



```
;assemble with macro 80.
```

```
0000'          aseq
                org      100h
                .z80

0005          bios      equ      5                ;bios

                ;show time function for microbee with RTC kit
                ;
                ;register      contents          range
                ;
                ;0            seconds          0-60
                ;1            alarm seconds    0-60
                ;2            minutes          0-60
                ;3            alarm minutes    0-60
                ;4            hours            0-23
                ;5            alarm hours      0-23
                ;6            day of week      1-7
                ;7            day of month     1-30,31,29,28
                ;8            month            1-12
                ;9            year             0-99
                ;
                ;all outputs are in bcd

0100          C3 012A          JP          START
0103          1E 1B 59 20     DATA:   DB          1EH,1BH,'Y Time Of Day.COM D.BRE
0107          54 69 6D 65
010B          20 4F 66 20
010F          44 61 79 2E
0113          43 4F 4D 20
0117          44 2E 42 52
011B          45 45 5A 45
011F          20 31 38 2F
0123          31 32 2F 38
0127          35 20 1A

                                                    ;bit of a wank i

                ;=====
                ;main line of program for where all other routin

012A          3E 0A          start:   ld          a,0ah          ;point to clock
012C          D3 04          out          (4),a          ;set clock ale
012E          DB 07          in          a,(7)          ;read register
0130          CB 7F          bit          7,a          ;if bit 7 =1 the
0132          20 F6          jr          nz,start       ;wait for end of

0134          0E 04          ld          c,04          ;point to hours
0136          CD 01DB        call         input         ;get hours
0139          CD 01BB        call         numbers       ;print hours
013C          3E 3A          ld          a,':'         ;print ':'
013E          CD 01D0        call         output        ;/
0141          0E 02          ld          c,02          ;point to minute
0143          CD 01DB        call         input         ;get minutes
0146          CD 01BB        call         numbers       ;print minutes
```

```

0149      3E 3A          ld      a,':'          ;print ':'
014B      CD 01D0      call     output        ;/
014E      0E 00          ld      c,00          ;point to second
0150      CD 01DB      call     input         ;get seconds
0153      CD 01BB      call     numbers       ;print seconds
0156      3E 20          ld      a,' '         ;print ' '
0158      CD 01D0      call     output        ;/
015B      0E 06          ld      c,06          ;point to day of
015D      CD 01DB      call     input         ;get day of week
0160      21 01F7      ld      hl,day        ;point to day: t
0163      CD 019A      call     calcstr       ;print day
0166      3E 20          ld      a,' '         ;print ' '
0168      CD 01D0      call     output        ;/
016B      0E 07          ld      c,07          ;point to date r
016D      CD 01DB      call     input         ;get date
0170      CD 01BB      call     numbers       ;print date
0173      3E 20          ld      a,' '         ;print ' '
0175      CD 01D0      call     output        ;/
0178      0E 08          ld      c,08          ;point to month
017A      CD 01DB      call     input         ;get date
017D      21 023D      ld      hl,month      ;point hl to mon
0180      CD 019A      call     calcstr       ;print month
0183      21 02B5      ld      hl,year       ;point to ' 19'
0186      CD 01B0      call     outstr        ;print it
0189      0E 09          ld      c,09          ;point to year r
018B      CD 01DB      call     input         ;get year
018E      CD 01BB      call     numbers       ;print year
0191      21 02B9      ld      hl,crlf       ;print a crlf
0194      CD 01B0      call     outstr        ;
0197      C3 0000      jp      0              ;exit

```

```

;=====
;enters with b='tens' c='units' hl points to sta
;required

```

```

019A      79          calcstr:ld      a,c          ;put units in a
019B      CB 40          bit      0,b              ;test for greate
019D      28 02          jr      z,lessten        ;if more than te
019F      C6 0A          add     a,0ah            ;add ten to coun
01A1      47          lessten:ld      b,a          ;ld b with binar
01A2      05          dec     b                ;dec it by 1
01A3      AF          xor     a                ;clear a
01A4      C6 0A          repeat: add     a,0ah       ;table entry len
01A6      10 FC          djnz   repeat           ;repeat until a=
01A8      16 00          print: ld      d,0         ;clear d
01AA      5F          ld      e,a              ;put offset in e
01AB      19          add     hl,de            ;find address of
01AC      CD 01B0      call     outstr          ;send to con:
01AF      C9          ret

```

```

;=====
;enters with hl pointing to start of string to b

```

```

01B0      D5          outstr: push    de
01B1      C5          push   bc
01B2      EB          ex     de,hl          ;point de to string

```

```

01B3      0E 09          ld      c,9          ;bios fuction 'print str
01B5      CD 0005      call    bios         ;go and do it
01B8      C1           pop     bc
01B9      D1           pop     de
01BA      C9           ret

```

```

;=====
;enters with b='tens' c='units'

```

```

01BB      21 01ED      numbers:ld    hl,digits      ;point to '0'
01BE      16 00          ld      D,0          ;clear d
01C0      58           ld      e,b          ;put 'tens' in e
01C1      19           add     hl,de        ;point to first
01C2      7E           ld      a,(hl)       ;get the char
01C3      CD 01D0      call    output       ;output it
01C6      21 01ED      ld      hl,digits    ;point to '0'
01C9      59           ld      e,c          ;put 'tens' in e
01CA      19           add     hl,de        ;point to second
01CB      7E           ld      a,(hl)       ;get the char
01CC      CD 01D0      call    output       ;output it
01CF      C9           ret

```

```

;=====
;enters with char in a

```

```

01D0      C5           output: push    bc
01D1      D5           push    de
01D2      0E 02      ld      c,02h        ;bios function
01D4      5F           ld      e,a          ;char in e to con:
01D5      CD 0005      call    bios         ;do that
01D8      D1           pop     de
01D9      C1           pop     bc
01DA      C9           ret

```

```

;=====
;enters with c containing address of clock regis
;and sets up bc to contain the bcd result
;b contains the upper nibble
;c contains the lower nibble

```

```

01DB      79           input:  ld      a,c          ;put register num in a
01DC      D3 04      out     (4),a        ;set register address in
01DE      DB 07      in     a,(7)         ;read desired register
01E0      F5           push   af            ;save it on stack
01E1      E6 F0      and    0f0h          ;scrub lower nibble
01E3      1F           rra                    ;shift upper nibble down
01E4      1F           rra                    ; /
01E5      1F           rra                    ; /
01E6      1F           rra                    ; /
01E7      47           ld     b,a           ;store in b
01E8      F1           pop    af            ;restore a from stack
01E9      E6 0F      and    0fh           ;scrub upper nibble
01EB      4F           ld     c,a           ;store in c
01EC      C9           ret

```

```

;=====

```

## ;ascii tables

01ED	30 31 32 33	digits: db	'0123456789'
01F1	34 35 36 37		
01F5	38 39		
01F7	53 75 6E 64	day: db	'Sunday\$000'
01FB	61 79 24 30		
01FF	30 30		
0201	4D 6F 6E 64	db	'Monday\$000'
0205	61 79 24 30		
0209	30 30		
020B	54 75 65 73	db	'Tuesday\$00'
020F	64 61 79 24		
0213	30 30		
0215	57 65 64 6E	db	'Wednesday\$'
0219	65 73 64 61		
021D	79 24		
021F	54 68 75 72	db	'Thursday\$0'
0223	73 64 61 79		
0227	24 30		
0229	46 72 69 64	db	'Friday\$000'
022D	61 79 24 30		
0231	30 30		
0233	53 61 74 75	db	'Saturday\$0'
0237	72 64 61 79		
023B	24 30		
023D	4A 61 6E 75	month: db	'January\$00'
0241	61 72 79 24		
0245	30 30		
0247	46 65 62 72	db	'February\$0'
024B	75 61 72 79		
024F	24 30		
0251	4D 61 72 63	db	'March\$0000'
0255	68 24 30 30		
0259	30 30		
025B	41 70 72 69	db	'April\$0000'
025F	6C 24 30 30		
0263	30 30		
0265	4D 61 79 24	db	'May\$000000'
0269	30 30 30 30		
026D	30 30		
026F	4A 75 6E 65	db	'June\$00000'
0273	24 30 30 30		
0277	30 30		
0279	4A 75 6C 79	db	'July\$00000'
027D	24 30 30 30		
0281	30 30		
0283	41 75 67 75	db	'August\$000'
0287	73 74 24 30		
028B	30 30		
028D	53 65 70 74	db	'September\$'
0291	65 6D 62 65		
0295	72 24		

```
0297    4F 63 74 6F          db      'October$00'  
029B    62 65 72 24  
029F    30 30  
02A1    4E 6F 76 65          db      'November$0'  
02A5    6D 62 65 72  
02A9    24 30  
02AB    44 65 63 65          db      'December$0'  
02AF    6D 62 65 72  
02B3    24 30  
  
02B5    20 31 39 24          year:   db      ' 19$'  
02B9    0D 0A 24             crlf:   db      0dh,0ah,'$'
```

end

Macros:

Symbols:

0005	BIOS	019A	CALCSTR	02B9	CRLF
0103	DATA	01F7	DAY	01ED	DIGITS
01DB	INPUT	01A1	LESSTEN	023D	MONTH
01BB	NUMBERS	01D0	OUTPUT	01B0	OUTSTR
01A8	PRINT	01A4	REPEAT	012A	START
02B5	YEAR				

No Fatal error(s)

→

end