```
'0:MAIN.PRG      v0.2.0a, 7 августа 2022
'Операционная система ПМК

'Авторское право (c) Васильев Илья Владимирович, г. Москва
'Это свободная программа, защищённая авторским правом
'Распространяется под лицензией GNU GPL v3 или старше

OPTION STRICT 'Все переменные должны быть объявлены
OPTION DEFINT 'Массивы по умолчанию целые

EXEC "RING.PRG",1      'Общие переменные и работа с ними
EXEC "CPU.PRG",2       'Процессор, выполняющий команды ПМК
EXEC "MATH.PRG",3      'Десятичная математика
EXEC "STRINGS.PRG",4 'Работа с литерами и строками

CONST #Hex$="0123456789abcdef",#CR=CHR$(13)

'Цвета ПМК, приближеные к "Электронике МК-152"
CONST #CORANGE=RGB(254,80,0),#CRED=RGB(224,0,77),#CBLUE=RGB(0,61,165)
CONST #CGREEN=RGB(112,208,0),#CWHITE=RGB(217,217,214),#CBLACK=RGB(44,42,41)
CONST #CGOLD=RGB(255,215,0)

CONST #SCRX=22, #SCRY=22      'Координаты индикатора ПМК

VAR PC                        'Счётчик адреса (0..65535)
VAR STATUS$                   'Строка комментариев
VAR OP, PAR                   'Код операции и параметр многокнопочной команды

VAR FLGON                     '0-ВЫКЛ, 1-ВКЛ
VAR PREFIX                    '2-F, 4-P, только для режима HEX
VAR ERROR, ERRORS$[13]        'Номер ошибки, сообщения об ошибках
COPY ERRORS$,@Errors          'Считать сообщения об ошибках

DIM BOARD$[]                  'Клавиатурные раскладки и программы пульта
LOADV "DAT:BOARD",BOARD$      'Считать откомпилированную таблицу
VAR LAYOUT=VAL(BOARD$[1])     'Начальная активная раскладка клавиатуры F АВТ

DIM BSTACK[4], BSP=0          'Стек пульта МК-261 и его указатель

DIM POCKET[20], PSP=0         'Карман для вставки и удаления 20 шагов

XSCREEN 1280,720:GCLS         'Инициализировать GRAPHIC (GPAGE0)
TSCREEN #TCONSOLE,16,40,25    'Инициализировать CONSOLE (TEXT4)
TOFS #TCONSOLE,320,160

LOADG "SKIN",0                'Отобразить кожу ПМК
LOADG "FONT",5                'Загрузить шрифт МК-261

WarnAlpha                     'Предупредить о нестабильности альфа-версии

VAR MODE                      'Режим работы МК-261
ENUM #AVT=1,#PRG,#RUN,#PRG2 'Должны совпадать с раскладками клавиатуры


VAR BUTTONS[39,6], MATRIX[39], ABUTTON=38, DIR
COPY BUTTONS,@Keys            'Считать расположение и порядок кнопок

VAR KBDSTATE                  'Режимы цифровой клавиатуры
ENUM #KBDNORM=0,#KBDF,#KBDK,#KBDP,#KBD1,#KBD2,#KBD3,#KBD4,\
     #KBDPF,#KBDPK,#KBDPP,#KBDPPK,#KBDPKM,#KBDPPM


VAR I, J, TT, TX, TY          'Вспомогательные переменные


SPSET 0,0
FILL MATRIX,0                 'Обнулить матрицу клавиатуры
Reset
'       -123456789012345-67
'SetX$ "S123456789000000 00"
Refresh

'Главный цикл
LOOP
  'Считать матрицу клавиатуры с тачскрина
  FOR I=0 TO 9
    TOUCH I OUT TT,TX,TY
    IF TT>0 THEN
      FOR J=0 TO 38
        IF TX>=BUTTONS[J,0] && TX<BUTTONS[J,0]+160 && \
           TY>=BUTTONS[J,1] && TY<BUTTONS[J,1]+100 \
        THEN MATRIX[J]=MATRIX[J] OR 1:BREAK
      NEXT
    ENDIF
  NEXT

  'Нажать кнопки по их отпусканию
  FOR I=0 TO 38
    IF MATRIX[I]==2 THEN Press I
    IF (MATRIX[I] AND 1) == 0 THEN MATRIX[I]=0 ELSE MATRIX[I]=2
  NEXT

  'Выбрать кнопку джойстиком и клавиатурой
  DIR=0
  IF BUTTON(0,#B_LRIGHT,3)  || KEYBOARD(79,3) THEN DIR=2
  IF BUTTON(0,#B_LLEFT,3)   || KEYBOARD(80,3) THEN DIR=3
  IF BUTTON(0,#B_LUP,3)     || KEYBOARD(82,3) THEN DIR=4
  IF BUTTON(0,#B_LDOWN,3)   || KEYBOARD(81,3) THEN DIR=5
  IF DIR!=0 THEN ABUTTON=BUTTONS[ABUTTON,DIR]
  SPOFS 0,BUTTONS[ABUTTON,0]+80-16,BUTTONS[ABUTTON,1]+50-16
  IF BUTTON(0,#B_RRIGHT,3) || KEYBOARD(40,3) THEN Press ABUTTON
  VSYNC
ENDLOOP


'Разместить мнемонику в строке комментариев
DEF State N
  STATUS$=Mnemonic$(N)
END


'Выполнить многокнопочную команду
DEF Complete N
  INC PAR,N
  CASE OP
    WHEN &h40: 'П
      IF N!=15 THEN State OP+N:Do OP+N ELSE STATUS$="П f":DoM 15
    WHEN &h4f: 'Р П, РР П
      PUSH STATUS$,#Hex$[N]:DoM PAR
    WHEN &h51: 'БП, Р БП
      PUSH STATUS$,#Hex$[N]:PC=PAR
    WHEN &h60: 'ИП
      IF N!=15 THEN State OP+N:Do OP+N ELSE STATUS$="ИП f":DoRM 15
```

```
 0  2  4  6  8  10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74
PRG0  MAIN.PRG                                                                                    a  11 21
```

```
      WHEN &h60: 'ИП
        IF N!=15 THEN State OP+N:Do OP+N ELSE STATUS$="ип f":DoRM 15
      WHEN &h6f: 'Р ИП, РР ИП
        PUSH STATUS$,#Hex$[N]:DoRM PAR
      WHEN &hb0: 'КП
        IF N!=15 THEN State OP+N:Do OP+N ELSE STATUS$="к п f":DoKM 15
      WHEN &hbf: 'РК П, РРК П
        PUSH STATUS$,#Hex$[N]:DoPKM PAR
      WHEN &hd0: 'КИП
        IF N!=15 THEN State OP+N:Do OP+N ELSE STATUS$="к ип f":DoKRM 15
      WHEN &hdf: 'РК ИП, РРК ИП
        PUSH STATUS$,#Hex$[N]:DoPKRM PAR
    ENDCASE
    KBDSTATE=#KBDNORM
  END

  'Обработать нажатие кнопки N в раскладке LAYOUT таблицы BOARD$
  DEF Press N
    VAR A$=BOARD$[LAYOUT+N]
    IF A$!="" THEN
      VAR PC1=VAL(A$)
      LOOP
        A$=BOARD$[PC1]:INC PC1
        CASE A$[0]
          WHEN "$":  STATUS$=MID$(A$,1):Refresh  'Комментарий полностью
          WHEN "+":  PUSH STATUS$,MID$(A$,1):Refresh 'Комментарий
          WHEN "B":  CALL A$                     'Исполнить команду пульта
          WHEN "E":                              'Конец программы пульта
            IF A$=="ENDAVT" THEN
              IF MODE==#AVT THEN                 'В автоматическом режиме
                LAYOUT=VAL(BOARD$[#AVT])         'сбросить префикс
                STATUS$="":RefreshAVT
                BREAK                            'и закончить
              ENDIF
            ELSE
              IF A$=="END0" THEN LAYOUT=VAL(BOARD$[MODE]) 'Сбросить префиксы
              BREAK
            ENDIF
          WHEN "J":  PC1=VAL(MID$(A$,1))         'Безусловный переход
          OTHERWISE:
            BSP=(BSP+1)AND 3:BSTACK[BSP]=VAL(A$) 'Число в стек
        ENDCASE
      ENDLOOP
    ENDIF
  END

  'Ввести строку
  DEF StrInput$()
    VAR A$
    GFILL 300,140,979,579,#CBLACK
    GBOX 310,150,969,569,#CGOLD
    CLS
    LINPUT A$
    CLS
    LOADG "SKIN",0
    RETURN A$
  END

  'Элемент стека пульта, находящийся под вершиной
  DEF Second()
    IF BSP>0 THEN RETURN BSTACK[BSP-1] ELSE RETURN BSTACK[3]
  END

  'Сложение в стеке пульта
  DEF BrdAdd
    IF BSP>0 THEN
      INC BSTACK[BSP-1],BSTACK[BSP]:DEC BSP
    ELSE
      INC BSTACK[3],BSTACK[0]:BSP=3
    ENDIF
  END

  'Ввод очередной шестнадцатеричной цифры
  DEF BrdDigit
    VAR N=BSTACK[BSP]
    IF BSP>0 THEN DEC BSP ELSE BSP=3
    N=(BSTACK[BSP]<<<4)OR N
    BSTACK[BSP]=N
    IF MODE==#PRG2 THEN STATUS$="" ELSE STATUS$="P "
    PUSH STATUS$, ToLower$(Hex$(N>>>16,4))+":"+ToLower$(Hex$(N AND 65535,4))
    Refresh
  END

  'Считать режим: 0 АВТ, -1 ПРГ
  DEF BrdGetMode
    IF BSP==3 THEN BSP=0 ELSE INC BSP
    IF MODE==#AVT THEN BSTACK[BSP]=0 ELSE BSTACK[BSP]=-1
  END

  'Установить префикс в режиме HEX: 2-F, 4-P
  DEF BrdSetPrefix
    PREFIX=BSTACK[BSP]
  END

  'Считать префикс в режиме HEX: 2-F, 4-P
  DEF BrdGetPrefix
    IF BSP==3 THEN BSP=0 ELSE INC BSP
    BSTACK[BSP]=PREFIX
  END

  'Шаг вперёд
  DEF BrdForward
    PC=(PC+1)AND 65535:Refresh
  END

  'Шесть шагов вперёд
  DEF BrdForward6
    PC=(PC+6)AND 65535:Refresh
  END

  'Шаг назад
  DEF BrdBackward
    PC=(PC+65535)AND 65535:Refresh
  END
```

```
  PC=(PC+65535)AND 65535:Refresh
END

'Шесть шагов назад
DEF BrdBackward6
  PC=(PC+65530)AND 65535:Refresh
END

'Переключить точку останова
DEF BrdBrkP
  VAR("1:PrgM")[PC]=VAR("1:PrgM")[PC] XOR &h8000000
END

'Ввести гекс
DEF BrdHexEnter
  VAR N=BSTACK[BSP]
  STATUS$=""
  CASE MODE:
    WHEN #AVT:
      DoGex N:RefreshAVT
    WHEN #PRG: WHEN #PRG2:
      StrUnlink VAR("1:PrgM")[PC] 'Забываем строку
      VAR("1:PrgM")[PC]=N:BrdForward
  ENDCASE
END

'Ввести строку
DEF BrdStrEnter
  VAR A$=StrInput$()
  STATUS$=StrQ$(A$)
  CASE MODE:
    WHEN #AVT:
      DoStr A$:RefreshAVT
    WHEN #PRG: WHEN #PRG2:
      StrUnlink VAR("1:PrgM")[PC] 'Забываем строку
      VAR("1:PrgM")[PC]=StrLink(A$):BrdForward
  ENDCASE
END

'Выполнить или запомнить команду ПМК, введённую с пульта
DEF BrdDo
  VAR N=BSTACK[BSP] 'КОП
  State N
  CASE MODE
    WHEN #AVT:
      Do N:RefreshAVT
    WHEN #PRG:
      StrUnlink VAR("1:PrgM")[PC] 'Забываем строку
      VAR("1:PrgM")[PC]=N OR &hFFFFFF00:BrdForward
  ENDCASE
END

'Команда с пульта "двухшаговая"
DEF BrdDo2
  VAR COP=Second(), OP=BSTACK[BSP], N, COP2 'КОП, операнд, шаг
  CASE MODE
    WHEN #AVT: Do2 COP,OP:RefreshAVT
    WHEN #PRG:
      CASE COP
        WHEN &h100: 'X БП
          N=VAR("1:PrgM")[PC]
          IF (N AND &hf7ff0000)==&hf7ff0000 THEN
            N=(N AND &hff000000)OR((N AND 255)<<<16)OR OP
          ELSE
            N=(N AND &hffff0000)OR OP
          ENDIF
          VAR("1:PrgM")[PC]=N
        OTHERWISE:
          StrUnlink VAR("1:PrgM")[PC] 'Забываем строку
          VAR("1:PrgM")[PC]=(COP<<16) OR &hFF000000 OR OP
      ENDCASE 'F ПРГ
      BrdForward
  ENDCASE
END

'Изменить раскладку клавиатуры
DEF BrdSetLayout
  LAYOUT=VAL(BOARD$[BSTACK[BSP]])
END

'Переключиться на следующую раскладку
DEF BrdNextLayout
  INC LAYOUT,39
END

'Изменить режим работы калькулятора
DEF BrdSetMode
  MODE=BSTACK[BSP]:Refresh
END

'Включить ПМК с пульта
DEF BrdTurnOn
  Reset:Refresh
END

'Выключить ПМК с пульта
DEF BrdTurnOff
  FLGON=0:DisplayClear
END

'Очистка В/О
DEF BrdRTN
  State &h52:PC=0:RefreshAVT
END

'Вставить шаг из кармана
DEF BrdInsert
  VAR I, Pmin=PC, Pmax=65535, A, B
  VAR Anew=POCKET[PSP] 'Считать шаг из кармана
  POCKET[PSP]=-1:PSP=(PSP+19)MOD 20
  FOR I=PC TO 65535    'Найти конец перемещаемой области
    IF VAR("1:PrgM")[I]==-1 THEN Pmax=I:BREAK
```

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74

PRG0 MAIN.PRG a 11:21

```
  FOR I=PC TO 65535      'Найти конец перемещаемой области
    IF VAR("1:PrgM")[I]==-1 THEN Pmax=I:BREAK
  NEXT
  Relocate 0,PC-1,Pmin,Pmax,1  'Обновить ссылки до перемещаемой области
  FOR I=Pmax TO Pmin+1 STEP -1 'Переместить область
    A=VAR("1:PrgM")[I-1]
    IF IsRefP(I-1,Pmin,Pmax) THEN 'Ссылки в ней обновить
      B=(A AND 65535)+1
      A=(A AND &hffff0000) OR (B AND 65535)
    ENDIF
    VAR("1:PrgM")[I]=A
  NEXT
  VAR("1:PrgM")[Pmin]=Anew          'Вставить шаг из кармана
  Relocate Pmax+1,65535,Pmin,Pmax,1  'Обновить ссылки после области
END

'Убрать шаг в карман
DEF BrdDelete
  VAR I, Pmin=PC+1, Pmax=65535, A, B
  PSP=(PSP+1)MOD 20
  StrUnlink POCKET[PSP]          'Забываем строку
  POCKET[PSP]=VAR("1:PrgM")[PC]  'Сохранить шаг в карман
  FOR I=Pmin TO 65535            'Найти конец перемещаемой области
    IF VAR("1:PrgM")[I]==-1 THEN Pmax=I:BREAK
  NEXT
  Relocate 0,PC-1,Pmin,Pmax,-1   'Обновить ссылки до перемещаемой области
  FOR I=Pmin TO Pmax             'Переместить область
    A=VAR("1:PrgM")[I]
    IF IsRefP(I,Pmin,Pmax) THEN 'Ссылки в ней обновить
      B=A AND 65535
      A=(A AND &hffff0000) OR (B-1)
    ENDIF
    VAR("1:PrgM")[I-1]=A
  NEXT
  Relocate Pmax+1,65535,Pmin,Pmax,-1 'Обновить ссылки после области
END

'Включить ПМК
'- сохранять программу (память и строки) и регистры при выключении
DEF Reset
  PrgClear:RegClear:StackClear
  FILL POCKET,-1
  PC=0
  MODE=#AVT
  STATUS$="":KBDSTATE=#KBDNORM:ERROR=0
  FLGON=1      'Включить ПМК
END

'Очистить индикатор
DEF DisplayClear
  GFILL #SCRX,#SCRY,#SCRX+128*4-1,#SCRY+64*4-1,#CGREEN
END

'Вывести число из регистра стека Y, Z и T
DEF StackPrint Y,A$
  VAR S0$, S$, E0$, E$

  DecryptDec A$ OUT S0$,S$,E0$,E$
  CASE S0$
    WHEN "#":
      GPUTCHR  #SCRX+21*4,Y, S$,16,2,2,#CBLACK,#G_NORMAL2
    WHEN "$": '+++ можно выводить больше литер
      GPUTCHR  #SCRX+21*4,Y, StrQ$(S$),16,2,2,#CBLACK,#G_NORMAL2
    WHEN " ": WHEN "G": WHEN "M": WHEN "S":
    WHEN "-": WHEN "g": WHEN "m": WHEN "s":
      IF INSTR("-gms",S0$)>=0 THEN S0$="-" ELSE S0$=" "
      IF LEN(S$)<10 THEN
        GPUTCHR  #SCRX+13*4,Y,S0$,16,2,2,#CBLACK,#G_NORMAL2
        GPUTCHR  #SCRX+21*4,Y,S$, 16,2,2,#CBLACK,#G_NORMAL2
      ELSEIF LEN(S$)<14 THEN ' Переходим на пропорциональный шрифт
        GPUTCHR  #SCRX+13*4,Y,S0$,16,2,2,#CBLACK,#G_NORMAL2
        GPUTCHRP #SCRX+21*4,Y,S$, 16,2,2,#CBLACK,#G_NORMAL2
      ELSE ' Уменьшаем кегль
        GPUTCHR  #SCRX+17*4,Y,S0$,16,1,2,#CBLACK,#G_NORMAL2
        GPUTCHR  #SCRX+21*4,Y,S$, 16,1,2,#CBLACK,#G_NORMAL2
      ENDIF
      GPUTCHR #SCRX+103*4,Y,E0$,16,2,2,#CBLACK,#G_NORMAL2
      GPUTCHR #SCRX+111*4,Y,E$, 16,2,2,#CBLACK,#G_NORMAL2
  ENDCASE
END


'Обновить экран автоматического режима
'- можно считать заготовку из файла картинки
'- выровнять ERROR
COMMON DEF RefreshAVT
  VAR S$, S0$, E$, E0$

  DisplayClear

  GFILL #SCRX,#SCRY+8*4,#SCRX+127*4+3,#SCRY+8*4+3,#CBLACK
  GBOX #SCRX+10*4,#SCRY+38*4,#SCRX+125*4+3,#SCRY+53*4+3,#CBLACK

  GPUTCHR #SCRX,#SCRY+11*4,"T",16,2,2,#CBLACK,#G_NORMAL2
  GPUTCHR #SCRX,#SCRY+20*4,"Z",16,2,2,#CBLACK,#G_NORMAL2
  GPUTCHR #SCRX,#SCRY+29*4,"Y",16,2,2,#CBLACK,#G_NORMAL2
  GPUTCHR #SCRX,#SCRY+42*4,"X",16,2,2,#CBLACK,#G_NORMAL2

  GPUTCHRP #SCRX+9*4,   #SCRY,LEFT$(TIME$(),5),16,2,2,#CBLACK,#G_NORMAL2
' GPUTCHRP #SCRX+9*4,   #SCRY,GetRad$(), 16,2,2,#CBLACK,#G_NORMAL2
  GPUTCHR  #SCRX+82*4,  #SCRY,"", 16,2,2,#CBLACK,#G_NORMAL2
  GPUTCHR  #SCRX+101*4,#SCRY,ToLower$(HEX$(PC,4)),16,1,2,#CBLACK,#G_NORMAL2

  StackPrint #SCRY+11*4,GetT$()
  StackPrint #SCRY+20*4,GetZ$()
  StackPrint #SCRY+29*4,GetY$()

  IF ERROR THEN
    GPUTCHRP #SCRX+21*4, #SCRY+39*4,"ERROR",16,2,4,#CBLACK,#G_NORMAL2
    GPUTCHRP #SCRX+10*4, #SCRY+56*4,ERRORS$[ERROR], \
      16,2,2,#CBLACK,#G_NORMAL2
    ERROR=0
  ELSE
    DecryptRX OUT S0$,S$,E0$,E$
    CASE S0$
      WHEN "#":
        GPUTCHR    #SCRX+21*4, #SCRY+39*4,S$, 16,2,4,#CBLACK,#G_NORMAL2
      WHEN "$": '++ можно выводить больше литер
        GPUTCHR #SCRX+21*4,#SCRY+39*4,StrQ$(S$),16,2,4,#CBLACK,#G_NORMAL2
      WHEN " ": WHEN "G": WHEN "M": WHEN "S":
      WHEN "-": WHEN "g": WHEN "m": WHEN "s":
```

```
        WHEN " ": WHEN "G": WHEN "M": WHEN "S":
        WHEN "-": WHEN "g": WHEN "m": WHEN "s":
          IF INSTR("-gms ",S0$)>=0 THEN S0$="-" ELSE S0$=" "
          IF LEN(S$)<10 THEN
            GPUTCHR  #SCRX+11*4, #SCRY+39*4,S0$,16,2,4,#CBLACK,#G_NORMAL2
            GPUTCHR  #SCRX+21*4, #SCRY+39*4,S$, 16,2,4,#CBLACK,#G_NORMAL2
          ELSEIF LEN(S$)<14 THEN ' Переходим на пропорциональный шрифт
            GPUTCHR  #SCRX+11*4, #SCRY+39*4,S0$,16,2,4,#CBLACK,#G_NORMAL2
            GPUTCHRP #SCRX+21*4, #SCRY+39*4,S$, 16,2,4,#CBLACK,#G_NORMAL2
          ELSE ' Уменьшаем кегль
            GPUTCHR  #SCRX+15*4, #SCRY+43*4,S0$,16,1,2,#CBLACK,#G_NORMAL2
            GPUTCHR  #SCRX+21*4, #SCRY+43*4,S$, 16,1,2,#CBLACK,#G_NORMAL2
          ENDIF

          GPUTCHR    #SCRX+95*4, #SCRY+39*4,E0$,16,2,4,#CBLACK,#G_NORMAL2
          GPUTCHR    #SCRX+105*4,#SCRY+39*4,E$, 16,2,4,#CBLACK,#G_NORMAL2
      ENDCASE

      IF LEN(STATUS$)<16 THEN
        GPUTCHR     #SCRX+10*4, #SCRY+56*4,STATUS$,16,2,2,#CBLACK,#G_NORMAL2
      ELSE
        GPUTCHRP    #SCRX+10*4, #SCRY+56*4,STATUS$,16,2,2,#CBLACK,#G_NORMAL2
      ENDIF
    ENDIF 'Error
  END

  'Обновить экран режима программирования
  COMMON DEF RefreshPRG
    VAR H$, M$, flgBrkP, I, Y=#SCRY, N=(PC-5)AND 65535

    DisplayClear

    FOR I=1 TO 6
      StepDecode N OUT H$,M$,flgBrkP
      IF flgBrkP THEN 'Точка останова
        GFILL #SCRX,Y,#SCRX+128*4-1,Y+9*4-1,#CBLACK
        GPUTCHRP #SCRX,Y,ToLower$(HEX$(N,4)),16,2,2,#CGREEN,#G_NORMAL2
        GPUTCHR  #SCRX+31*4,Y,H$,            16,2,2,#CGREEN,#G_NORMAL2
        GPUTCHR  #SCRX+51*4,Y,M$,            16,2,2,#CGREEN,#G_NORMAL2
      ELSE
        GPUTCHRP #SCRX,Y,ToLower$(HEX$(N,4)),16,2,2,#CBLACK,#G_NORMAL2
        GPUTCHR  #SCRX+31*4,Y,H$,            16,2,2,#CBLACK,#G_NORMAL2
        GPUTCHRP #SCRX+51*4,Y,M$,            16,2,2,#CBLACK,#G_NORMAL2
      ENDIF
      N=(N+1)AND 65535:INC Y,9*4
    NEXT

    GFILL #SCRX,#SCRY+53*4,#SCRX+127*4+3,#SCRY+53*4+3,#CBLACK

    IF LEN(STATUS$)<10 THEN
      GPUTCHR  #SCRX+51*4, #SCRY+56*4,STATUS$,16,2,2,#CBLACK,#G_NORMAL2
    ELSE
      GPUTCHRP #SCRX+51*4, #SCRY+56*4,STATUS$,16,2,2,#CBLACK,#G_NORMAL2
    ENDIF
  END

  'Обновить экран режима шестнадцатеричного редактирования
  COMMON DEF RefreshPRG2
    VAR I, Y=#SCRY, N=(PC-5)AND 65535, S, C

    DisplayClear

    FOR I=1 TO 6
      S=VAR("1:PrgM")[N]:C=S AND 65535
      GPUTCHRP #SCRX,Y,ToLower$(HEX$(N,4)),16,2,2,#CBLACK,#G_NORMAL2
      GPUTCHRP #SCRX+31*4,Y,ToLower$(HEX$(S)>>16,4)+":"+HEX$(C,4)), \
        16,2,2,#CBLACK,#G_NORMAL2
'     GPUTCHRP  #SCRX+116*4,Y,";"+GetUnicode$(C),16,2,2,#CBLACK,#G_NORMAL2
      N=(N+1)AND 65535:INC Y,9*4
    NEXT

    GFILL #SCRX,#SCRY+53*4,#SCRX+127*4+3,#SCRY+53*4+3,#CBLACK

    GPUTCHRP #SCRX,      #SCRY+56*4,"HEX",  16,2,2,#CBLACK,#G_NORMAL2
    GPUTCHRP #SCRX+31*4, #SCRY+56*4,STATUS$,16,2,2,#CBLACK,#G_NORMAL2
  END

  'Обновить экран
  COMMON DEF Refresh
    CASE MODE
      WHEN #AVT:  RefreshAVT
      WHEN #PRG:  RefreshPRG
      WHEN #PRG2: RefreshPRG2
    ENDCASE
  END

  'Предупреждение о том, что разработка МК-261 только началась
  DEF WarnAlpha
    DIALOG "Перед вами - альфа-версия ПМК Будущего."+#CR+\
      "Пожалуйста, дождитесь стабильной бета-версии или"    +#CR+\
      "кандидата в релиз для практического использования."  +#CR+#CR+\
      "Выпуск МК-261 ожидается на отечественном железе."    +#CR+#CR+\
      "WARNING! This is early alpha.",0,"ВНИМАНИЕ"
  END

  @Keys
  DATA 480,600,   10,30,1,7      ' 0 0
  DATA 480,500,   2,29,4,0       ' 1 1
  DATA 640,500,   3,1,5,10       ' 2 2
  DATA 800,500,   19,2,6,11      ' 3 3
  DATA 480,400,   5,28,7,1       ' 4 4
  DATA 640,400,   6,4,8,2        ' 5 5
  DATA 800,400,   15,5,9,3       ' 6 6
  DATA 480,300,   8,27,0,4       ' 7 7
  DATA 640,300,   9,7,38,5       ' 8 8
  DATA 800,300,   16,8,32,6      ' 9 9
  DATA 640,600,   11,0,2,38      '10 ,
  DATA 800,600,   12,10,3,35     '11 /-/
  DATA 960,600,   13,11,19,36    '12 ВП
  DATA 1120,600,  31,12,14,34    '13 СХ
  DATA 1120,500,  22,19,17,13    '14 В↑
  DATA 960,400,   17,9,35,15     '15 +
  DATA 960,300,   18,9,37,15     '16 -
  DATA 1120,400,  21,15,18,14    '17 ×
  DATA 1120,300,  20,16,33,17    '18 ÷
  DATA 960,500,   14,3,15,12     '19 ⏎
  DATA 0,300,     23,18,31,21    '20 F
  DATA 0,400,     24,17,20,22    '21 K
  DATA 0,500,     25,14,21,31    '22 P
  DATA 160,300,   27,20,26,24    '23 ШГ→
  DATA 160,400,   28,21,23,25    '24 ШГ←
  DATA 160,500,   29,22,24,26    '25 В/О
  DATA 160,600,   30,31,25,23    '26 С/П
  DATA 320,300,   7,23,30,28     '27 ИП
  DATA 320,400,   4,24,27,29     '28 ПП
  DATA 320,500,   1,25,28,30     '29 БП
  DATA 320,600,   0,26,29,27     '30 ПП
  DATA 0,600,     26,13,22,20    '31 ГМС
  DATA 800,175,   37,33,35,9     '32 ВЫХОД
  DATA 1120,175,  32,37,34,18    '33 ВВОД
  DATA 1120,75,   35,36,13,33    '34 →
  DATA 800,75,    36,34,11,32    '35 ←
  DATA 960,25,    34,35,12,37    '36 ↑
  DATA 960,125,   33,32,36,16    '37 ↓
  DATA 640,25,    38,38,10,8     '38 ВКЛ

  @Errors
  DATA ""                        '00 нет ошибки
  DATA "?"                       '01 неверное представление внутренних данных
  DATA "Переполнение"            '02 ошибка переполнения
  DATA "Деление на ноль"         '03 ошибка деления на ноль
  DATA "Неверный аргумент"       '04 ошибка аргумента
  DATA "Несуществующий адрес"    '05 ошибка несуществующего адреса
  DATA "Переполнен стек ПП"      '06 переполнение стека подпрограмм
  DATA "?"                       '07 переполнение очереди прерываний
  DATA "?"                       '08 ошибка контрольной суммы памяти программ
  DATA "Ошибка К -"              '09 команда аварийного останова
  DATA "Ошибка К ÷"              '10 команда аварийного останова
  DATA "Ошибка ВП ,"             '11 ошибка формата "ВП,"
  DATA "Переполнение строк"      '12 переполнение памяти строк

  0  2  4  6  8  10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74
PRG0 MAIN.PRG                                                                                    a  11:21
```