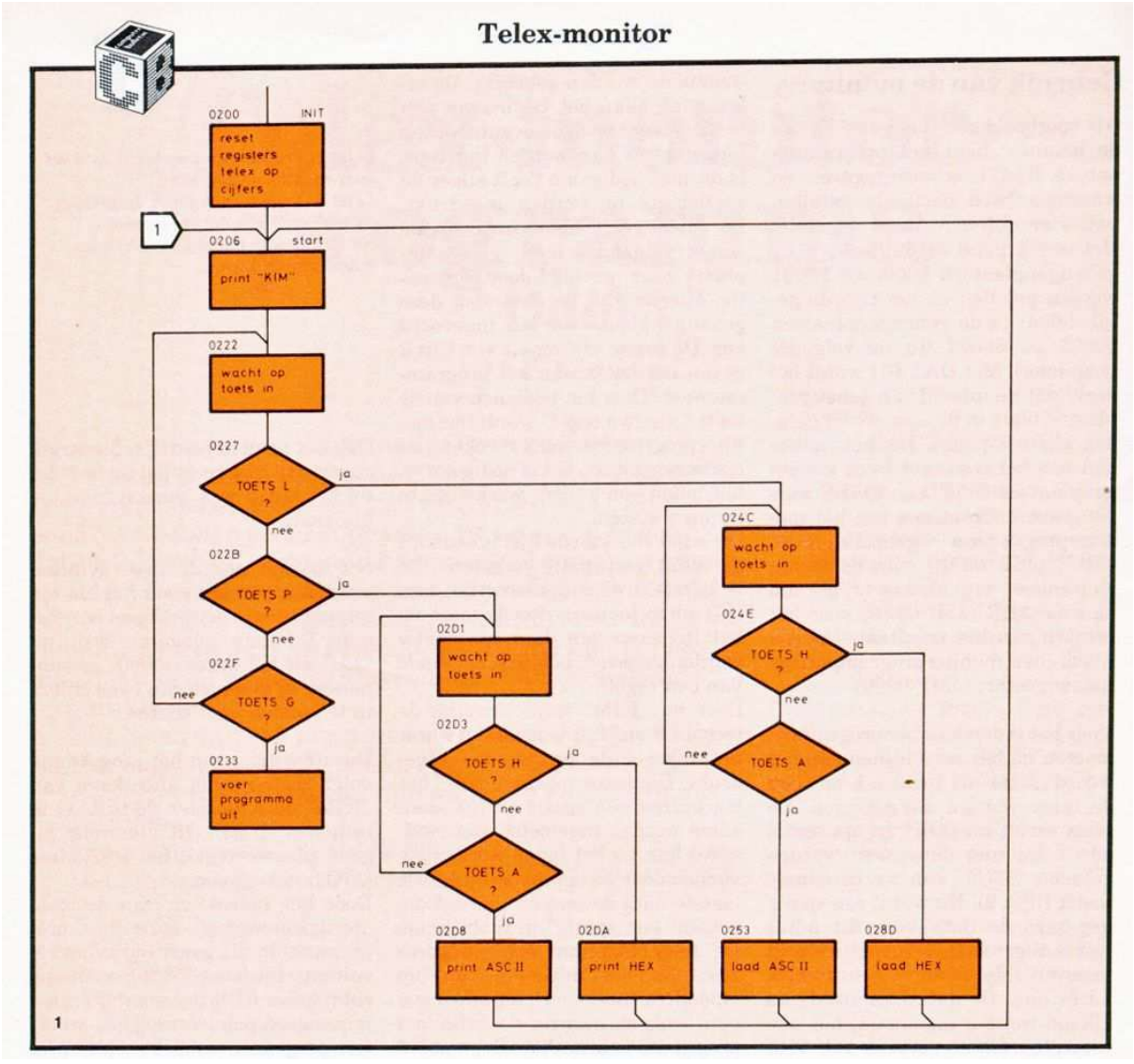
A given piece of text, written using the characters available on the keyboard, can also be loaded into memory. To do this, keys L and A must be chosen respectively. After "OK", the first address must again be keyed in, and after a space one can then start typing the text directly - or otherwise after a carriage return. Listing 3 shows the sequence of events. Here a mistake was made in the word "MONITORPROGRAMMA". This mistake was corrected by using the "BEL" key as many times as the number of characters (including punctuation) that need to be gone back. To print the text, keys P and A must be used respectively. Here too, the start and end address must be entered. After "carriage return" the text is printed. In this example, the correct end address was not entered, but a higher one. That the monitor nevertheless displayed the correct text is because, after the ASCII character for the last period of the text, it encountered an unrecognized character in the next memory location and stopped printing. Had the correct end address been entered, the monitor program would have restarted with the printing of "KIM". Here too it is possible, after loading, to find the end address. To do this, the program must first be restarted, and after pressing "KIM" the "LH" mode must be chosen. After "OK", the end address can then be made visible with a space.

Lijst 3

```
KIM
LA
OK
0000
KIM TELEX-MONITORPROGRAMMA.
ONGEVEER 2,5 PAG. GEHEUGENRUIMTE.
                  M.B.IMMERZEEL.

KIM
PA
OK
0000 0100

KIM TELEX-MONITORPROGRAMMA.
ONGEVEER 2,5 PAG. GEHEUGENRUIMTE.
                  M.B.IMMERZEEL.
```



Description of the program

Fig. 1 shows the flowchart of the main parts of the monitor program. Essentially this concerns the selection among the three main functions, namely loading memory locations (L), printing certain characters (P), and starting a program (G). A second selection during loading and printing determines whether the hexadecimal code system (H) or the ASCII code system (A) is used. After printing, the program returns to START. When loading, this must be done by pressing the "line feed" key.

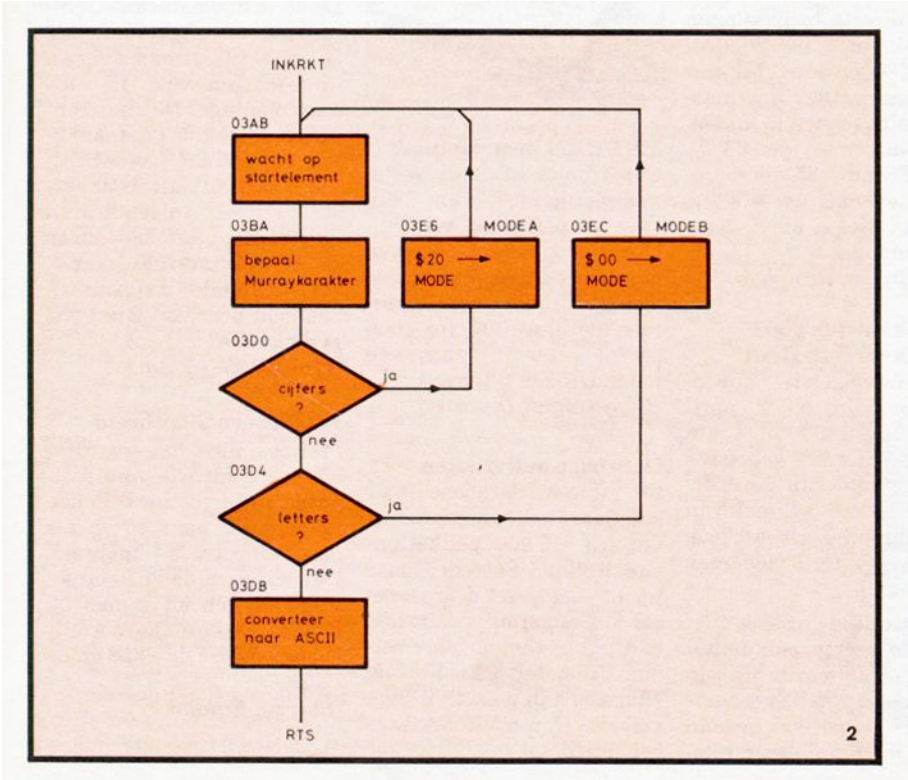
The code characters, as displayed by the telex transmitter, are characterized by being preceded by a currentless start element ("0"). "Waiting for a key press" therefore means nothing other than waiting for the input port, to which the telex transmit contact is connected, to become "zero". To account for any variations in the speed of the telex transmitter, the value of the input port must be determined at the middle of the code element. To do this, after the start of the start element, a wait of first 10 ms (03BA) and then another 20 ms takes place, in order to determine the value of the first code element (03BF). Fig. 2 shows the flowchart of the INKRKT subroutine. Here a loop is run through seven times to measure the five Baudot code elements plus two stop elements (03BF-03CA). The two stop elements are later shifted back out of the ACCU by two LSR instructions (03CE).
The ACCU contains one of the 32 (2^5) possible code characters. However, 64 are needed (digits and letters), and therefore the code characters that were preceded by "figures" are increased by $20 (03D8). The ACCU then contains a number that, for letters, can have a value between $00 and $1F, and for digits, a value between $20 and $3F. Listing 4 shows the corresponding characters and ASCII codes. This table is located at addresses $0100 through $013F. Using the TAY transfer, the (Abs,Y) addressing mode can be used to load the accumulator with the correct ASCII code (03DB).

Fig. 1 Flowchart of the main parts of the telex monitor program.

Fig. 2 Flowchart of the INKRKT subroutine.

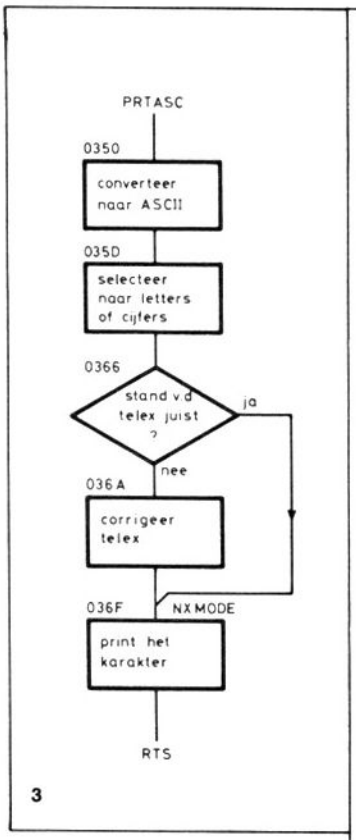
Listing 3 Loading, correcting, and printing ASCII characters using the telex monitor program.

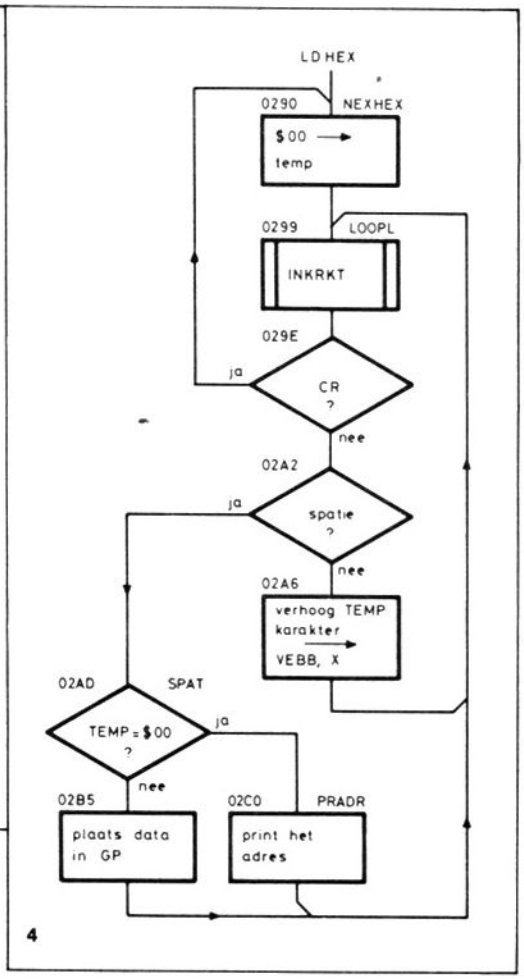
Listing 4 Overview of the Murray code characters with the corresponding characters and ASCII codes. The numbers shown in brackets do not belong to the ASCII code.



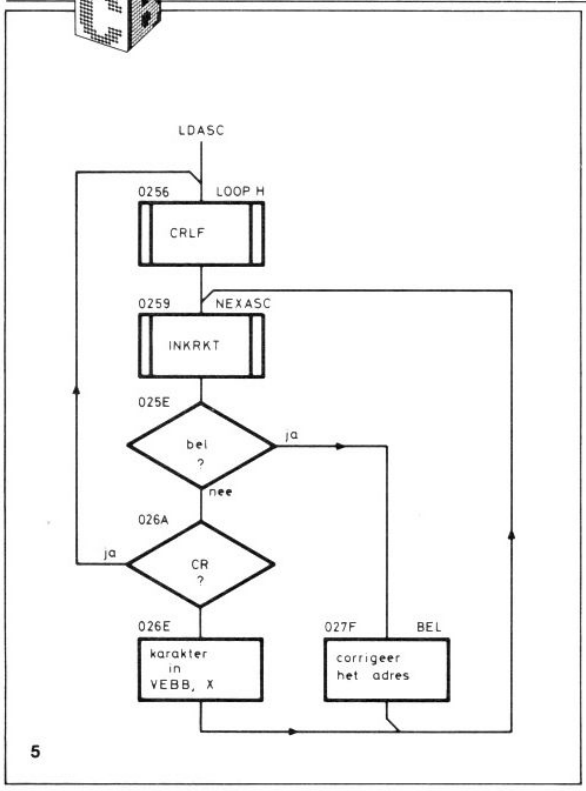
Murray code (Figures+$20) HEX	ASCII code HEX	Character
00	00	
01	54	T
02	0D	CR
03	4F	0
04	20	space
05	48	H
06	4E	N
07	4D	M
08	0A	LF
09	4C	L
0A	52	R
0B	47	G
0C	49	I
0D	50	P
0E	43	C
0F	56	V
10	45	E
11	5A	Z
12	44	D
13	42	B
14	53	S
15	59	Y
16	46	F
17	58	X
18	41	A
19	57	W
1A	4A	J
1B	(00)	figures
1C	55	U
1D	51	Q
1E	4B	K
1F	(FF)	letters
20	00	
21	35	5
22	0D	CR
23	39	9
24	20	space
25	(26)	
26	2C	.
27	2E	.
28	0A	LF
29	29	)
2A	34	4
2B	24	§
2C	38	8

2D	30	Ø
2E	3A	:
2F	3D	=
30	33	3
31	2B	+
32	(23)	#
33	3F	?
34	27	'
35	36	6
36	25	%
37	2F	/
38	2D	-
39	32	2
3A	07	BEL
3B	(00)	figures
3C	37	7
3D	31	1
3E	28	(
3F	(FF)	letters





Afb. 3 *Stroomdiagram van de subroutine PRTASC.*
Afb. 4 *Stroomdiagram van de subroutine LDHEX.*



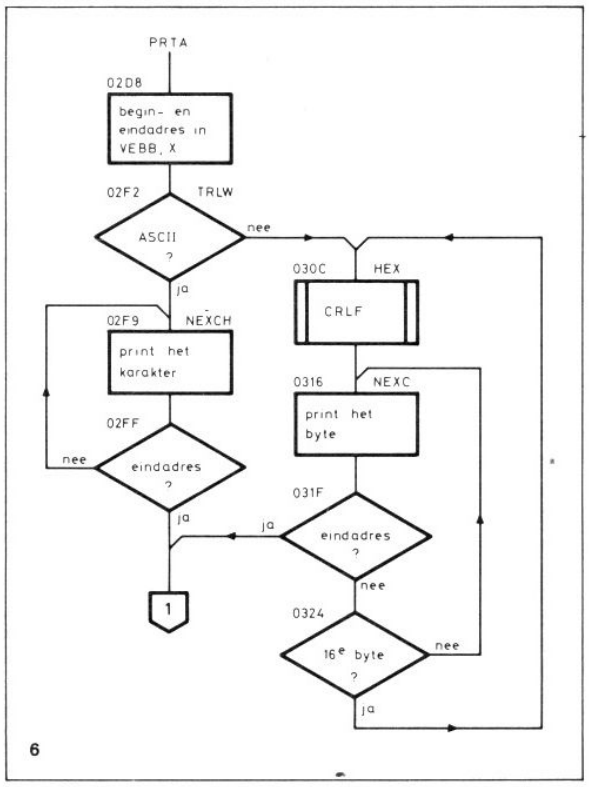


Fig. 3 Flowchart of the PRTASC subroutine.

Fig. 4 Flowchart of the LDHEX subroutine.

The INKRKT subroutine ensures that when a character is keyed in on the telex keyboard, the correct ASCII code is loaded into the ACCU. The reverse is also needed, namely that an ASCII code in the ACCU results in a character being printed by the telex. A conversion from ASCII to Murray is then needed. This conversion takes place in the PRTASC subroutine, the flowchart of which is given in fig. 3. To do this, the contents of the accumulator are compared with a code character from the table (listing 4) indicated by the contents of the X register. If they match, the contents of the X register determine the Murray code. If there is no match, another code character in the table is examined by giving X a different value. The subroutine starts with X=$3F (0350), and in the loop X is decremented by one each time until the correct value is found (0352-0358).

It must then first be determined whether the character should be preceded by a character for figures or for letters (035C). If the contents of the X register are equal to or greater than $20, it concerns a digit, and the contents of X must be decreased by $20 (0389). Using TXA and PHA, the character is saved on the STACK. If the telex is not currently in the correct mode (figures or letters), it is corrected. The mode the telex is in is indicated by the contents of memory location "MODE". The relevant character can then be printed. This is handled by the UITKT subroutine, in which the two stop bits are first added to the code character.

The "Load Hex" block from fig. 1 is worked out in more detail in fig. 4. The purpose of this part of the program is that the data being keyed in is loaded into the relevant memory location at each

space. After printing "OK" and returning to a new line, two registers are reset: TEMP at address $00FC and TEMPX at address $00FD. The hexadecimal digits now being keyed in are shifted by the LD subroutine into memory locations $17ED and $17EE. This happens digit by digit, so that after four digits have been entered, the most significant byte is loaded into $17EE and the least significant byte into $17ED. The (Abs,X) addressing mode is used here, with the contents of memory location TEMPX taken over by the X register (still $00 at this point). This is the loading of the address.

After the following space, the contents of TEMP are still zero, and the contents of TEMPX become $03 (02CB). As a result, the following hexadecimal digits are shifted into memory locations $17F0 and $17F1 ($17ED + $03 = $17F0). This is the entry of the data, and only memory location $17F0 is relevant here. On the next space, its contents are loaded into the addressed memory location using a subroutine VEB, in the same way as in the KIM monitor program. The address is also incremented by 1. When "carriage return" is keyed in, the whole process starts again from the beginning. If the first character given is a space, the address stored in memory locations $17ED and $17EE is printed, after which the new data can be entered directly (TEMPX becomes $03).
As shown in fig. 5, when loading ASCII characters the first address is likewise formed by shifting the previously keyed-in hex codes (the LD subroutine accepts no other codes) into addresses $17EE and $17ED (026E). TEMPX is then zero (it is reset at START, 0266). Once a space has been given, each character is loaded into the correct place in memory by the LDKRKT subroutine, whereby, if it concerns an ASCII code for "carriage return", a code for "line feed" is also added. The code for "BEL" only results in the address being decreased, so that a mistake that has been made can be corrected (027F).

The first part of the program for printing "HEX" is the same as for printing "ASCII", and is given in fig. 6. The contents of TEMP ($00 or $01) later determine the relevant mode. First the start address is shifted into $17ED and $17EE (02ED), and after a space (TEMPX=$03, 02F9) the end address into $17F0 and $17F1. A "carriage return" (02E5) brings about the selection of "ASCII" or "HEX". In the case of "ASCII" (02F6), printing of the characters begins at the start of a new line and continues until the end address is reached. The program then returns to START (0302).
In the case of "HEX" (030C), the address is printed first (0313), then the contents of that address (031C), space, next byte, and so on, until 16 bytes have been printed (16 decimal=$10). A new line then follows, again with an address and 16 bytes, and so on, until the end address is reached. Printing of hexadecimal characters (subroutine PR) is done by first determining the ASCII codes for them and then applying the PRTASC subroutine.
The complete monitor program is given in listing 5.

Connecting the telex

Connecting the telex to the KIM is done in the same way as connecting the TTY. The transmit contact goes between points R and T of connector A. The output is point U of this connector (fig. 7). The receiver coil can be connected via a transistor to the 12 V supply of the KIM. The solenoid coil has a resistance of about 300 Ω, so that with a conducting transistor a current of about 40 mA flows through the coil. If the original telex power supply unit (40 mA current stabilizer) is used, a circuit can be used as described for the Morse-telexconvertor. In doing so, it must be taken into account that in the idle state (current-carrying situation) output U of the KIM is logic "0". A transistor as shown in fig. 7 can also be used. When choosing the transistor, however, it should be borne in mind that during the currentless periods the voltage across the transistor can rise to 120 V.

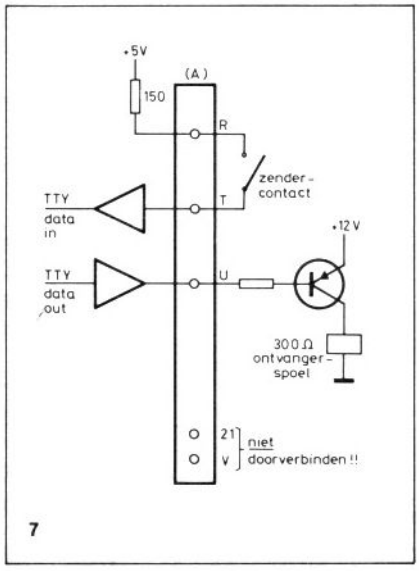


Fig. 5 Flowchart of the LDASC subroutine.

Fig. 6 Flowchart of the PRTA subroutine.

Fig. 7 Connection diagram of the telex and the KIM.

Listing 5 Telex monitor program for the 6502.

Lijst 5

```
            MICRO-WARE ASSEMBLER 65XX-1.0 PAGE 01

                 **************************************
                 *                                    *
                 *          SEELEX-MONITOR VOOR       *
                 *          DE KIM - 6502             *
                 *          APRIL 1981                *
                 *          AUTEUR: M. B. IMMERZEEL   *
                 *                                    *
                 **************************************

0140                         ORG    $0140

                 ZERO-PAGE ADRESSEN

0140             CHKHI  *    $00F6
0140             CHKSUM *    $00F7
0140             INL    *    $00F8
0140             TEMP   *    $00F9
0140             TMPX   *    $00FC
0140             MODE   *    $00FD

                 PIA-ADRESSEN

0140             SAD    *    $1740
0140             SBD    *    $1742

                 HOGE RAM-ADRESSEN

0140             VEB    *    $17EC
0140             VEBB   *    $17ED
0140             VEBL   *    $17EE
0140             VEBE   *    $17F0
0140             VEBF   *    $17F1

                 KIM-ROUTINES

0140             INTVEB *    $1932
0140             INCVEB *    $1A01
0140             PACKT  *    $1A00
0140             INITS  *    $1E88
0140             TAB    *    $1FD0

0140             TABLE  *    $0100

                 0100: 00 54 00 4F 20 40 4E 40
                       08 40 52 47 49 50 43 56
                 0110: 45 58 44 42 53 50 46 58
                       41 57 48 00 55 51 40 00
                 0120: 00 35 00 39 20 20 2E 2E
                       00 29 34 24 38 30 34 30
                 0130: 33 28 23 3F 27 30 20 2F
                       2D 22 07 00 37 31 20 11

                 SUBROUTINE VERHOOGT STARTADRES
                 EN VERGELIJKT DIT MET EINDADRES.

0140 20 EA 19    VEBIN  JSR   INCVEB VERHOOG ADRES
0143 AD ED 17           LDA   VEBB   VERGELIJK LAGE BYTE MET
0146 CD F0 17           CMP   VEBE   EINDADRES.GELIJK C=1
0149 AD EE 17           LDA   VEBL   VERGELIJK HOGE BYTE
014C ED F1 17           SBC   VEBF   GELIJK   C=1
014F 60                 RTS

                 SUBROUTINE LAADT EEN KARAKTER VAN
                 START- OF EINDADRES (X=$00 OF $03)

0150 20 00 1A    LD     JSR   PACKT  ASCII NAAR HEX
0153 D0 12              BNE   LOOPZ  GEEN HEX
0155 A6 FD              LDXZ  TMPX   START- OF EINDADRES
0157 0A                 ASLA         ISOLEER
0158 0A                 ASLA
0159 0A                 ASLA         KARAKTER
015A 0A                 ASLA
015B A0 04              LDYIM $04    VIER BITS
015D 0A          LOOPAA ASLA         BIT
015E 3E EC 17           ROLAX VEBB   NAAR
0161 3E EE 17           ROLAX VEBL   ADRES
0164 88                 DEY
0165 D0 F6              BNE   LOOPAA VOLGENDE BIT
0167 A0 00       LOOPZ  LDYIM $00    UNNITIAL BIT
0169 60                 RTS

                 SUBROUTINE WACHT 20 MS. HELE BIT(1200
                 OF 10 MS. HALVE BIT(1200)

016A A0 5E       DELAY  LDYIM $5E    VOOR 20 MS
016C A2 2A       DELAYH LDXIM $2A    WACHT 10 MS VANAF
016E CA          LOOPDD DEX
016F D0 FD              BNE   LOOPDD
0171 88                 DEY
0172 D0 F8              BNE   DELAYH
0174 60                 RTS

                 SUBROUTINE LAADT EEN KARAKTER IN
                 AANGEGEVEN ADRES

0175 C9 0D       LDKART CMPIM $0D    "TERUGLOOPWAGEN"
0177 D0 0B              BNE   LOOPCC NEEN
0179 20 EC 17           JSR   VEB    LAAD KARAKTER IN ADRES
017C 20 EA 19           JSR   INCVEB VERHOOG ADRES
017F 20 9E 03           JSR   CRLF
0182 A9 0A              LDAIM $0A    GEEF NIEUWE REGEL
0184 20 EC 17    LOOPCC JSR   VEB    LAAD KARAKTER IN ADRES
0187 60                 RTS

                 HOOFDPROGRAMMA

0200                    ORG   $0200

0200 20 88 1E    INIT   JSR   INITS  ZET IN- EN UITGANGEN
0203 20 32 19           JSR   INTVEB INIT SUBR. VEB
0206 A9 00       START  LDAIM $00    RESET
0208 85 FC              STAZ  TEMP   DE
020A 85 FD              STAZ  TMPX   REGISTERS
020C 20 50 03           JSR   PRTASC "CIJFERS" (TTY) NAAR UIT
020F 20 9E 03           JSR   CRLF   MODE WORDT $20
0212 A2 05              LDXIM $05    PRINT
0214 86 F7       LOOPA  STXZ  CHKSUM "KIM"
0216 BD D4 1F           LDAAX TAB
0219 20 50 03           JSR   PRTASC
021C A6 F7              LDXZ  CHKSUM
021E CA                 DEX          VOLGENDE KARAKTER
021F D0 F3              BNE   LOOPA
0221 EA                 NOP
0222 20 A0 03    FKEY   JSR   INPRKT WACHT OP KARAKTER
0225 F0 DF              BEQ   START  RESTART
0227 C9 4C              CMPIM $4C    "L"
0229 F0 21              BEQ   CKEY
022B C9 50              CMPIM $50    "P"
022D F0 1A              BEQ   LOOPD
022F C9 47              CMPIM $47    "G"
0231 D0 EF              BNE   FKEY
0233 20 92 03           JSR   PRTCR  JA. PRINT "GO"
0236 A2 00              LDXIM $00
0238 20 A0 03    LOOPE  JSR   INPRKT
023B F0 C9              BEQ   START  RESTART
023D C9 0D              CMPIM $0D    "TERUGLOOPWAGEN"
023F F0 05              BEQ   LOOPD  ADRES GELADEN
0241 20 50 01           JSR   LD     HAAL KARAKTER VAN STARTADRES
0244 F0 F2              BEQ   LOOPE  VOLGENDE
0246 6C ED 17    LOOPD  JMI   VEBB   START OP DIT ADRES
0249 4C D1 02    LOOPG  JMP   SECKEY PRINTEN
024C A9 8D       CKEY   LDAIM $8D    INSTAL "STX" NAAR VEB
024E 20 2A 03           JSR   HEXASC X=$00.HEX(X=$03).ASCII
0251 F0 3A              BEQ   LDHEX  LAAD HEX KARAKTERS
0253 20 92 03    LDASC  JSR   PRTCR  LAAD ASCII KAR.
0256 20 9E 03    LOOPH  JSR   CRLF
0259 20 A0 03    NEXASC JSR   INPRKT
025C F0 A8       LOOPK  BEQ   START  RESTART
025E C9 07              CMPIM $07    "BEL"
0260 F0 1D              BEQ   BEL    CORRIGEER FOUT
0262 C9 20              CMPIM $20    "SPATIE"
0264 F0 0D              BEQ   SPATI  ADRES EERSTE KAR. GELADEN
0266 A6 FD              LDXZ  TMPX   $00.ADRES(ANDERS KAR. LADEN)
0268 D0 0D              BNE   LOOPG  KARAKTER LADEN
026A C9 0D              CMPIM $0D    "TERUGLOOPWAGEN"
```

```
026C F0 E8              BEQ    LOOPH  GEEF NIEUWE REGEL
026E 20 50 01           JSR    LD     LAAD EEN KAR. VAN ADRES
0271 F0 E6              BEQ    NEXASC VOLGENDE KAR.
0273 A2 01     SPATI    LDXIM  $01    ADRES GELADEN
0275 86 FD              STXZ   TMPX   BEWAAR KARAKTER
0277 20 75 01  LOOPG    JSR    LDKRKT LAAD KARAKTER
027A 20 EA 19           JSR    INCVEB VERHOOG ADRES
027D D0 DA              BNE    NEXASC VOLGENDE
027F AD ED 17  BEL      LDA    VEBB   VERLAAG
0282 D0 03              BNE    LOOP1  HET
0284 CE EE 17           DEC    VEBC   ADRES
0287 CE ED 17  LOOP1    DEC    VEBB
028A 4C 59 02           JMP    NEXASC VOLGENDE
028D 20 92 03  LDHEX    JSR    PRTOK  LAAD HEX KAR.
0290 20 9E 03  NEXHEX   JSR    CRLF
0293 A2 00              LDXIM  $00    RESET
0295 86 FC              STXZ   TEMP   DE
0297 86 FD              STXZ   TMPX   REGISTERS
0299 20 AB 03  LOOPL    JSR    INKRKT
029C F0 BE     LOOPP    BEQ    LOOPK  RESTART
029E C9 0D              CMPIM  $0D    "TERUGLOOPWAGEN"?
02A0 F0 EE              BEQ    NEXHEX NIEUWE REGEL
02A2 C9 20              CMPIM  $20    "SPATIE"?
02A4 F0 07              BEQ    SPAT   PRINT ADRES OF LAAD
02A6 E6 FC              INCZ   TEMP   NIET 0. LAAD HEX
02A8 20 50 01           JSR    LD     TMPX=0.LAAD ADRES;NIET 0.LAAD HEX
02AB F0 EC              BEQ    LOOPL  VOLGENDE
02AD A4 FC     SPAT     LDYZ   TEMP
02AF F0 0F              BEQ    PRADR  PRINT ADRES AAN BEGIN
02B1 A6 FD              LDXZ   TMPX
02B3 F0 16              BEQ    CONTC  ADRES GELADEN
02B5 AD F0 17           LDA    VEBE   HEX DATA NAAR ACCU
02B8 20 EC 17           JSR    VEB    LAAD OP DIT ADRES
02BB 20 EA 19           JSR    INCVEB VERHOOG ADRES
02BE D0 D9              BNE    LOOPL  VOLGENDE
02C0 A9 0D     PRADR    LDAIM  $0D    "TERUGLOOPWAGEN"
02C2 20 50 03           JSR    PRTASC BEGIN VAN REGEL
02C5 20 F0 03           JSR    ADR    PRINT ADRES
02C8 20 A7 03           JSR    HBIT   PRINT "SPATIE"

02CB A2 03     CONTC    LDXIM  $03    KARAKTERS
02CD 86 FD              STXZ   TMPX   NAAR VEBE
02CF D0 C8              BNE    LOOPL  VOLGENDE
02D1 A9 AD     SECKEY   LDAIM  $AD    INSTR. "LDA" NAAR VEB
02D3 20 2A 03           JSR    HEXASC X=$00.HEX;X=$01.ASCII
02D6 F0 02              BEQ    PRTH   PRINT HEX
02D8 E6 FC     PRTA     INCZ   TEMP   $00.HEX;ANDERS ASCII
02DA 20 92 03  PRTH     JSR    PRTOK
02DD 20 9E 03  NEXA     JSR    CRLF
02E0 20 AB 03  LOOPN    JSR    INKRKT
02E3 F0 B7     LOOPO    BEQ    LOOPP  RESTART
02E5 C9 0D              CMPIM  $0D    "TERUGLOOPWAGEN"?
02E7 F0 09              BEQ    TRLW   ADRESSEN GELADEN
02E9 C9 20              CMPIM  $20    "SPATIE"?
02EB F0 19              BEQ    CONTD  EINDE STARTADRES
02ED 20 50 01           JSR    LD     LAAD KAR. START EINDADRES
02F0 F0 EE              BEQ    LOOPN  VOLGENDE
02F2 A4 FC     TRLW     LDYZ   TEMP   =$00.HEX;ANDERS ASCII
02F4 F0 16              BEQ    HEX    PRINT HEX
02F6 20 9E 03  ASCII    JSR    CRLF
02F9 20 EC 17  NEXCH    JSR    VEB
02FC 20 50 03           JSR    PRTASC CONVERTEER NAAR MURRAY
02FF 20 40 01           JSR    VEBIN  VERHOOG EN VERGELIJK MET EINDADRES
0302 B0 98              BCS    LOOPP  LAATSTE KARAKTER.RESTART
0304 90 F3              BCC    NEXCH  VOLGENDE
0306 A2 03     CONTD    LDXIM  $03    EIND-
0308 86 FD              STXZ   TMPX   ADRES
030A 10 D4              BPL    LOOPN
030C A0 10     HEX      LDYIM  $10    PRINT HEX
030E 84 FC              STYZ   TEMP   16 BYTES OP REGEL
0310 20 9E 03           JSR    CRLF
0313 20 F0 03           JSR    ADR    PRINT ADRES
0316 20 A7 03  NEXC     JSR    HBIT   PRINT "SPATIE"
0319 20 EC 17           JSR    VEB    LAAD BYTE UIT ADRES IN ACCU
031C 20 3C 03           JSR    PR     PRINT BYTE
031F 20 40 01           JSR    VEBIN  VERHOOG EN VERGELIJK EINDADRES
0322 B0 BF              BCS    LOOPO  LAASTE BYTE? RESTART
0324 C6 FC              DECZ   TEMP
0326 D0 EE              BNE    NEXC   VOLGENDE BYTE
0329 F0 E2              BEQ    HEX    VOLGENDE REGEL

               SUBROUTINE BEPAALT HEX OF ASCII

032A 8D EC 17  HEXASC   STA    VEB    INIT VEB
032D 20 AB 03  LOOPR    JSR    INKRKT
0330 A2 00              LDXIM  $00    X=$00 VOOR HEX
0332 C9 48              CMPIM  $48    "H"?
0334 F0 05              BEQ    RTSA   VOOR HEX
0336 C9 41              CMPIM  $41    "A"?
0338 D0 F3              BNE    LOOPR  VERKEERDE AANSLAG
033A E8                 INX           X=$01 VOOR ASCII
033B 60        RTSA     RTS

               SUBROUTINE PRINT EEN BYTE

033C 48        PR       PHA           BEWAAR BYTE
033D 4A                 LSRA          HAAL
033E 4A                 LSRA          DE
033F 4A                 LSRA          HOGE
0340 4A                 LSRA          NIBBLE
0341 20 47 03           JSR    PRI    PRINT HOGE NIBBLE
0344 68                 PLA           HAAL BYTE TERUG
0345 29 0F              ANDIM  $0F    HAAL LAGE NIBBLE
0347 18        PRI      CLC
0348 69 30              ADCIM  $30    HEX NAAR ASCII
034A C9 3A              CMPIM  $3A    CIJFERS OF LETTERS?
034C 30 02              BMI    PRTASC CIJFERS
034E 69 06              ADCIM  $06    CORRIGEER VOOR ASCII

               ROUTINE PRINT EEN ASCII KARAKTER

0350 A2 3F     PRTASC   LDXIM  $3F    ZOEK IN TABEL NAAR
0352 DD 00 01  CONTA    CMPAX  TABEL  ASCII KARAKTER
0355 F0 05              BEQ    MURR   MURRAY-CODE IN X
0357 CA                 DEX
0358 10 F8              BPL    CONTA
035A 30 F4              BMI    PRTASC
035C 8A        MURR     TXA           MURRAY NAAR ACCU
035D C9 20              CMPIM  $20    CIJFERS OF LETTERS?
035F 10 28              BPL    CIFF   CIJFERS
0361 48        LETT     PHA           BEWAAR MURRAY-CODE
0362 A9 1F              LDAIM  $1F    MURRAY VOOR "LETTERS"
0364 A2 00              LDXIM  $00    LETTERS
0366 E4 FF     XMODE    CPXZ   MODE   WELKE STAND HEEFT TELEX
0368 F0 05              BEQ    NXMODE STAND OK
036A 86 FF              STXZ   MODE   NIET OK, CORRIGEER MODE
036C 20 70 03           JSR    UITKT  CORRIGEER TELEX
036F 68        NXMODE   PLA           HAAL MURRAY-CODE TERUG

               ROUTINE PRINT EEN MURRAY KARAKTER

0370 0A        UITKT    ASLA          MAAK
0371 0A                 ASLA          STOPELEMENT
0372 09 03              ORAIM  $03
0374 A2 08              LDXIM  $08    TOTAAL
0376 86 F6              STXZ   CHKHI  8 BITS
0378 A2 07     LOOPT    LDXIM  $07    VOOR UITGANGSPOORT
037A 0A                 ASLA          SCHUIF BIT IN CARRY
037B B0 01              BCS    LOOPS  BIT IS "1"
037D CA                 DEX           BIT IS "0"
037E 8E 42 17  LOOPS    STX    SBD    BIT NAAR UITGANG
0381 20 6A 01           JSR    DELAY  WACHT 20 MS
0384 C6 F6              DECZ   CHKHI  VOLGENDE BIT
0386 D0 F0              BNE    LOOPT  KLAAR?
0388 60                 RTS
0389 E9 20     CIFF     SBCIM  $20    CIJFERS. CORRIGEER MURRAY
038B 48                 PHA           BEWAAR MURRAY-CODE
038C A9 1B              LDAIM  $1B    "CIJFERS" IN MURRAY
038E A2 20              LDXIM  $20    VOOR CIJFERS IN MODE
0390 10 D4              BPL    XMODE  VOOR PRINTEN

               ROUTINE PRINT "OK"

0392 20 9E 03  PRTOK    JSR    CRLF
0395 A2 03              LDXIM  $03 *  "O" (MURRAY)
0397 20 5C 03           JSR    MURR   PRINT
039A A2 1E              LDXIM  $1E    "K" (MURRAY)
039C D0 BE              BNE    MURP   PRINT

               ROUTINE PRINT NIEUWE REGEL

039E A9 02     CRLF     LDAIM  $02    "TERUGLOOPWAGEN" (MURR.)
03A0 20 70 03           JSR    UITKT  NAAR UITGANG
03A3 A9 08              LDAIM  $08    "NIEUWE REGEL" (MURRAY)
03A5 D0 C9              BNE    UITKT  NAAR UITGANG

               ROUTINE PRINT SPATIE

03A7 A9 04     HBIT     LDAIM  $04    "SPATIE" (MURRAY)
03A9 D0 C5              BNE    UITKT  NAAR UITGANG

               SUBROUTINE WACHT OP STARTBIT EN
               ZET MURRAY-CODE OM IN ASCII

03AB A2 07     INKRKT   LDXIM  $07    5 BITS. 2 STOPBITS
03AD 86 F6              STXZ   CHKHI
03AF A9 00              LDAIM  $00    RESET
03B1 85 F8              STAZ   INL    INL
03B3 A9 80     LOOPW    LDAIM  $80    MASKEER
03B5 2D 40 17           AND    SAD    MEET INGANG
03B8 D0 F9              BNE    LOOPW  GEEN STARTBIT. POORT=1
03BA A0 2F              LDYIM  $2F    WACHT 10 MS
03BC 20 6C 01           JSR    DELAYH
03BF 20 6A 01  CONTB    JSR    DELAY  WACHT 20 MS
03C2 AD 40 17           LDA    SAD    MEET INGANG
03C5 0A                 ASLA          BIT NAAR
03C6 26 F8              ROLZ   INL    INL
03C8 C6 F6              DECZ   CHKHI  VOLGENDE BIT
03CA D0 F3              BNE    CONTB  KLAAR?
03CC A5 F8              LDAZ   INL    VERWIJDER
03CE 4A                 LSRA          DE
03CF 4A                 LSRA          STOPBITS
03D0 C9 1B              CMPIM  $1B    "CIJFERS"? (MURRAY)
03D2 F0 12              BEQ    MODEA
03D4 C9 1F              CMPIM  $1F    "LETTERS"? (MURRAY)
03D6 F0 14              BEQ    MODEB
03D8 18                 CLC
03D9 65 FF              ADCZ   MODE   CORRIGEER;MODE=$20.CIJF;=$00.LETT
03DB A8                 TAY           CONVERTEER
03DC B9 00 01           LDAAY  TABEL  NAAR ASCII
03DF C9 0A              CMPIM  $0A    "NIEUWE REGEL"?
03E1 D0 02              BNE    RTSB   OVERSLAAN
03E3 A9 00              LDAIM  $00    "NIEUWE REGEL"
03E5 60        RTSB     RTS           RESTART
03E6 A9 20     MODEA    LDAIM  $20    "CIJFERS" IN MODE
03E8 85 FF     LOOPX    STAZ   MODE
03EA 10 BF              BPL    INKRKT
03EC A9 00     MODEB    LDAIM  $00    "LETTERS" IN MODE
03EE 10 F8              BPL    LOOPX

               SUBROUTINE PRINT ADRES

03F0 A2 01     ADR      LDXIM  $01    VOOR HOOGSTE BYTE
03F2 8A        LOOPY    TXA           BEWAAR
03F3 48                 PHA           X
03F4 BD ED 17           LDAAX  VEBB   BYTE NAAR ACCU
03F7 20 3C 03           JSR    PR     PRINT EEN BYTE
03FA 68                 PLA           HAAL
03FB AA                 TAX           X TERUG
03FC CA                 DEX
03FD 10 F3              BPL    LOOPY  KLAAR?
03FF 60                 RTS
```