KIM TIMER AND CLOCK

HERMAN PERK

While working with the KIM I once encountered the problem that a clock had to be maintained during the execution of a program. Not for the KIM itself, but for a system with the TIM monitor, namely the BEM-bus-based 6502 system from the company Brutech, which is highly recommended for its high quality and well-thought-out design (hence the prices). At the end of 1978, however, both the KIM and the BEM system were still lacking 6522 VIAs with their versatile timers, unlike the SYM and AIM-65. Worse still, the TIM monitor uses the only timer available in the system for communication with the terminal. If I nevertheless wanted to use the 6530 timer for the clock, I had to write my own program for communication, based on the so-called software timing loops. I did so, since I no longer needed TIM itself for the result. As for the timer (those of the 6530 and 6532 are identical), the “obvious” operating instructions say nothing about a possible “free-running” mode as would be usual for a clock and as is described for the 6522.

It is possible to make a clock in ‘one-shot’ mode, but then the Non Maskable Interrupt (NMI) must not be used as in my case, because otherwise there is a possibility that the clock will unpredictably fall behind. There must not be an unknown delay between reading the timer and restarting it. (A clock made in this way is known at the KIM club in the Netherlands. Another complication was that the terminal communication (interrupt-based), which I had made myself with timing loops at 1200 baud, could not be interrupted for too long by the clock in1terrupt; the clock interrupt service routine had to be short. In the meantime I also had three other fairly long IRQ routines and an NMI routine, all of which had to be capable of being interrupted 11111by a short IRQ from the clock.

With the KIM I use, a booklet called ‘KIM Hints’ was included when I bought it (also published in its entirety in the publication ‘KIM Kenner’ of the aforementioned KIM Club). In it I found a clue to the solution, namely in the printed listing of the example for using the interval timer, where the following comment line appears:

WHEN THE TIMER IS READ THE DIVIDER IS RESTORED TO ITS ORIGINAL VALUE AND THE INTERRUPT IS RESET

and above it:

TRE = S170E READ TIME ENABLE INT

(I recommend that those who have difficulties with the KIM timer track down the article in question and study it carefully. It contains clearer operating instructions.)

What seemed possible was, after the interrupt, simply to read the timer (this gives the one's complement of the time in clock periods that had elapsed since the interrupt), whereby the whole process starts again by itself with a known value, namely the value that was read. Thus, only a correction equal to the time elapsed between the interrupt and reading the timer would have to be made.

After keeping track of the time for several days, however, I found that the timer was even better than I had suspected. No correction was necessary! The basic counters of the timer turned out not to be reset by reading, as is the case when writing. I will not go into the hardware aspects of the timer any further. I refer to the manufacturer's literature from MOS Technology, Rockwell, Synertek and the booklet KIM Hints concerning the 6530 and 6532
(In Rockwell's product description of the 6532 there is another nice detail concerning the elapsed time between the interrupt and reading the timer. Subtract one from the incremented complement: thus, instead of two's complement, one's complement should be used in the 'one-shot' mode of the timer.)

The final result, adapted for publication, works as follows. During an initialization phase I start the timer by writing to S170F (division ratio 1024) with a 'random' number, e.g. SFF (in line 300). The timer countdown then takes 256 times 1024 clock periods, or a little more than a quarter of a second.
An interrupt then occurs, provided, of course, that PB7 of the 6530 is connected to the IRQ 'interrupt request line': application connector pin 15 to expansion connector pin 4. After an undetermined time, but less than 256 clock periods, I read the timer at address S170E. This restarts the timer with the value read, with division ratio 1024 (as KIM Hints indicates), and enables the interrupt. I now have information about how much time will elapse between the interrupt that has just occurred and the next interrupt! (after all, the correction is unnecessary). I add this information to a 16-bit counter TIMEX and leave the interrupt service routine (lines 1110 through 1300) with RTI.
The main program consists of the loop in lines 320 through 400 in which the display is read and the time is updated (and written to the KIM buffer),
Subroutine TIMKEP keeps track of the time. In my situation this was too long to put into the interrupt service routine, so I made it a separate subroutine, which I have to call from the main program each time before the interrupt service routine finds the 16-bit counter too small. Not all 16 bits may be used, because in lines 490 through 530 two of them are discarded.
'TIMER' is a 24-bit counter to which 1024 times "TIMEX" is added each time (490 through 590), and TIMEX is then reset (420 through 480). It is then compared with '$0F4240 (1000000), since there are $0F4240 clock periods in one second, and if at least one second has elapsed, $0FA240 is subtracted from "TIMER" (600 through 800). When this happens, the time is maintained decimally in seconds (TIME+02), minutes (TIME+01), hours (TIME) and in binary in days (DAGNO). This is done in lines 810 through 1090. (I also needed the date, hence this rudimentary section here.)

Setting the clock is done by filling "TIME" with the desired hour, minute and second using the monitor, like an ordinary digital 24-hour clock, and then starting at TEST (address $0200) or by pressing the ST button on the KIM keyboard. The latter, however, only if lines 150 through 220 have been executed at least once, because these set the vectors to the values required for this application. In most cases the clock will, after some time, no longer keep exact time (I had a loss of 13 seconds per day). This is due to the processor clock crystal not being adjusted exactly. The test loop in the interrupt routine (1140 through 1200) ensures a regular change of the seconds (the number that is read, and with which the timer is therefore set, becomes smaller). Here I have a routine that tests which peripheral device the interrupt originates from: device polling. For me it was not important that the seconds appeared to tick irregularly. This can be changed in another way by choosing a different division ratio. Lines 490 through 590 must then also be changed (they add 1024 times "TIMEX" to "TIMER").
The listing was made using MICRO-ADE, an Assembler, Disassembler, Editor combination,

which requires 8K of additional memory and a terminal in addition to the bare KIM.)
The clock program itself, however, runs on the bare KIM without additions.

```
0010:  0200                 TIJD    ORG     $0200
0020:
0030:  0200                 TIMKEN  *       $170F
0040:  0200                 TIMTRE  *       $170E
0050:
0060:  0200                 SCANS   *       $1F1F
0070:
0080:  0200                 TIME    *       $0000
0090:  0200                 DAGNO   *       $0003
0100:  0200                 TIMER   *       $0005
0110:  0200                 TIMTMP  *       $0008
0120:  0200                 TIMEX   *       $0009
0130:
0140:  0200  D8             TEST    CLD
0150:  0201  A9  B5                 LDAIM   INTERR
0160:  0203  8D  FE  17             STA     $17FE
0170:  0206  A9  02                 LDAIM   INTERR  /
0180:  0208  8D  FF  17             STA     $17FF
0190:  020B  A9  00                 LDAIM   TEST
0200:  020D  8D  FA  17             STA     $17FA
0210:  0210  A9  02                 LDAIM   TEST    /
0220:  0212  8D  FB  17             STA     $17FB
0230:  0215  A2  00                 LDXIM   $00
0240:  0217  86  05                 STX     TIMER
0250:  0219  86  06                 STX     TIMER   +01
0260:  021B  86  07                 STX     TIMER   +02
0270:  021D  86  09                 STX     TIMEX
0280:  021F  86  0A                 STX     TIMEX   +01
0290:  0221  CA                     DEX
0300:  0222  8E  0F  17             STX     TIMKEN
0310:  0225  58                     CLI
0320:  0226  20  1F  1F     LOOP    JSR     SCANS
0330:  0229  20  3B  02             JSR     TIMKEP
0340:  022C  A5  00                 LDA     TIME
0350:  022E  85  FB                 STA     $00FB
0360:  0230  A5  01                 LDA     TIME    +01
0370:  0232  85  FA                 STA     $00FA
0380:  0234  A5  02                 LDA     TIME    +02
0390:  0236  85  F9                 STA     $00F9
0400:  0238  4C  26  02             JMP     LOOP
0410:
0420:  023B  A0  00         TIMKEP  LDYIM   $00
0430:  023D  78                     SEI
0440:  023E  A5  0A                 LDA     TIMEX   +01
0450:  0240  A6  09                 LDX     TIMEX
0460:  0242  84  0A                 STY     TIMEX   +01
0470:  0244  84  09                 STY     TIMEX
0480:  0246  58                     CLI
0490:  0247  86  08                 STX     TIMTMP
0500:  0249  0A                     ASLA
0510:  024A  26  08                 ROL     TIMTMP
0520:  024C  0A                     ASLA
0530:  024D  26  08                 ROL     TIMTMP
0540:  024F  18                     CLC
0550:  0250  65  06                 ADC     TIMER   +01
0560:  0252  85  06                 STA     TIMER   +01
0570:  0254  A5  08                 LDA     TIMTMP
0580:  0256  65  05                 ADC     TIMER
0590:  0258  85  05                 STA     TIMER
0600:  025A  D8             TIMTST  CLD
0610:  025B  A5  05                 LDA     TIMER
0620:  025D  C9  0F                 CMPIM   $0F
0630:  025F  90  53                 BCC     EXTTIM
0640:  0261  D0  0E                 BNE     UPTIM
0650:  0263  A5  06                 LDA     TIMER   +01
0660:  0265  C9  42                 CMPIM   $42
0670:  0267  90  4B                 BCC     EXTTIM
0680:  0269  D0  06                 BNE     UPTIM
0690:  026B  A5  07                 LDA     TIMER   +02
0700:  026D  C9  40                 CMPIM   $40
0710:  026F  90  43                 BCC     EXTTIM
0720:  0271  A5  07         UPTIM   LDA     TIMER   +02
0730:  0273  E9  40                 SBCIM   $40
0740:  0275  85  07                 STA     TIMER   +02
0750:  0277  A5  06                 LDA     TIMER   +01
0760:  0279  E9  42                 SBCIM   $42
0770:  027B  85  06                 STA     TIMER   +01
0780:  027D  A5  05                 LDA     TIMER
0790:  027F  E9  0F                 SBCIM   $0F
0800:  0281  85  05                 STA     TIMER
0810:  0283  F8                     SED
0820:  0284  38                     SEC
0830:  0285  98                     TYA
0840:  0286  65  02                 ADC     TIME    +02
0850:  0288  85  02                 STA     TIME    +02
0860:  028A  C9  60                 CMPIM   $60
0870:  028C  90  CC                 BCC     TIMTST
0880:  028E  84  02                 STY     TIME    +02
0890:  0290  98                     TYA
0900:  0291  65  01                 ADC     TIME    +01
0910:  0293  85  01                 STA     TIME    +01
0920:  0295  C9  60                 CMPIM   $60
0930:  0297  90  C1                 BCC     TIMTST
0940:  0299  84  01                 STY     TIME    +01
0950:  029B  98                     TYA
0960:  029C  65  00                 ADC     TIME
0970:  029E  85  00                 STA     TIME
0980:  02A0  C9  24                 CMPIM   $24
0990:  02A2  90  B6                 BCC     TIMTST
1000:  02A4  84  00                 STY     TIME
1010:  02A6  D8                     CLD
1020:  02A7  98                     TYA
1030:  02A8  65  04                 ADC     DAGNO   +01
1040:  02AA  85  04                 STA     DAGNO   +01
1050:  02AC  98                     TYA
1060:  02AD  65  03                 ADC     DAGNO
1070:  02AF  85  03                 STA     DAGNO
1080:  02B1  4C  5A  02             JMP     TIMTST
1090:  02B4  60             EXTTIM  RTS
1100:
1110:  02B5  D8             INTERR  CLD
1120:  02B6  48                     PHA
1130:                           *** BEGIN TESTLOOPJE ***
1140:  02B7  8A                     TXA
1150:  02B8  48                     PHA
1160:  02B9  A2  05                 LDXIM   $05
1170:  02BB  CA             TESTLP  DEX
1180:  02BC  D0  FD                 BNE     TESTLP
1190:  02BE  68                     PLA
1200:  02BF  AA                     TAX
1210:                           ***EINDE TESTLOOPJE ***
1220:  02C0  AD  0E  17             LDA     TIMTRE
1230:  02C3  38                     SEC
1240:  02C4  65  0A                 ADC     TIMEX   +01
1250:  02C6  85  0A                 STA     TIMEX   +01
1260:  02C8  A9  00                 LDAIM   $00
1270:  02CA  65  09                 ADC     TIMEX
1280:  02CC  85  09                 STA     TIMEX
1290:  02CE  68                     PLA
1300:  02CF  40                     RTI
1310:
ID =

—T0
   SYMBOL TABLE 3000 3066
   DAGNO   0003   EXTTIM  02B4   INTERR  02B5   LOOP    0226
   SCANS   1F1F   TEST    0200   TESTLP  05BB   TIJD    0200
   TIME    0000   TIMER   0005   TIMEX   0009   TIMKEN  170F
   TIMKEP  023B   TIMTMP  0008   TIMTRE  170E   TIMTST  025A
   UPTIM   0271

—T1

   SYMBOL TABLE 3000 3066
   TIME    0000   DAGNO   0003   TIMER   0005   TIMTMP  0008
   TIMEX   0009   TEST    0200   TIJD    0200   LOOP    0226
   TIMKEP  023B   TIMTST  025A   UPTIM   0271   EXTTIM  02B4
   INTERR  02B5   TESTLP  02BB   TIMTRE  170E   TIMKEN  170F
   SCANS   1F1F
—
```