

```

REM ' *****
REM '      Подпрограмма Menu  PH_Control_v5.BAS
REM '      VER 4.0
REM ' *****

REM ' *****
REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC
REM ' *****

REM ' *****
REM '      DEMO template
REM ' *****

REM ' *****
REM ' Старт программы примера с нулевого адреса.
REM ' *****
0000
[RESET]
CLR:NOERR 1:GOTO [START]
REM 'CLR:GOTO [START]

REM ' Перезапуск программы после ошибки
0001
CLR:LINIT 1:LPRINT "RESTART":PAUSE 1000:NOERR 1

REM ' IF DEBUG need STOP, ELSE - rem
STOP

GOTO [START]
REM ' *****

REM ' *****
REM '      Подгружаемые модули
REM ' *****
REM ' Выводы контроллера
REM ' *****

#define PUMP_1_ON EQU PORT 3,1
#define PUMP_1_OFF EQU PORT 3,0

#define PUMP_2_ON EQU PORT 4,1
#define PUMP_2_OFF EQU PORT 4,0

REM ' название переменных в программе
REM ' *****

#define PH_Up_Status EQU #(760)
#define PH_Down_Status EQU #(761)

REM ' Таймер
#define RT_OldSec EQU #(762)

REM ' *****

REM ' Счетчик времени показа меню
#define M_Count EQU #(763)
REM ' *****
REM ' PH_module_I_O.BAS

REM ' Адрес датчика
#define PH_Addr EQU 764
REM ' Переменная для вычисление и поиска текста
#define PH_tmp EQU 765
REM ' Счетчик ожидания калибровки
#define PH_Count EQU 766

```

```

REM ' Текстовый буффер запрос/ответ на 32 символа max
#define TXT_Buffer EQU 700

REM ' *****

REM ' Реальная Температура
#define Var_Real_T EQU A
REM ' Реальный PH
#define Var_Real_pH EQU B

REM ' Установленная температура
#define Var_T EQU C
REM ' гистерезис температуры
#define Var_d_T EQU D
REM ' Установленный pH
#define Var_pH EQU E
REM ' гистерезис pH
#define Var_d_pH EQU F

REM ' Адрес модуля
#define Var_addrPH EQU a

REM ' Время работы помпы
#define Var_PumpWork EQU b

REM ' Время паузы помпы
#define Var_PumpPause EQU c

REM ' Режим работы помпы
#define Var_Pump1_mode EQU d
#define Var_Pump2_mode EQU e

REM ' Таймер
#define RT_Count_1 EQU z
#define RT_Count_2 EQU y
#define RT_Count_3 EQU x
#define RT_Count_4 EQU w
#define RT_Count_5 EQU v
REM ' *****
REM ' адреса в еепром
REM ' *****
#define EE_pH EQU 0
#define EE_d_pH EQU 2
#define EE_PumpWork EQU 4
#define EE_PumpPause EQU 6
#define EE_Pump1_mode EQU 8
#define EE_Pump2_mode EQU 10

#define EE_T EQU 12
#define EE_d_T EQU 14
#define EE_addrPH EQU 16

REM ' *****
REM ' Константы и текст
REM ' *****

REM ' ИНТЕРВАЛ опроса PH , sec >=3
#define TimeIntervalPH EQU 5

REM ' таймаут меню , сек
#define TimeOut EQU 10
#define TimeOut_1 EQU 6

```

```

REM ' Адрес модуля PH
#define MIN_addr EQU 48
#define MAX_addr EQU 57
REM ' название на экране
#define TXT_A_pH EQU Адрес PH
#define TXT1_A_pH EQU Адр.

REM ' min/max Температуры - настройка пределов выбора в меню
#define MIN_T EQU 100
#define MAX_T EQU 400
REM ' название на экране
#define TXT_T EQU Температура
#define TXT1_T EQU Т

REM ' Гистерезис Температуры
#define MIN_d_T EQU 0
#define MAX_d_T EQU 50
REM ' название на экране
#define TXT_d_T EQU Гистерезис Т

REM ' min/max pH - настройка пределов выбора в меню
#define MIN_pH EQU 10
#define MAX_pH EQU 140
REM ' название на экране
#define TXT_pH EQU Уровень pH
#define TXT1_pH EQU pH

REM ' Гистерезис pH
#define MIN_d_pH EQU 0
#define MAX_d_pH EQU 30
REM ' название на экране
#define TXT_d_pH EQU Гистерезис pH

REM ' Подача сек (0-60)
#define MIN_PumpWork EQU 0
#define MAX_PumpWork EQU 60

REM ' название на экране
#define TXT_PumpWork EQU Подача, сек

REM ' Пауза сек (0-360)
#define MIN_PumpPause EQU 0
#define MAX_PumpPause EQU 360

REM ' название на экране
#define TXT_PumpPause EQU Пауза, сек

REM ' название на экране
#define TXT_Pump1 EQU Насос 1
#define TXT_Pump2 EQU Насос 2

#define MIN_ModePump EQU 0
#define MAX_ModePump EQU 2

REM ' название на экране
#define TXT_0 EQU Кислота
#define TXT_1 EQU Щелочь
#define TXT_2 EQU Выкл

#define ModePUMP_OFF EQU 2
#define ModePUMP_Shel EQU 1
#define ModePUMP_Kisl EQU 0

REM ' Группы
#define TXT_Group_0 EQU "Настройки PH"
#define TXT_Group_1 EQU "Гистерезис PH"
#define TXT_Group_2 EQU "Время работы"
#define TXT_Group_3 EQU "Время паузы"
#define TXT_Group_4 EQU "Настройка насоса"

```

```

#define TXT_Group_5 EQU "Настройка темп-ры"
#define TXT_Group_6 EQU "Гистерезис темп."
#define TXT_Group_7 EQU "Датчик PH"
#define TXT_Group_8 EQU "Калибровка PH"
REM '*****

REM '*****
REM ' Меню Управления - ВЕРХНИЙ УРОВЕНЬ
REM '*****

REM '*****
REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC
REM '*****

REM '*****
REM '
REM ' Расположение и коды кнопок
REM '*****
REM '
REM ' *****
REM ' * *
REM ' * *
REM ' * *
REM ' *****
REM '
REM ' O O O O O
REM ' <- -> ok up down
REM '*****
REM ' Коды 16 8 2 4 1 - USB
REM ' Кнопки 16 1 4 8 2 - Device
REM '*****
REM ' *** USB ***
#define _up EQU 4
#define _down EQU 1
#define _left EQU 16
#define _right EQU 8
#define _ok EQU 2
REM ' timeout
#define _to EQU 0
REM ' *** Device ***
REM '#define _up EQU 8
REM '#define _down EQU 2
REM '#define _left EQU 16
REM '#define _right EQU 1
REM '#define _ok EQU 4
REM '#define _to EQU 0
REM '*****
[GM_Timeout]
[GM_Exit]
§=GKEY{}:IF §<>0 THEN GOTO +0
RETURN

[Global_MENU]
REM '*****

```

```

REM ' ЗАПУСКАЕМ программу MENU
REM ' *****

REM ' Первый пункт меню , покажем
REM ' если вниз - на следующий, вверх - предыдущий
REM ' лево право - ничего или выход см. ниже
[LEVEL_1_1]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_0
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_10], _down, [LEVEL_1_2], _ok, [SEL_1_1]
RETURN:REM GOTO [LEVEL_1_1]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_2]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_1
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_1], _down, [LEVEL_1_3], _ok, [SEL_1_2]
RETURN:REM GOTO [LEVEL_1_2]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_3]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_2
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_2], _down, [LEVEL_1_4], _ok, [SEL_1_3]
RETURN:REM GOTO [LEVEL_1_3]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_4]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_3
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_3], _down, [LEVEL_1_5], _ok, [SEL_1_4]
RETURN:REM GOTO [LEVEL_1_4]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_5]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_4
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_4], _down, [LEVEL_1_6], _ok, [SEL_1_5]
RETURN:REM GOTO [LEVEL_1_5]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_6]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM ' Покажем меню
CLS:AT 0:LPRINT "          МЕНЮ          "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_5
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_5], _down, [LEVEL_1_7], _ok, [SEL_1_6]
RETURN:REM GOTO [LEVEL_1_6]:REM ' ВЫХОД ИЛИ ОСТАТЬСЯ

[LEVEL_1_7]

```

```

§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "      МЕНЮ      "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_6
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_6], _down, [LEVEL_1_8], _ok, [SEL_1_7]
RETURN:REM GOTO [LEVEL_1_7]:REM 'выход или остаться

[LEVEL_1_8]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "      МЕНЮ      "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_7
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_7], _down, [LEVEL_1_9], _ok, [SEL_1_8]
RETURN:REM GOTO [LEVEL_1_8]:REM 'выход или остаться

[LEVEL_1_9]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "      МЕНЮ      "
REM ' Показ 1 пункта
AT 64:LPRINT TXT_Group_8
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_8], _down, [LEVEL_1_10], _ok, [SEL_1_9]
RETURN:REM GOTO [LEVEL_1_9]:REM 'выход или остаться

[LEVEL_1_10]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "      МЕНЮ      "
REM ' Показ 1 пункта
AT 64:LPRINT "      Выйти"
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_1_9], _down, [LEVEL_1_11], _ok, [GM_Exit]
RETURN:REM GOTO [LEVEL_1_10]:REM 'выход или остаться

REM '*****
REM '*****

REM '*****
REM 'Вызов подчиненного пункта меню, изменение или подменю
[SEL_1_1]

REM '*** пример изменния *****
CLS:AT 0:LPRINT "Меню>TXT_pH"
REM §=1 размерность параметра (число знаков после запятой)
REM §=3 размерность параметра (число знаков всего)
REM §=140 максимальное значение
REM §=10 минимальное значение
REM §=0 адрес хранения параметра в EEPROM
REM '*****
§=1:§=3:§=MAX_pH:§=MIN_pH:§=EE_pH

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_1]
REM '*****

```

```

REM '*****
[SEL_1_2]
CLS:AT 0:LPRINT "Меню>TXT_d_pH"
§=1:§=2:§=MAX_d_pH:§=MIN_d_pH:§=EE_d_pH

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_2]
REM '*****

REM '*****
[SEL_1_3]
CLS:AT 0:LPRINT "Меню>TXT_PumpWork"
§=0:§=2:§=MAX_PumpWork:§=MIN_PumpWork:§=EE_PumpWork

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_3]
REM '*****

REM '*****
[SEL_1_4]
CLS:AT 0:LPRINT "Меню>TXT_PumpPause"
§=0:§=3:§=MAX_PumpPause:§=MIN_PumpPause:§=EE_PumpPause

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_4]
REM '*****

REM '*****
[SEL_1_6]
CLS:AT 0:LPRINT "Меню>TXT_T"
§=1:§=3:§=MAX_T:§=MIN_T:§=EE_T

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_6]
REM '*****

REM '*****
[SEL_1_7]
CLS:AT 0:LPRINT "Меню>TXT_d_T"
§=1:§=3:§=MAX_d_T:§=MIN_d_T:§=EE_d_T

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_1_7]
REM '*****

REM '*****
[SEL_1_8]
CLS:AT 0:LPRINT "Меню>TXT_A_pH"
§=0:§=3:§=MAX_addr:§=MIN_addr:§=EE_addrPH

GOSUB [Change_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
Var_addrPH=RDEE{EE_addrPH}:GOSUB [INIT_PH_1]
GOTO [LEVEL_1_8]
REM '*****

REM '*****
REM Меню второго уровня - 1
REM '*****
[SEL_1_9]

```

```

[SEL_1_9_1]
#(PH_Count)=0
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>>Калибр.4"
[SEL_1_9_1a]
#(PH_Count)=#(PH_Count)+1:IF #(PH_Count)>50 THEN #(PH_Count)=0:GOTO [GM_Timeout]
AT 64:LPRINT "Запрос"
GOSUB [Read_PH]
PAUSE 500
AT 64:LPRINT %3,.1,"pH=",Var_Real_pH,"T=",Var_Real_T,"OK>Go"
§=KEY{TimeOut_1}:BEEP 15,2
CASE §,_up,[SEL_1_9_4],_down,[SEL_1_9_2],_ok,[SEL_1_9_1go]
GOTO [SEL_1_9_1a]

```

```

[SEL_1_9_1go]
AT 64:LPRINT "Выполняется"
Var_addrPH=RDEE{EE_addrPH}
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"CL6"
PAUSE 100:SGET #(TXT_Buffer):PAUSE 5500
GOTO [SEL_1_9_1]

```

```

[SEL_1_9_2]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>>Калибр.7"
[SEL_1_9_2a]
#(PH_Count)=#(PH_Count)+1:IF #(PH_Count)>50 THEN #(PH_Count)=0:GOTO [GM_Timeout]
AT 64:LPRINT "Запрос"
GOSUB [Read_PH]
PAUSE 500
AT 64:LPRINT %3,.1,"pH=",Var_Real_pH,"T=",Var_Real_T,"OK>Go"
§=KEY{TimeOut_1}:BEEP 15,2
CASE §,_up,[SEL_1_9_1],_down,[SEL_1_9_3],_ok,[SEL_1_9_2go]
GOTO [SEL_1_9_2a]

```

```

[SEL_1_9_2go]
AT 64:LPRINT "Выполняется"
Var_addrPH=RDEE{EE_addrPH}
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"CL4"
PAUSE 100:SGET #(TXT_Buffer):PAUSE 5500
GOTO [SEL_1_9_2]

```

```

[SEL_1_9_3]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>>Сброс Калибр."
[SEL_1_9_3a]
#(PH_Count)=#(PH_Count)+1:IF #(PH_Count)>50 THEN #(PH_Count)=0:GOTO [GM_Timeout]
AT 64:LPRINT "Запрос"
GOSUB [Read_PH]
PAUSE 500
AT 64:LPRINT %3,.1,"pH=",Var_Real_pH,"T=",Var_Real_T,"OK>Go"
§=KEY{TimeOut_1}:BEEP 15,2
CASE §,_up,[SEL_1_9_2],_down,[SEL_1_9_4],_ok,[SEL_1_9_3go]
GOTO [SEL_1_9_3a]

```

```

[SEL_1_9_3go]

```



```
AT 64:LPRINT "Выполняется"
Var_addrPH=RDEE{EE_addrPH}
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr), "CL3"
PAUSE 100:SGET #(TXT_Buffer):PAUSE 5500
GOTO [SEL_1_9_3]
```

```
[SEL_1_9_4]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>> Выход"
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [SEL_1_9_3], _down, [SEL_1_9_1], _ok, [LEVEL_1_9]
RETURN:REM GOTO [SEL_1_9_4]:REM 'выход или остаться
```

```
REM '*****
REM Меню второго уровня - 2
REM '*****
[SEL_1_5]
```

```
[LEVEL_2_1]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>>"
AT 64:LPRINT "TXT_Pump1"
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_2_3], _down, [LEVEL_2_2], _ok, [SEL_2_1]
RETURN:REM GOTO [LEVEL_1_1]:REM 'выход или остаться
```

```
[LEVEL_2_2]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>>"
AT 64:LPRINT "TXT_Pump2"
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_2_1], _down, [LEVEL_2_3], _ok, [SEL_2_2]
RETURN:REM GOTO [LEVEL_1_1]:REM 'выход или остаться
```

```
[LEVEL_2_3]
§=GKEY{}:IF §<>0 THEN GOTO +0
REM 'Покажем меню
CLS:AT 0:LPRINT "Меню>> Выход"
§=KEY{TimeOut}:BEEP 15,2
CASE §, _to, [GM_Timeout], _up, [LEVEL_2_2], _down, [LEVEL_2_1], _ok, [SEL_2_3]
RETURN:REM GOTO [LEVEL_1_1]:REM 'выход или остаться
```

```
REM '*****
[SEL_2_1]
CLS:AT 0:LPRINT "Меню>>TXT_Pump1"
REM §=140 максимальное значение
REM §=10 минимальное значение
REM §=0 адрес хранения параметра в EEPROM
§=2:§=0:§=EE_Pump1_mode
GOSUB [SEL_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_2_1]
REM '*****
```

```
REM '*****
```

```

[SEL_2_2]
CLS:AT 0:LPRINT "Меню>>TXT_Pump2"
§=2:§=0:§=EE_Pump2_mode
GOSUB [SEL_PARAM_EE]
IF §=0 THEN GOTO [GM_Timeout]
GOTO [LEVEL_2_2]
REM '*****

[SEL_2_3]
GOTO [LEVEL_1_5]
REM '*****

REM '*****
REM ' Меню , второй уровень
REM ' FOR use - ADD #include CHANGE_PARAM_V4.BAS
REM '*****
REM ' Функция изменения параметра в EEPROM с кнопок и
REM ' отображение его значения на дисплее во второй строке
REM ' *****

REM ' *****
REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC
REM '*****

REM ' Передаваемые параметры

REM ' §=размерность параметра (число знаков после запятой)
REM ' §=размерность параметра (число знаков всего)
REM ' §=адрес текста названия параметра
REM ' §=максимальное значение
REM ' §=минимальное значение
REM ' §=адрес хранения параметра в EEPROM

REM ' возвращает в стек 0 если таймаут

REM ' Используются стек переменных +3
REM ' Обычные переменные не используются
REM А сам параметр
REM Б адрес хранения параметра в EEPROM
REM В минимальное значение
REM Г максимальное значение
REM Д размерность параметра (число знаков всего)
REM Е размерность параметра (число знаков после запятой)

REM ' sample
REM §=1 размерность параметра (число знаков после запятой)
REM §=3 размерность параметра (число знаков всего)
REM §=140 максимальное значение
REM §=10 минимальное значение
REM §=0 адрес хранения параметра в EEPROM
REM GOSUB [Change_PARAM]

REM ' таймаут меню 2 УРОВНЯ, сек
REM '#define TimeOut EQU 10

REM '*****
REM ' Расположение и коды кнопок

```

```

REM ' *****
REM ' *****
REM ' * *
REM ' * *
REM ' * *
REM ' *****
REM ' 0 0 0 0 0
REM ' <- -> ok up down
REM ' *****
REM ' Коды 16 8 2 4 1 - USB
REM ' Кнопки 16 1 4 8 2 - Device
REM ' *****
REM ' *** USB ***
REM ' #define _up EQU 4
REM ' #define _down EQU 1
REM ' #define _left EQU 16
REM ' #define _right EQU 8
REM ' #define _ok EQU 2
REM ' #define _to EQU 0
REM ' *** Device ***
REM ' #define _up EQU 8
REM ' #define _down EQU 2
REM ' #define _left EQU 16
REM ' #define _right EQU 1
REM ' #define _ok EQU 4
REM ' #define _to EQU 0
REM ' *****

REM ' *****
[Change_PARAM_EE]
§+=A=RDEE{B}

[CH_1_1]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
GOTO [SEL_CH_1_1]

AT 64:LPRINT " Изменить "
§=KEY{TimeOut}:BEEP 15,2
CASE
§, _up, [CH_1_3], _down, [CH_1_2], _ok, [SEL_CH_1_1], _left, [CH_1_3], _right, [CH_1_2], 0, [CH_EE_timeout]
GOTO [CH_1_1]

[CH_1_2]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
AT 64:LPRINT " Выход без сохран"
§=KEY{TimeOut}:BEEP 15,2
CASE
§, _up, [CH_1_1], _down, [CH_1_3], _ok, [SEL_CH_1_2], _left, [CH_1_1], _right, [CH_1_3], 0, [CH_EE_timeout]
GOTO [CH_1_2]

[CH_1_3]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
AT 64:LPRINT " Сохранить, выйти"
§=KEY{TimeOut}:BEEP 15,2
CASE
§, _up, [CH_1_2], _down, [CH_1_1], _ok, [SEL_CH_1_3], _left, [CH_1_2], _right, [CH_1_1], 0, [CH_EE_timeout]
GOTO [CH_1_3]

[SEL_CH_1_1]
IF A>Г THEN A=B
IF A<B THEN A=Г
REM ' ждем отпуска
AT 64:LPRINT %Д, .Е, " ", А, " "
§=GKEY{}:IF §<>0 THEN GOTO +0

§=KEY{TimeOut}:BEEP 15,2

```

```

CASE
§,0,[CH_EE_timeout],_ok,[CH_1_2],_up,[INC_P],_down,[DEC_P],_left,[dec_P],_right,[inc_P]
GOTO [CH_1_1]
REM ' изменение
[inc_P]
A=A+1:GOTO [SEL_CH_1_1]
[dec_P]
A=A-1:GOTO [SEL_CH_1_1]
[INC_P]
A=A+15:GOTO [SEL_CH_1_1]
[DEC_P]
A=A-15:GOTO [SEL_CH_1_1]

[SEL_CH_1_2]
§-:§-:§-:§-:§-:§-:§=1:RETURN
[SEL_CH_1_3]
WREE A,B:§-:§-:§-:§-:§-:§-:§=1:RETURN

REM ' *****

[CH_EE_timeout]
§-:§-:§-:§-:§-:§=0:RETURN
REM ' *****
REM ' Меню , второй уровень
REM ' FOR use - ADD #include SEL_PARAM_V4.BAS
REM ' *****
REM ' Функция изменения параметра в EEPROM с кнопок и
REM ' отображение его Названия на дисплее во второй строке
REM ' *****

REM ' *****
REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC
REM ' *****

REM ' Передаваемые праметры

REM ' §=максимальное значение
REM ' §=минимальное значение
REM ' §=адрес хранения параметра в EEPROM

REM ' возвращает в стек 0 если таймаут

REM ' Используются стек переменных +3
REM ' Обычные переменные не используются
REM A сам параметр
REM B адрес хранения параметра в EEPROM
REM V минимальное значение
REM Г максимальное значение

REM ' sample
REM §=140 максимальное значение
REM §=10 минимальное значение
REM §=0 адрес хранения параметра в EEPROM
REM GOSUB [Change_PARAM]

REM ' таймаут меню 2 УРОВНЯ, сек
REM '#define TimeOut EQU 10

REM ' *****
REM ' Расположение и коды кнопок
REM ' *****
REM ' *****
REM ' * *
REM ' * *
REM ' * *
REM ' *****
REM ' O O O O O
REM ' <- -> ok up down
REM ' *****

```

```

REM ' Коды      16      8      2      4      1 - USB
REM ' Кнопки    16      1      4      8      2 - Device
REM '*****
REM '   *** USB ***
REM '#define  _up    EQU    4
REM'#define  _down  EQU    1
REM'#define  _left  EQU   16
REM'#define  _right EQU    8
REM'#define  _ok    EQU    2
REM'#define  _to    EQU    0
REM '   *** Device ***
REM'#define  _up    EQU    8
REM'#define  _down  EQU    2
REM'#define  _left  EQU   16
REM'#define  _right EQU    1
REM'#define  _ok    EQU    4
REM'#define  _to    EQU    0
REM '*****

REM '*****
[SEL_PARAM_EE]
§+:A=RDEE{B}

[SL_1_1]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
GOTO [SL_SEL_1_1]

AT 64:LPRINT "      Изменить      "
§=KEY{TimeOut}:BEEP 15,2
IF §=0 THEN §-:§-:§-:§-:§=0:GOTO [SL_Exit]
§+:CASE §, _up, [SL_1_3], _down, [SL_1_2], _ok, [SL_SEL_1_1], _left, [SL_1_3], _right, [SL_1_2]
GOTO [SL_Exit_Timeout]

[SL_1_2]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
AT 64:LPRINT "Выход без сохран"
§=KEY{TimeOut}:BEEP 15,2
IF §=0 THEN §-:§-:§-:§-:§=0:GOTO [SL_Exit]
§+:CASE §, _up, [SL_1_1], _down, [SL_1_3], _ok, [SL_SEL_1_2], _left, [SL_1_1], _right, [SL_1_3]
GOTO [SL_Exit_Timeout]

[SL_1_3]
§=GKEY{}:IF §<>0 THEN GOTO +0:REM ' ждем отпуска
AT 64:LPRINT "Сохранить, выйти"
§=KEY{TimeOut}:BEEP 15,2
IF §=0 THEN §-:§-:§-:§-:§=0:GOTO [SL_Exit]
§+:CASE §, _up, [SL_1_2], _down, [SL_1_1], _ok, [SL_SEL_1_3], _left, [SL_1_2], _right, [SL_1_1]
GOTO [SL_Exit_Timeout]

[SL_SEL_1_1]
IF A>Г THEN A=B
IF A<B THEN A=Г
REM ' ждем отпуска
§=GKEY{}:IF §<>0 THEN GOTO +0

IF A=ModePUMP_Kisl THEN AT 64:LPRINT "      TXT_0      "
IF A=ModePUMP_Shel THEN AT 64:LPRINT "      TXT_1      "
IF A=ModePUMP_OFF THEN AT 64:LPRINT "      TXT_2      "

§=KEY{TimeOut}:BEEP 15,2
CASE §, _ok, [SL_1_2], _up, [INC_SL], _down, [DEC_SL], _left, [dec_SL], _right, [inc_SL]
GOTO [SL_Exit_Timeout]
REM ' изменение
[inc_SL]
A=A+1:GOTO [SL_SEL_1_1]
[dec_SL]
A=A-1:GOTO [SL_SEL_1_1]
[INC_SL]
A=A+1:GOTO [SL_SEL_1_1]

```

```

[DEC_SL]
A=A-1:GOTO [SL_SEL_1_1]

[SL_SEL_1_2]
§=GKEY{}:IF §<>0 THEN GOTO +0
§-:§-:§-:§-:§=1:GOTO [SL_Exit]
[SL_SEL_1_3]
§=GKEY{}:IF §<>0 THEN GOTO +0
WREE A,B:§-:§-:§-:§-:§=1:GOTO [SL_Exit]

REM '*****
[SL_Exit_Timeout]
§=GKEY{}:IF §<>0 THEN GOTO +0
§-:§-:§-:§-:§=0
[SL_Exit]
§=GKEY{}:IF §<>0 THEN GOTO +0
RETURN
REM '*****
REM '*****
[INIT_PH_CMD]
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"MD1":PAUSE 100
RETURN

[START_PH_CMD]
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"ST0":PAUSE 20
RETURN

REM '*****
REM '*****
[INIT_PH]
CLS:LPRINT "INIT pH"
[INIT_PH_1]
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"MDO":PAUSE 100
SPRINT $(PH_Addr),"IT2"

[INFO_PH_CMD]
SPRINT $(PH_Addr),"ATI":PAUSE 100:SGET #(TXT_Buffer)
§=INSTR${$(TXT_Buffer),"PH"}:AT 64
IF §=0 THEN LPRINT "Датчик не найден":PAUSE 2000:RETURN
LPRINT "Датчик ",Var_addrPH," найден":PAUSE 2000:RETURN
REM '*****
REM '*****
[Read_PH]
Var_Real_T=999
Var_Real_pH=999
#(PH_Addr)=Var_addrPH
SPRINT $(PH_Addr),"GT0":PAUSE 50:SGET #(TXT_Buffer)
IF #(TXT_Buffer)=0 THEN RETURN
#(PH_tmp)=INSTR${$(TXT_Buffer),"H="}
IF #(PH_tmp)=0 THEN RETURN
#(PH_tmp)=#(PH_tmp)+2:Var_Real_pH=VAL{$(TXT_Buffer),#(PH_tmp)}
Var_Real_pH=Var_Real_pH/10

SPRINT $(PH_Addr),"GT1":PAUSE 50:SGET #(TXT_Buffer)
IF #(TXT_Buffer)=0 THEN RETURN
#(PH_tmp)=INSTR${$(TXT_Buffer),"T="}:#(PH_tmp)=#(PH_tmp)+2
IF #(PH_tmp)=2 THEN RETURN
Var_Real_T=VAL{$(TXT_Buffer),#(PH_tmp)}

RETURN
REM '*****
REM '*****

REM '*****

```

```

[WORK_PH]
IF PH_Up_Status=1 THEN GOTO [WORK_PH_Up]
IF PH_Down_Status=1 THEN GOTO [WORK_PH_Down]
PUMP_1_OFF:PUMP_2_OFF
RETURN
[WORK_PH_Up]
IF Var_Pump1_mode<>ModePUMP_Kisl THEN GOTO [Pump1_Timer]
IF Var_Pump2_mode<>ModePUMP_Kisl THEN GOTO [Pump2_Timer]
PUMP_1_OFF:PUMP_2_OFF
RETURN
[WORK_PH_Down]
IF Var_Pump1_mode<>ModePUMP_Shel THEN GOTO [Pump1_Timer]
IF Var_Pump2_mode<>ModePUMP_Shel THEN GOTO [Pump2_Timer]
PUMP_1_OFF:PUMP_2_OFF
RETURN
REM '*****
REM '      функция удержания в пределе

[PH_PREDEL]
IF Var_Real_pH>Var_pH+Var_d_pH THEN PH_Up_Status=1
IF Var_Real_pH<=Var_pH THEN PH_Up_Status=0

IF Var_Real_pH<Var_pH-Var_d_pH THEN PH_Down_Status=1
IF Var_Real_pH>=Var_pH THEN PH_Down_Status=0
RETURN

REM '*****
REM ' Таймер времени - два шт REAL TIMER
REM ' каждую секунду значение увеличивается на 1 до 32767
REM ' Должен постоянно вызываться в MAIN
REM ' GOSUB [R_Timer]
REM '*****

[R_Timer]
§=SEC{}
IF §=RT_OldSec THEN RETURN
RT_Count_1=RT_Count_1+1:IF RT_Count_1>32760 THEN RT_Count_1=32760
RT_Count_2=RT_Count_2+1:IF RT_Count_2>32760 THEN RT_Count_2=32760
RT_Count_3=RT_Count_3+1:IF RT_Count_3>32760 THEN RT_Count_3=32760
RT_Count_4=RT_Count_4+1:IF RT_Count_4>32760 THEN RT_Count_4=32760
RT_Count_5=RT_Count_5+1:IF RT_Count_5>32760 THEN RT_Count_5=32760
§+:RT_OldSec=§
RETURN

REM '*****
REM '*****
REM 'Интервальный таймер (параметры: флаг - 0,1 , интервал работы,
REM 'адрес для сохранения значения 0,1      выполняется один раз в
REM 'указанный интервал

[PH_Timer]
GOSUB [R_Timer]
IF RT_Count_1<TimeIntervalPH THEN RETURN
RT_Count_1=0
GOSUB [Read_PH]
GOSUB [PH_PREDEL]
RETURN
REM '*****
REM '*****
REM ' Таймер 1 помпы
REM ' Возвращает в стеке включение
[Pump1_Timer]
IF Var_Pump1_mode=ModePUMP_OFF THEN RT_Count_2=0:PUMP_1_OFF:RETURN
IF RT_Count_2>Var_PumpWork+Var_PumpPause THEN RT_Count_2=0
IF Var_PumpWork>RT_Count_2 THEN PUMP_1_ON:RETURN
PUMP_1_OFF:RETURN

REM '*****
REM '*****
REM ' Таймер 2 помпы

```

```

[Pump2_Timer]
IF Var_Pump2_mode=ModePUMP_OFF THEN RT_Count_3=0:PUMP_2_OFF:RETURN
IF RT_Count_3>Var_PumpWork+Var_PumpPause THEN RT_Count_3=0
IF Var_PumpWork>RT_Count_3 THEN PUMP_2_ON:RETURN
PUMP_2_OFF:RETURN

```

```

REM '*****
REM ' Настройки
REM '*****
REM '1. Уровень pH (1.0 - 14.0)
REM '2. Гистерезис (0.0 - 3.0)
REM '3. Подача сек (0-60)
REM '4. Пауза сек (0-360)
REM '5. Насос 1 (0-Кислота, 1-Щелочь, 2-Выкл)
REM '6. Насос 2 (0-Кислота, 1-Щелочь, 2-Выкл)

```

```

REM '*****
REM ' Start DEMO system
REM '*****

```

```

[START]
LINIT 1:BEEP 15,2:UART 1,5
LPRINT "PH DEMO":AT 64:LPRINT "V5.0 Start"
PAUSE 5500

```

```

REM ' LOAD data
GOSUB [LoadData]
GOSUB [TestValidData]

```

```

GOSUB [INIT_PH]
PAUSE 2000
CLS:LPRINT "Device work"
PAUSE 2000

```

```

REM '*****
REM '*****

```

```

[MAIN]
GOSUB [PH_Timer]
GOSUB [WORK_PH]

```

```

M_Count=M_Count+1:IF M_Count>200 THEN M_Count=0

```

```

IF M_Count=100 THEN GOSUB [Read_PH]

```

```

IF M_Count>100 THEN GOTO [Ekran_2]

```

```

[Ekran_1]
REM ' Вывод текущих настроек
AT 0:LPRINT %3,.1,"TXT1_pH=",Var_pH
LPRINT %2,.1," +/-",Var_d_pH," "
AT 64:LPRINT %0,.0,Var_PumpWork,"/",Var_PumpPause
IF Var_Pump1_mode=0 THEN LPRINT "TXT_0","/"
IF Var_Pump1_mode=1 THEN LPRINT "TXT_1","/"
IF Var_Pump1_mode=2 THEN LPRINT "TXT_2","/"
IF Var_Pump2_mode=0 THEN LPRINT "TXT_0"
IF Var_Pump2_mode=1 THEN LPRINT "TXT_1"
IF Var_Pump2_mode=2 THEN LPRINT "TXT_2"
GOTO [END_Ekran]

```

```

[Ekran_2]
REM ' Вывод текущих настроек_2
AT 0:LPRINT %3,.1,"TXT1_T=",Var_T

```



```

LPRINT %2,.1," +/-",Var_d_T," "

AT 64:LPRINT %2,.0,"A:",Var_addrPH,"pH"
LPRINT %3,.1,Var_Real_pH,"T:",Var_Real_T

GOTO [END_Ekran]

[END_Ekran]

REM ' функцию можно вызывать в любом месте программы
REM ' если в момент вызова не нажата кнопка то
REM ' происходит выход, с кодом '0', дисплей не изменяется
GOSUB [MENU]

GOTO [MAIN]
REM '*****
REM '*****

REM '*****
[INIT_DEVICE]
REM ' программа устройства , отключающее все работающее
REM ' на время нахождения в меню
PUMP_1_OFF:PUMP_2_OFF:CLS
RETURN

REM '*****
[LoadData]
Var_pH=RDEE{EE_pH}
Var_d_pH=RDEE{EE_d_pH}
Var_PumpWork=RDEE{EE_PumpWork}
Var_PumpPause=RDEE{EE_PumpPause}
Var_Pump1_mode=RDEE{EE_Pump1_mode}
Var_Pump2_mode=RDEE{EE_Pump2_mode}

Var_T=RDEE{EE_T}
Var_d_T=RDEE{EE_d_T}
Var_addrPH=RDEE{EE_addrPH}
RETURN

[SaveData]
WREE Var_pH,EE_pH
WREE Var_d_pH,EE_d_pH
WREE Var_PumpWork,EE_PumpWork
WREE Var_PumpPause,EE_PumpPause
WREE Var_Pump1_mode,EE_Pump1_mode
WREE Var_Pump2_mode,EE_Pump2_mode

WREE Var_T,EE_T
WREE Var_d_T,EE_d_T
WREE Var_addrPH,EE_addrPH
RETURN

REM '*****
REM '      Выполнить 1 раз перед тестовой программой
REM '*****
[TestValidData]
IF Var_pH>MAX_pH THEN GOTO [SetDefault]
IF Var_pH<MIN_pH THEN GOTO [SetDefault]

IF Var_d_pH>MAX_d_pH THEN GOTO [SetDefault]
IF Var_d_pH<MIN_d_pH THEN GOTO [SetDefault]

IF Var_PumpWork>MAX_PumpWork THEN GOTO [SetDefault]
IF Var_PumpWork<MIN_PumpWork THEN GOTO [SetDefault]

IF Var_PumpPause>MAX_PumpPause THEN GOTO [SetDefault]

```

```

IF Var_PumpPause<MIN_PumpPause THEN GOTO [SetDefault]

IF Var_Pump1_mode>MAX_ModePump THEN GOTO [SetDefault]
IF Var_Pump1_mode<MIN_ModePump THEN GOTO [SetDefault]

IF Var_Pump2_mode>MAX_ModePump THEN GOTO [SetDefault]
IF Var_Pump2_mode<MIN_ModePump THEN GOTO [SetDefault]

IF Var_T>MAX_T THEN GOTO [SetDefault]
IF Var_T<MIN_T THEN GOTO [SetDefault]

IF Var_d_T>MAX_d_T THEN GOTO [SetDefault]
IF Var_d_T<MIN_d_T THEN GOTO [SetDefault]

IF Var_addrPH>MAX_addr THEN GOTO [SetDefault]
IF Var_addrPH<MIN_addr THEN GOTO [SetDefault]

RETURN
[SetDefault]
Var_pH=70:Var_d_pH=10:Var_PumpWork=10
Var_PumpPause=150:Var_Pump1_mode=1:Var_Pump2_mode=2
Var_d_T=10:Var_T=200:Var_addrPH=48
GOSUB [SaveData]
RETURN

REM '*****
REM '              END DEMO system
REM '*****

REM '$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

REM '*****
REM '              File - MENU.BAS
REM '*****
REM '*****
REM '      Подпрограмма Menu  MENU.BAS
REM '      VER 4.0
REM '*****

REM ' *****
REM ' Требуется ОБРАБОТКА ПРЕКОМПИЛЯТОРОМ BASIC
REM ' *****

REM '*****
REM '      Вызов меню по нажатию
REM '      Если кнопка ОК не нажата, то данный раздел
REM '      не вызывается
REM '*****

[ MENU ]
REM ' проверка надо ли запускать ?
§=GKEY{}:IF §=0 THEN RETURN
REM ' обработка случайного нажатия
BEEP 15,2:CLS:LPRINT "  Вход в МЕНЮ-ОК  "
PAUSE 100:§=KEY{2}:BEEP 15,2:IF §=_ok THEN GOTO [MENU_START]
§=GKEY{}:IF §<>0 THEN GOTO +0
RETURN

[ MENU_START ]
REM ' Перед запуском все выключить
BEEP 15,2:GOSUB [INIT_DEVICE]
REM ' запуск самого меню
GOSUB [Global_MENU]

```

```
REM ' LOAD UPDATE data  
GOSUB [LoadData]  
RETURN  
REM '*****
```